

Claudio Quirós, Adalberto (2023) *Deep unsupervised learning of cancer tissue representations.* PhD thesis.

https://theses.gla.ac.uk/83374/

Copyright and moral rights for this work are retained by the author

A copy can be downloaded for personal non-commercial research or study, without prior permission or charge

This work cannot be reproduced or quoted extensively from without first obtaining permission in writing from the author

The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the author

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given

Enlighten: Theses
https://theses.gla.ac.uk/
research-enlighten@glasgow.ac.uk

DEEP UNSUPERVISED LEARNING OF CANCER TISSUE REPRESENTATIONS

Adalberto Claudio Quirós

Submitted in fulfilment of the requirements for the Degree of
Doctor of Philosophy

School of Computing Science
College of Science and Engineering
University of Glasgow



January 2023

Abstract

Histopathological images of tumors contain abundant information about how tumors grow and how they interact with their micro-environment. Characterizing and improving our understanding of phenotypes can reveal factors related to tumor progression and their underlying biological processes, ultimately improving diagnosis and treatment. In recent years, the field of deep learning applications in pathology has witnessed a steady progress, yet most of these applications focus on a supervised approach, relating tissue with associated sample labels or annotations.

The impact of supervised approaches is limited by three factors. Firstly, high-quality labels are expensive in time and effort, which makes them not easily scalable. Secondly, these methods focus on identifying known histologic patterns on pathology images, fundamentally restricting the discovery of new tissue phenotypes. Thirdly, supervised methods rely on annotations by experts and oftentimes there is significant subjectivity and bias in determining certain histologic patterns. These limitations emphasize the importance of using novel methods capable of characterizing tissue by the entire spectrum of features enclosed in the image, without pre-defined annotation or supervision.

This thesis explores the use of generative adversarial networks and self-supervised learning for unsupervised learning of cancer tissue representations, making the following contributions.

We propose PathologyGAN, a generative adversarial network(GAN) for unsupervised representation learning of cancer tissue images. We demonstrate that PathologyGAN generates high-fidelity tissue images through quantitative measures and pathologist analyses. Additionally, we show how PathologyGAN captures key tissue characteristics such as cancer cell, lymphocyte, or stroma density to provide a structured and interpretable latent space.

We further develop PathologyGAN by introducing an encoder that allows us to find representations of real tissue images. We show evidence that these representations capture meaningful tissue information by illustrating their applicability in three different scenarios: We visualize representations with clinical annotations, we quantify how informative they are by training a linear classifier to predict tissue types, and we demonstrate their suitability on a multiple instance learning (MIL) task, predicting the presence of tumor on whole slide images (WSIs).

Finally, we introduce Histomorphological Phenotype Learning (HPL), a methodology to automatically identify and cluster tissue morphologies through self-supervised learning and community detection. We demonstrate the significance of the clustered tissue morphologies by re-

lating them with cancer types, patient outcomes, cell types, growth patterns, and omics-based immune signatures.

Throughout this thesis, we propose GAN-based and self-supervised learning-based methods for unsupervised learning of cancer tissue representations. We evaluate the relevance of the tissue information enclosed in the representations by correlating them with tissue types or cell type density. In addition, we apply the representations in real-world tasks such as tumor presence or cancer type classification over WSIs. Furthermore, we study the general applicability of the self-supervised learning methods by relating representations with clinically relevant annotations such as patient outcomes or omics-based immune signatures. Overall, this thesis provides examples of the potential of unsupervised learning for cancer tissue representations, delivering a tool not only to alleviate the bottleneck of high-quality annotations but also as a means to further study tissue morphologies through growth patterns and associations with molecular phenotypes.

Contents

Abstract **i**

1 Introduction **1**
1.1 Thesis outline . . . 1
1.2 List of publications . . . 2
1.3 Code and data availability . . . 3
1.4 Deep learning in Histopathology . . . 4
1.5 Datasets . . . 9

2 Background **13**
2.1 Deep Generative Models . . . 13
2.1.1 Variational Auto-Encoders . . . 14
2.1.2 Generative Adversarial Networks . . . 17
2.1.3 Performance Comparison of Deep Generative Models . . . 28
2.2 Self-Supervised Learning . . . 29
2.2.1 Contrastive Learning . . . 30
2.2.2 Distillation . . . 34
2.2.3 Clustering . . . 37
2.2.4 Redundancy Reduction . . . 40
2.2.5 Performance Comparison of Self-Supervised Methods . . . 41

3 PathologyGAN: Learning deep representations of cancer tissue **44**
3.1 Introduction . . . 44
3.2 Methods . . . 45
3.2.1 PathologyGAN . . . 45
3.2.2 Evaluation metric on PathologyGAN . . . 46
3.2.3 Cell type quantification on generated images . . . 47
3.2.4 Tissue type assignation on generated images . . . 47
3.2.5 Pathologists' classification test . . . 48
3.3 Results . . . 49
3.3.1 Image quality analysis . . . 49

3.3.2 Pathologists' results 50
3.3.3 Analysis of latent representations 52
3.3.4 Mapping network and style mixing regularization impact on representations 57
3.3.5 Dataset size impact on representations 57
3.4 Conclusion 62

4 Adversarial learning of cancer tissue representations 64
4.1 Introduction 64
4.2 Methods 65
4.2.1 PathologyGAN and real tissue image representations 65
4.3 Results 66
4.3.1 Visualizations of tissue representations and reconstructions 66
4.3.2 Tissue type classification over tissue representations 67
4.3.3 Multiple Instance Learning on tissue representations 71
4.4 Conclusion 73

5 Histomorphological phenotype learning: a self-supervised methodology for discovery of morphological tissue patterns 75
5.1 Introduction 75
5.2 Methods 76
5.2.1 HPL: Histomorphological phenotype learning 76
5.2.2 Whole slide image processing 78
5.2.3 Self-supervised learning 78
5.2.4 Clustering tile vector representations 79
5.2.5 Whole slide image and patient vector representations 80
5.2.6 Evaluation 81
5.2.7 Cluster characterizations 82
5.2.8 Cluster histological assessment 84
5.3 Results 84
5.3.1 HPL-discovered histomorphological phenotypes can classify lung adenocarcinoma and squamous cell carcinoma 85
5.3.2 HPL identifies prognostic histomorphological phenotypes in lung adenocarcinoma associated with clinical outcomes 89
5.3.3 A systematic association of HPL-discovered patterns with cell types, histological growth patterns, and molecular phenotypes 91
5.4 Conclusion 95

6 Conclusion and discussion **99**

6.1 Future directions 100

Appendices **118**

A PathologyGAN: Learning deep representations of cancer tissue 118

A.1 Hinge vs Relativistic Average Discriminator 118

A.2 Nearest neighbors 119

A.3 Vector operations 120

A.4 Visualization of linear interpolations and vector operations in the latent space. 123

B Adversarial learning of cancer tissue representations 125

B.1 Tissue Reconstructions 125

B.2 Receiver operating characteristic curve and confusion matrix of a logistic regression for tissue type classification 127

C Histomorphological phenotype learning: a self-supervised methodology for discovery of morphological tissue patterns 128

C.1 Methods: Evaluation 128

C.2 Results: Pan-cancer classification 128

C.3 Results: LUAD versus LUSC classification 129

C.4 Results: LUAD survival 132

List of Tables

1.1 Pan-cancer dataset description. 12

2.1 ImageNet supervised performance on self-supervised representations 42
2.2 ImageNet semi-supervised performance on self-supervised representations . . . 42

3.1 PathologyGAN's FID performance evaluation 49
3.2 Pathologists' ratings and kappa scores on PathologyGAN's images 51
3.3 PathologyGAN's FID performance for different training sample sizes. 57

4.1 Tissue type classification over PathologyGAN's tissue representations 71
4.2 Tissue type classification performance comparison. PathologyGAN vs supervised methods 71

List of Figures

1.1 Example of a whole slide image 4
1.2 Example of tissue micro-array 5
1.3 Dataset descriptions of TCGA and NYU cohorts for LUAD overall and recurrence free survival 10

2.1 High level description of *VAE*'s architecture. 15
2.2 Example of disentanglement in VAEs 16
2.3 High level description of a *GAN*'s architecture. 18
2.4 High level description of *InfoGAN*'s architecture. 25
2.5 Example of disentanglement in InfoGAN 25
2.6 High level description of *AAE*'s architecture. 27
2.7 High level description of *BigBiGAN*'s architecture. 27
2.8 High level description of *StyleGAN*'s architecture. 28
2.9 High level description of *ALAE*'s architecture. 28
2.10 Common setup for self-supervised learning models 30
2.11 High level description of *SimCLR*'s architecture. 32
2.12 High level description of *MoCo*'s architecture. 33
2.13 High level description of *Self-Supervised Relational Reasoning* architecture . . 34
2.14 High level description of *BYOL*'s architecture. 35
2.15 High level description of *SimSiam*'s architecture. 36
2.16 High level description of *DeepCluster*'s architecture. 38
2.17 High level description of *SwAV*'s architecture. 39
2.18 High level description of *Barlow Twins*' architecture. 41

3.1 High level description of PathologyGAN's architecture. 46
3.2 Hover-Net cell type identification 47
3.3 Pathologists test example of breast cancer tissue. 48
3.4 Pathologists test example of colorectal cancer tissue. 48
3.5 PathologyGAN's nearest neighbors in Inception-V1 feature space 50
3.6 ROC curve of pathologists' performance over real versus fake classification for PathologyGAN's generated samples 51

3.7 UMAP of PathologyGAN's latent space trained on breast cancer tissue 53
3.8 UMAP of PathologyGAN's latent space trained on colorectal cancer tissue . . . 54
3.9 Linear interpolations in PathologyGAN's latent space w 55
3.10 Linear vector operations in PathologyGAN's latent space w 55
3.11 UMAP of linear interpolations in PathologyGAN's latent space w and associated images 56
3.12 UMAP of linear vector operations in PathologyGAN's latent space w and associated images 56
3.13 Impact of introducing the mapping network and style mixing regularization in PathologyGAN's latent space w for breast cancer tissue. Scatter plot. 58
3.14 Impact of introducing the mapping network and style mixing regularization in PathologyGAN's latent space w for breast cancer tissue. Density plot. 59
3.15 Impact of introducing the mapping network and style mixing regularization in PathologyGAN's latent space w for colorectal cancer tissue. Scatter plot. . . . 60
3.16 Impact of introducing the mapping network and style mixing regularization in PathologyGAN's latent space w for colorectal cancer tissue. Density plot . . . 61
3.17 UMAP of PathologyGAN's latent space trained on a 5K subsample 62

4.1 High level architecture of proposed modification of PathologyGAN. 66
4.2 UMAP of PathologyGAN's and Inception-V1's latent representations of breast cancer tissue images - Neoplastic epithelial cell density 68
4.3 UMAP of PathologyGAN's and Inception-V1's latent representations of breast cancer tissue images - Non-neoplastic epithelial cell density 68
4.4 UMAP of PathologyGAN's and Inception-V1's latent representations of breast cancer tissue images - Inflammatory cell density 69
4.5 UMAP of PathologyGAN's and Inception-V1's latent representations of breast cancer tissue images - Connective cell density 69
4.6 UMAP of PathologyGAN's and Inception-V1's latent representations of colorectal cancer tissue images - Tissue type 70
4.7 Real tissue images and their PathologyGAN's reconstructions 70
4.8 UMAP of PathologyGAN's latent representations of lung cancer tissue 72

5.1 Overview of HPL methodology architecture 77
5.2 Kolmogorov-Smirnov test for cell type over and under-representation 83
5.3 Hypergeometric test for lung adenocarcinoma histological subtype enrichment . 83
5.4 HPL LUAD versus LUSC classification performance and representation analysis 87
5.5 HPL LUAD versus LUSC interpretability behind classification performance . . 88
5.6 HPL LUAD overall survival analysis 91
5.7 HPL LUAD recurrence free survival analysis 92

5.8 HPL LUAD survival analysis and HPC correlations 95
5.9 HPL LUAD survival HPC representations and reference HPCs analysis 96

A.1 Image generation comparison between the Relativistic Average Discriminator and Hinge losses. 118
A.2 Loss comparison between the Relativistic Average Discriminator and Hinge losses 118
A.3 Additional examples: PathologyGAN's nearest neighbors in Inception-V1 feature space 119
A.4 Additional examples: vector operation in breast cancer tissue. 120
A.5 Additional examples: vector operation C and D in colorectal cancer tissue. . . . 121
A.6 Additional examples: vector operation E and F in colorectal cancer tissue. . . . 122
A.7 Additional examples: UMAP of linear interpolations in PathologyGAN's latent space w and associated images in colorectal cancer tissue 123
A.8 Additional examples: UMAP of linear interpolations in PathologyGAN's latent space w and associated images in breast cancer tissue 123
A.9 Additional examples: UMAP of linear vector operations in PathologyGAN's latent space w and associated images in colorectal cancer tissue 124
A.10 Additional examples: UMAP of linear vector operations in PathologyGAN's latent space w and associated images in breast cancer tissue 124
B.1 Additional examples: Real tissue images and their PathologyGAN's reconstructions of breast cancer tissue 125
B.2 Additional examples: Real tissue images and their PathologyGAN's reconstructions of colorectal cancer tissue 126
B.3 ROC curve and confusion matrix of a logistic regression for tissue type classification 127
C.1 HPL Clustering impact performance on 5-fold cross-validation 128
C.2 HPL pan-cancer classification performance and representation analysis 128
C.3 HPL LUAD versus LUSC classification performance per institution 129
C.4 HPL LUAD versus LUSC institution distribution per HPC 130
C.5 HPC TCGA and NYU tile samples for LUAD versus LUSC classification . . . 131
C.6 HPC TCGA and NYU tile samples for LUAD overall survival 132
C.7 HPC TCGA and NYU tile samples for LUAD recurrence free survival 133
C.8 HPC histological assessment of LUAD overall and recurrence free survival . . 134

Acknowledgements

I would like to thank my supervisor, Ke Yuan. Thank you for giving me the opportunity to pursue this research and degree, and thank you for your support, patience, guidance, and mentoring during these last four years.

I would also like to extend my appreciation to my collaborators during this time. In particular, Nicolas Coudray, Anna Yeaton, Aris Tsirigos, and John Le Quesne. Thank you for the opportunity to work together and for the insightful and productive conversations. Working together throughout this period has considerably pushed my work and skills as a researcher.

I would like to express my sincere gratitude to Sydney Williams for her patience and support over the last few years. Thank you for your kindness and encouragement to follow my own personal interests.

Finally, I would like to thank Antia Claudio Quirós, Volga Claudio Quirós, Francisco Adalberto Claudio Vázquez, and Maria del Pilar Quirós Canga. Thank you for providing me with the environment and opportunities so I could pursue this degree.

Declaration

With the exception of chapters 1 and 2 which contain introductory material, all work in this thesis was carried out by the author unless otherwise explicitly stated.

Chapter 1

Introduction

1.1 Thesis outline

This thesis is structured in the following chapters:

- **Chapter 1** introduces the field of histopathology and computational pathology.
- **Chapter 2** presents the background context for unsupervised representation learning with deep generative models and self-supervised learning.
- In **Chapter 3**, we propose PatholgyGAN, a generative adversarial network(GAN) for unsupervised representation learning of cancer tissue images.
- In **Chapter 4**, we improve on PathologyGAN by introducing an encoder that allows us to find representations of real tissue images. In this chapter we further demonstrate the applicability of the tissue representations.
- **Chapter 5** presents Histomorphological Phenotype Learning (HPL), a methodology to automatically identify and cluster tissue morphologies through self-supervised learning and community detection. HPL does not require labels and is interpretable. We demonstrate the significance of the tissue morphologies by relating them with cancer types, patient outcomes, cell types, growth patterns, and omics-based immune signatures.
- **Chapter 6** concludes and discusses the research contributions of this thesis.

1.2 List of publications

The following publications serve as a foundation for the research contributions:

Chapter 3: PathologyGAN: Learning deep representations of cancer tissue.

Adalberto Claudio Quirós, Roderick Murray-Smith, Ke Yuan. Pathology GAN: Learning deep representations of cancer tissue. Proceedings of the Third Conference on Medical Imaging with Deep Learning. PMLR, 2020.

Adalberto Claudio Quirós, Roderick Murray-Smith, Ke Yuan. Pathology GAN: Learning deep representations of cancer tissue. Journal of Machine Learning for Biomedical Imaging. Special Issue: Medical Imaging with Deep Learning, 2020.

Contributions: Adalberto Claudio Quirós designed and executed the experiments, prepared the manuscript, and wrote the corresponding code.

Chapter 4: Adversarial learning of cancer tissue representations.

Adalberto Claudio Quirós, Nicolas Coudray, Anna Yeaton, Wisuwat Sunhem, Roderick Murray-Smith, Aristotelis Tsirigos, Ke Yuan. Adversarial Learning of Cancer Tissue Representations. 24th International Conference on Medical Image Computing and Computer Assisted Intervention. Springer, 2021.

Contributions: Adalberto Claudio Quirós designed and executed the experiments, prepared the manuscript, and wrote the corresponding code.

Chapter 5: Histomorphological phenotype learning: a self-supervised methodology for discovery of morphological tissue patterns.

Adalberto Claudio Quirós, Nicolas Coudray, Anna Yeaton, Xinyu Yang, Luis Chiriboga, Afreen Karimkhan, Navneet Narula, Harvey Pass, Andre L. Moreira, John Le Quesne, Aristotelis Tsirigos, Ke Yuan. Self-supervised learning in non-small cell lung cancer discovers novel morphological clusters linked to patient outcome and molecular phenotypes. 2022.

Contributions: Adalberto Claudio Quirós designed and executed the experiments, prepared the manuscript, and wrote the corresponding code.

1.3 Code and data availability

The code and data for this thesis are available at:

Chapter 3: PathologyGAN: Learning deep representations of cancer tissue.

https://github.com/AdalbertoCq/Pathology-GAN

Chapter 4: Adversarial learning of cancer tissue representations.

https://github.com/AdalbertoCq/Adversarial-learning-of-cancer-tissue-representations

Chapter 5: Histomorphological phenotype learning: a self-supervised methodology for discovery of morphological tissue patterns.

https://github.com/AdalbertoCq/Histomorphological-Phenotype-Learning

1.4 Deep learning in Histopathology

Histopathology is the field of medicine that studies cancer through the analysis of tissue biopsies. The histological process starts with the extraction of a tissue sample and it is followed by tissue staining. During tissue staining, one or more pigments are used to highlight parts of tissue that are relevant for evaluation. The most commonly used are hematoxylin and eosin (H&E) which stain cell nuclei in blue and cytoplasm and connective tissue as pink (**Figure 1.1** and **Figure 1.2**). Tissue staining allows to identify features that are relevant for diagnosis and prognosis such as nuclear cell atypia, gland formation, presence of mitosis, inflammation, or different histological patterns. Pathologists review these tissue sections providing an evaluation of how aggressive the disease is or how likely recurrence could be. However, pathologist interpretation is time-consuming and prone to inter-observer variation, depending on expertise, knowledge, and on the difficulty in characterizing certain tumors or patterns[1–4].

The standardization of digital scanners during the 1990s-2000s allowed to store and analyze tissue biopsies through computers leading to the field of digital pathology. Histopathology images such as whole slide images (WSIs) and tissue micro-arrays (TMAs) are the standard in digital pathology, they make the largest source of data for studying cancer tissue diversity, ranging from morphology of cell types to complex interactions within the tumor microenvironment[5–7]. **Figure 1.1** shows a whole slide image (WSI), which usually captures a large section of tissue. These images can contain not only tumor tissue but also healthy tissue. On the other hand, **Figure 1.2** shows a tissue micro-array (TMA). Contrary to WSIs, TMAs are smaller images and contain a specific section of the tumor that characterize the diagnosis.

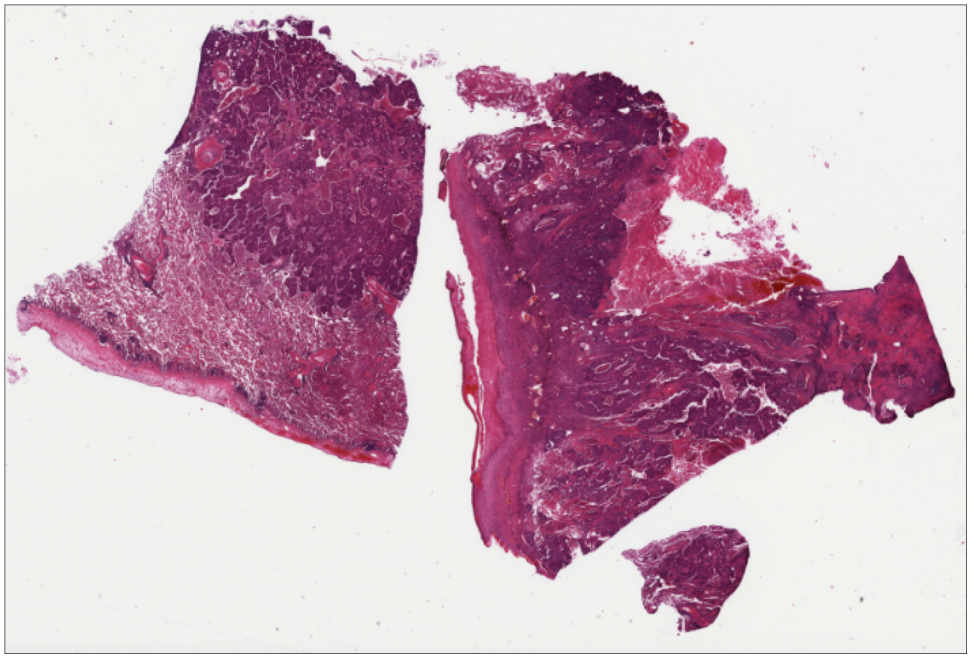


Figure 1.1: Whole slide image (WSI) example of lung adenocarcinoma (LUAD).

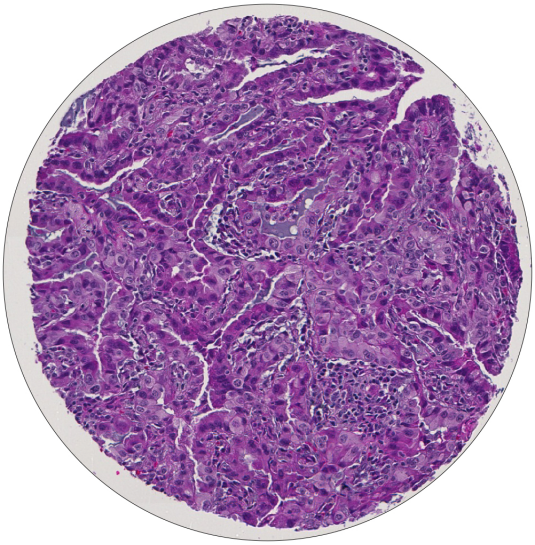


Figure 1.2: Tissue micro-array (TMA) example of lung adenocarcinoma (LUAD).

Digitization of tissue sections allowed to perform more complex computational analysis of these images. Initial milestone publications used traditional computer vision approaches to discover phenotype patterns associated with patient survival[8] or to show how cellular heterogeneity is complementary to genomic profiling[9]. Later on, deep learning applications will see a considerable increase in histopathology, becoming the standard in computational pathology. Ciresan *et al.* [10], Cruz-Roa *et al.*,[11] and Wang *et al.*[12] provided the first deep learning examples, showing that convolutional neural networks (CNN) could rival pathologists on tasks such as mitosis detection or invasive cancer detection. Since then, deep learning applications have expanded to a range of different tasks:

- Cell or nuclei detection, classification, and segmentation.
- Gland, tissue, or tumor segmentation.
- Cancer detection or grading.
- Overall or recurrence survival analysis.
- Stain normalization and transfer
- Genomic or molecular alteration prediction from tissue.

Whole slide images (WSIs) are usually large, within thousands of pixels in height and width depending on the microscopic magnification. This dimensionality is unmanageable by a CNN, the common solution in deep learning applications is to divide the WSI into smaller image sections (tiles) which can then be used by the CNN. This process is usually known as tiling.

An additional pre-processing step is stain normalization, it attempts to reduce color and intensity variability among tissue images. The imaging process from different institutions or laboratories can result in this variability of color and intensity in the WSIs, this is the result of

the staining process where color intensity may vary based on the time a biomarker was applied. In addition, some variations in contrast and brightness can be introduced based on the imaging equipment. Stain normalization aims to reduce this variability since it can prevent models from being able to generalize across different institutions or cohorts, this phenomenon is also known as batch effect. After tiling and stain normalization, histological images are ready to be used in a different range of applications.

We can differentiate between deep learning approaches based on the learning schemes. These include supervised learning, weakly and semi-supervised learning, pre-trained models, unsupervised learning, and self-supervised learning.

Supervised Learning

In fully supervised approaches, we generally have a collection of WSIs with one of the following settings: (A) classify tiles (e.g. cancerous versus healthy), (B) do cell type detection or segmentation on tiles, (C) perform tissue segmentation over WSIs, or (D) predict genetic alterations from tissue.

Examples of tissue tile classification include mitosis detection[10,12], invasive carcinoma detection[11], tumor detection[13–15], metastasis detection[16], cancer type prediction[13], or cancer score prediction[17]. Cell segmentation methods focus on cell type identification and segmentation[18–20]. Within them, it is worth highlighting Hover-Net[19] which has become the standard tool for this task. In addition, tissue segmentation solutions pivot around gland[21], metastasis[22], or lymphocyte infiltration segmentation[23]. Finally, supervised approaches have seen an increased emphasis on relating tissue with genetic alterations[13,24,25] or immunotherapy response[26].

While supervised approaches can lead to accurate models, obtaining clinical annotations is costly in effort and time. Furthermore, it requires expertise from clinicians and even in that case, inter-observer annotations are subject to variability.

Weakly and semi-supervised Learning

Weakly supervised learning predominantly refers to multiple instance learning whereas semi-supervised applies to solutions where a small subset of annotations is used to train the model. Both approaches aim to reduce the dependency on human annotations.

Multiple instance learning (MIL) naturally fits tasks for WSI label prediction. In these cases, we have a bag of instances (i.e. tissue tiles) with an individual label at the WSI level (e.g. presence of metastasis). We further assume that each instance of the bag has an associated label to which we have no access (e.g. presence of metastasis in tile). MIL aims to find which tile instances are correlated with the WSI annotation. MIL has increased in popularity due to its fitness for histopathology data collection where we have large tissue sections with individual annotations at the image or patient level such as genomic mutations, metastasis, cancer detection, or survival/recurrence events. It has been extensively used and adapted to histopathology[27–32].

Additionally, semi-supervised models have been proposed to soften the requiment of large number of annotations in deep learning models. Examples of these applications include cancer detection[33],metastasis and mitosis detection[34,35], and tissue type segmentation[36].

Pre-trained models

Pre-trained models, unsupervised learning, and self-supervised learning applications have seen a steady increase motivated by the idea of reducing the dependency on patholgistists' annotations. By using these approaches, we can find tissue tile representations without the need for any kind of label. In the case of pre-trained models, a common setting is to use a CNN architecture trained on natural images, such as ImageNet, to directly translate tissue tiles into vector representations; simply using the pre-trained CNN as a frozen feature extractor. After this step, we can use tile vector representations to perform any of the previous tasks reviewed in supervised or weakly supervised learning. Examples of these applications include metastasis detection[37,38], pan-cancer classification[39], survival analysis[30], or genomic prediction from tissue[13].

Using a pre-trained model releases the user from the computational burden of training a CNN from scratch. However, these approaches may be limited by how efficiently they extract tissue features. In fact, some recent work shows that other approaches such as self-supervised learning might provide more robust feature extraction from tile images[40].

Unsupervised Learning

Generally, unsupervised learning applications on histopathology focus on generative models. As pre-trained models, they offer an alternative to obtaining data representations without the need for labels or annotations. However, generative models with representation learning properties learn data representations from the features of data themselves, contrary to pre-trained models that were trained with an unrelated kind of data and used as feature extractors. On the other hand, if the generative process is not a requirement, it can be an additional burden compared to pre-trained or self-supervised learning models.

Generative adversarial networks (GANs) have been largely applied in histopathology, especially for their ability on generating high-quality images. Their applications range from nuclei[41,42], tissue[43], or region-of-interest segmentation[44,45], stain transformation[46,47] and normalization[48], tissue sample generation to prevent overfitting[49,50], to large tissue section generation from segmentation masks[51].

Still, generative models for unsupervised representation learning in histopathology have not been as widely applied. These approaches include InfoGANs[52], Sparse Auto-Encoders[53], or Variational Auto-Encoders[54] for representation learning of cells or nuclei. Nonetheless, these models focus on representations of small-size images or cells instead of larger sections of tissue.

Self-supervised Learning

Self-supervised learning has risen as an alternative to pre-trained and unsupervised learning models. These approaches create data representations that can generalize across tasks without labels. Data representations are created by training a model to solve a predetermined task, by doing so the model implicitly learns to extract data features into a representation.

Recently, self-supervised tissue representations have shown to be more robust than pre-trained models, showing better performance across a range of tasks[40,55]. In addition, these methods can be easily combined with other settings, such as weak supervision[56–58], semi-supervised learning[58,58], tissue and nuclei segmentation[58–60], or survival analysis[57,60]. The main advantage of self-supervised representations is their ability to generalize across tasks, making them especially suitable for semi-supervised learning where a small subset of labels can be used to achieve similar performance to using the complete set of them. In addition, since they lack a generative process, the computational burden of these methods is much less than generative models.

Motivation for unsupervised learning of cancer tissue representations

The impact of supervised approaches is limited by three factors. Firstly, high-quality labels are expensive in time and effort, which makes them not easily scalable. Secondly, these methods focus on identifying known histologic patterns on pathology images, fundamentally restricting the discovery of new tissue phenotypes. Thirdly, supervised methods rely on annotations by experts and oftentimes there is significant subjectivity and bias in determining certain histologic patterns. These limitations emphasize the importance of using methods capable of characterizing tissue by the features enclosed in the image and not only correlated with a given label or annotation.

Unsupervised representation learning offers the opportunity to build these tissue representations solely by extracting information directly from the image. They can characterize cellular content and behavior as well as spatial patterns found in the tissue and capture them into representations without the need for annotations and solely from the information contained in the image. These representations can generalize to multiple tasks and are not only correlated with a selected predicted outcome as compared to discriminative models. They can be used across multiple clinical annotations without the need to retrain the model, as would be required by supervised and weakly-supervised end-to-end solutions. These methods can help to provide insight into the relationship between tissue patterns and clinical annotations such as overall and recurrence free survival or their genomic underpinnings. Ultimately providing a tool for pattern discovery and study for pathologists, and a better understanding of the disease.

Throughout this thesis, we propose GAN-based and self-supervised learning-based methods for unsupervised learning of cancer tissue representations. We evaluate the relevance of the tissue information enclosed in the representations by correlating them with tissue types or cell type density. In addition, we apply the representations in real-world tasks such as tumor presence or

cancer type classification over WSIs. Furthermore, we study the general applicability of the self-supervised learning methods by relating representations with clinically relevant annotations such as patient outcomes or omics-based immune signatures. Overall, this thesis provides examples of the potential of unsupervised learning for cancer tissue representations, delivering a tool not only to alleviate the bottleneck of high-quality annotations but also as a means to further study tissue morphologies through growth patterns and clinical annotations.

1.5 Datasets

In this section we describe the datasets used throughout the thesis.

Breast Cancer Dataset

The H&E breast cancer dataset was built from the Netherlands Cancer Institute (NKI) cohort and the Vancouver General Hospital (VGH) cohort with 248 and 328 patients, respectively. Each of them includes TMA images, along with clinical patient data such as survival time, and estrogen-receptor (ER) status. The original TMA images all have a resolution of 1128×720 pixels, and we split each of the images into smaller patches of 224×224, allowing them to overlap by 50%. We also performed data augmentation on these images, a rotation of 90°, and 180°, and vertical and horizontal flipping. We filter out images in which the tissue covers less than 70% of the area and apply stain normalization[61]. In total this process yielded a training set of $249K$ tiles and 460 patients, and a test set of 62K tiles and 116 patients, with no overlapping patients between sets. This dataset is used in **Chapters 3** and **Chapter 4**.

Colorectal Cancer Dataset

The H&E colorectal cancer dataset is a pre-processed dataset from the National Center for Tumor diseases (NCT, Germany)[62]. It provides tissue images of 224×224 resolution with an associated type of tissue label: adipose, background, debris, lymphocytes, mucus, smooth muscle, normal colon mucosa, cancer-associated stroma, and colorectal adenocarcinoma epithelium (tumor). All tissue images are provided with stain normalization already applied[61]. The dataset is divided into a training set of $100K$ tissue patches and 86 patients, and a test set of $7K$ tissue patches and 50 patients, there are no overlapping patients between training and test sets. In **Chapter 4-Section 4.3.2**, we combine the classes stroma and smooth muscle into a class 'simple stroma' as in reference publications[63,64]. We do so in order to compare performance between our approach and other supervised methods. This dataset is used in **Chapters 3** and **Chapter 4**.

Lung Cancer Datasets

We use two different lung cancer datasets. One composed by frozen WSI samples from the Cancer Genome Atlas (TCGA) and used in **Chapter 4-Section 4.3.3**, and another composed by formalin-fixed paraffin-embedded (FFPE) WSI samples from TCGA and New York University (NYU) and used in **Chapter 5**.

The frozen H&E lung cancer dataset contains samples with adenocarcinoma (LUAD), squamous cell carcinoma (LUSC), and normal tissue, composed of 1807 Whole Slide Images (WSIs) of 1184 patients from The Cancer Genome Atlas (TCGA). We make use of the DeepPATH pipeline provided in Coudray *et al.*[13], dividing each WSI into patches of 224×224 and filtering out images with less than 50% tissue in total area, and applying stain normalization[65]. In addi-

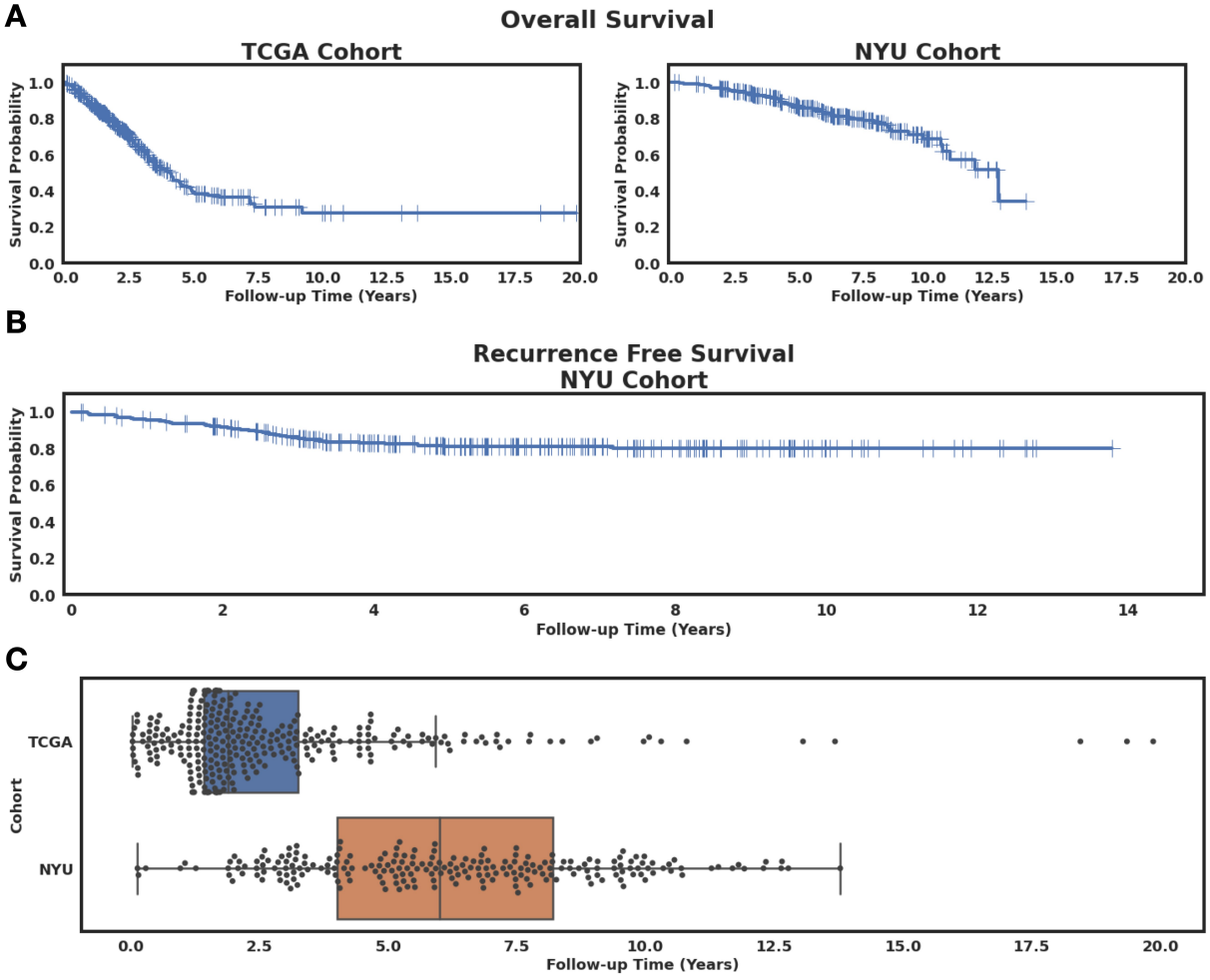


Figure 1.3: Dataset description for lung adenocarcinoma overall and recurrence free survival. A Kaplan-Meier curves of overall survival events of lung adenocarcinoma (LUAD) of The Cancer Genome Atlas (TCGA) and New York University (NYU) cohorts. Censored samples are represented by a cross. **B** Kaplan-Meier curves of recurrence free survival events of LUAD systemic and loco-regional recurrence for the New York University (NYU) cohort. The NYU cohort is composed by three types: systemic, loco-regional, and new primary. We censored new primary cases as we focus on systemic and loco-regional. Censored samples are represented by a cross. **C** Follow-up times from patients of TCGA and NYU cohorts. NYU patients are considerably larger follow-up times allowing a more refined study of recurrence.

tion, we label each slide as tumor and non-tumor depending on the presence of lung cancer in the tissue. Finally, we split the dataset into a training set of $916K$ tissue patches and 666 patients, and a test set of $569K$ tissue patches and 518 patients, with no overlapping patients between both sets.

The FFPE H&E lung cancer dataset uses sample from TCGA and NYU as independent external cohort labeled as NYU_1. The TCGA cohort is composed of 1021 WSIs, 513 of adenocarcinoma (LUAD) and 508 WSIs of squamous cell carcinoma (LUSC). NYU_1 is composed of 138 WSIs, 72 of LUAD and 66 of LUSC and comparable to the one used in Coudray *et al.*[13]. We process WSIs through DeepPATH[13] dividing each WSI into non-overlapping tiles of 224×224 at 5X magnification (with tiles re-sampled to ensure $2.016\mu m$ per pixel), filter out images with less than 60% tissue area, and apply stain normalization[65].

The adenocarcinoma overall survival analysis in **Chapter 5-Section 5.3.2** used the TCGA LUAD cohort and an additional cohort from NYU[5] labeled in this study as NYU_2. The TCGA cohort is composed by 443 patients with a censorhip rate of 0.64 and all stages (Stage 1-4), NYU_2 is composed of 276 patients with a censorship rate of 0.80 and only Stage 1 cases. Further details on both cohorts such as Kaplan-Meier curves and a distribution comparison between follow-up times can be found in the **Figure 1.3**.

The adenocarcinoma recurrence free survival analysis in **Chapter 5-Section 5.3.2** focuses on recurrence types of systemic and loco-regional from NYU_2. It is composed by three types: systemic, loco-regional, and new primary. We censored new primary cases as we focus on systemic and loco-regional. The motivation behind only using the NYU cohort is based on the detailed information for recurrence types and larger follow-up times, allowing a more refined study of recurrence. This dataset is composed by the same 276 patients from NYU_2 used in the overall survival analysis, the censorship rate for recurrence is 0.82. In addition, 70 NYU WSIs have manual annotations from pathologists on histological subtypes (solid, papillary, micropapillary, acinar, and lepidic) allowing individual tile labeling.

Patients from TCGA are also annotated with immune landscape features derived from a immune signature analysis of 33 cancer types published by Thorsson *et al.*[66]. These immune signatures were derived from bulk RNASeq data.

Pan-Cancer Dataset

The FFPE H&E pan-cancer cancer dataset is composed by seven types of cancer, all from TCGA: Breast invasive carcinoma (BRCA), head and neck squamous cell carcinoma (HSNC), kidney renal clear cell carcinoma (KIRC), kidney renal papillary cell carcinoma (KIRP), kidney chromophobe (KICH), lung adenocarcinoma (LUAD) and lung squamous cell carcinoma (LUSC). **Table 1.1** shows the distribution of WSIs and patients per cancer type. We process WSIs through DeepPATH[13] dividing each WSI into non-overlapping tiles of 224×224 at 5X magnification (with tiles re-sampled to ensure $2.016\mu m$ per pixel), filter out images with less than 60% tissue area,

and apply stain normalization[65]. This dataset is used in **Chapters 5**.

Cancer Type	WSIs	Patients
LUAD	516	454
LUSC	508	474
BRCA	1053	1004
KIRC	504	498
KIRP	295	273
KICH	119	107
HSNC	468	448

Table 1.1: Pan-cancer dataset description. The FFPR H&E pan-cancer cancer dataset is composed by seven types of cancer, all from TCGA: Breast invasive carcinoma (BRCA), head and neck squamous cell carcinoma (HSNC), kidney renal clear cell carcinoma (KIRC), kidney renal papillary cell carcinoma (KIRP), kidney chromophobe (KICH), lung adenocarcinoma (LUAD) and lung squamous cell carcinoma (LUSC).

Chapter 2

Background

In the following sections, we will review the background of deep generative models and self-supervised learning, focusing specifically on unsupervised representation learning through these methods. In addition, we will describe the field of histopathology and characterize deep learning applications in cancer tissue imaging.

2.1 Deep Generative Models

Deep generative models are a field of unsupervised learning that focuses on reproducing a data distribution by using neural networks. Given a data distribution and samples $x_i \sim p_{data};\ i = 1, ..., n$, generative models aim to parametrize a distribution p_θ that minimizes the distance between them. We are interested in learning the distribution p_θ for the following:

1. Sample generation: we can obtain new samples by sampling from the learned distribution $x_{new} \sim p_\theta$.

2. Sample density: we can estimate the probability $p_\theta(x)$ of obtaining a sample x.

3. Unsupervised representation learning: we can automatically learn features in data and encode them in some latent variable $z \sim p_\theta(z);\ p_\theta(x) = \int_z p_\theta(x/z)p_\theta(z)dz$.

Deep generative models can be divided into likelihood-based and likelihood-free models. Likelihood-based models explicitly maximize likelihood, these include autoregressive models, variational auto-encoders, normalizing flows, and diffusion models. On the other hand, likelihood-free models implicitly maximize likelihood by sampling from the model's probability distribution. Generative adversarial networks are the representative example of likelihood-free models.

We can further distinguish deep generative models by considering if they are latent variable models or not. Latent variable models consider an unobserved or hidden variable z that conditions the observed variable x which distribution we want to model. In these cases, the generative process is described as in **2.1**. Latent variables z can encode high-level features of

the observed data x, therefore providing unsupervised representation learning properties. Variational auto-encoders (VAEs) and generative adversarial networks (GANs) are exemplary cases of deep learning latent variable models.

$$p_\theta(x) = \int_z p_\theta(x, z)dz = \int_z p_\theta(x/z)p_\theta(z)dz \tag{2.1}$$

In contrast, we have fully observed models where we directly model the distribution of the observed variable x. A representative example of these are autoregressive models. Autoregressive models define the joint probability of the observed variables x as the conditional probabilities through the chain rule (**2.2**) without making any assumption of an underlying latent variable.

$$p_\theta(x) = \prod_{i=1}^{n} p(x_i/x_1, x_2, ..., x_{i-1}) = \prod_{i=1}^{n} p(x_i/x_{<i}) \tag{2.2}$$

We will focus on variational auto-encoders (VAEs) and generative adversarial networks (GANs) since they support unsupervised representation learning. We will especially emphasize on GANs considering that **Chapter 3** and **Chapter 4** study GANs for unsupervised learning of cancer tissue representations.

2.1.1 Variational Auto-Encoders

Variational Auto-Encoders (VAEs)[67] are deep generative models that combine ideas from auto-encoders and variational inference[68]. VAEs explicitly maximize the marginal likelihood $p_\phi(x)$ in order to replicate a given data distribution $p_{data}(x)$ (**2.3**) where z can be refereed to as latent or prior variables:

$$\begin{aligned} argmax_\phi \prod_{i=1}^{N} p_\phi(x_i) &= argmax_\phi \sum_{i=1}^{N} log p_\phi(x_i) \\ p_\phi(x_i) &= \int_z p_\phi(x_i/z)p_\phi(z)dz \end{aligned} \tag{2.3}$$

We are interested in finding the posterior $p_\phi(z/x)$ so we can find useful data representations z and approximate marginal inferences for unseeing data samples x_{new}:

$$p_\phi(z/x) = \frac{p_\phi(x/z)p_\phi(z)}{p_\phi(x)};\ p(x_{new}/x) = \int_z p_\phi(x_{new}/z)p_\phi(z/x)dz \tag{2.4}$$

However, we cannot approximate the marginal likelihood $p_\phi(x_i)$ since solving the integral is computationally intractable. To overcome this limitation, we can define an approximation $q_\theta(z/x)$ of the posterior $p_\phi(z/x)$ and minimize their difference through the Kullback–Leibler divergence D_{KL}, a method to measure similarity between two probability distributions. In traditional variational inference[68] the approximation $q_\theta(z)$ of $p_\phi(z/x)$ has independent parameters θ_i each latent

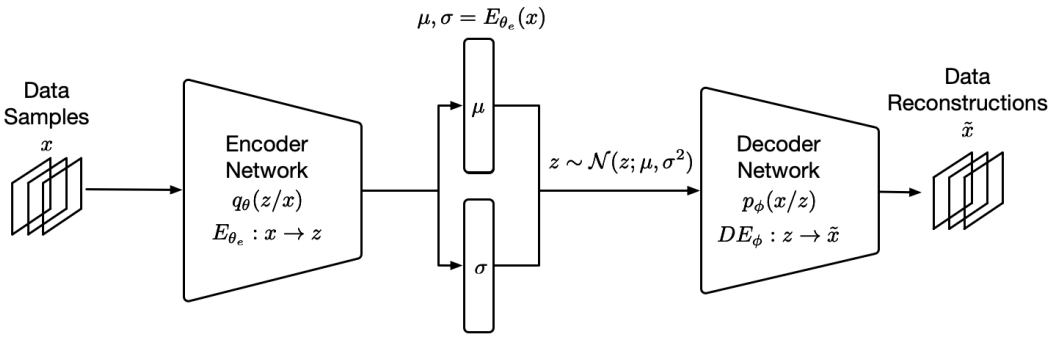


Figure 2.1: High level description of *VAE*'s architecture.

sample z_i. In VAEs, the parameters θ are amortized across all samples x_i[69].

To minimize the Kullback–Leibler divergence D_{KL}, we can further expand its definition and determine which terms do we have access to (**2.5**):

$$\begin{gathered} D_{KL}(q_\theta(z/x)||p(z/x)) = logp_\phi(x) + D_{KL}(q_\theta(z/x)||p_\phi(z)) - \mathbb{E}_{z\sim q_\theta(z/x)} logp_\phi(x/z) \\ \text{where } \mathbb{E}_{z\sim q_\theta(z/x)} logp_\phi(x/z) = \int_z q_\theta(z/x) logp_\phi(x/z) dz \\ logp_\phi(x) = D_{KL}(q_\theta(z/x)||p_\phi(z/x) + \underbrace{\mathbb{E}_{z\sim q_\theta(z/x)} logp_\phi(x/z) - D_{KL}(q_\theta(z/x)||p_\phi(z))}_{\text{ELBO}} \end{gathered} \tag{2.5}$$

As previously mentioned, we do not have access to the marginal likelihood $p_\phi(x)$, also known as evidence. Since the D_{KL} is always greater or equivalent to zero by definition, we can maximize the Evidence Lower Bound (ELBO) in order to maximize the marginal likelihood. This is objective function of the VAE (**2.6**), which can be seen as a reconstruction term and a regularization term. Maximizing the ELBO requires a balance between these two terms. The reconstruction term places density over latent variables that are able to reconstruct the evidence and the regularization term minimizes the difference between the variational approximation $q_\theta(z/x)$ and the assumed prior $p(z)$. The distributions $p_\phi(x/z)$ and $q_\theta(z/x)$ are parametrized as neural networks and optimized through stochastic gradient descent:

$$\mathcal{L}_{VAE} = \mathbb{E}_{x_i\sim p_{data}}[\underbrace{D_{KL}(q_\theta(z/x_i)||p_\phi(z))}_{\text{Latent Regularization}} - \underbrace{\mathbb{E}_{z\sim q_\theta(z/x_i)} logp_\phi(x_i/z)}_{\text{Reconstruction}}] \tag{2.6}$$

In practice, the VAE makes the assumption of modeling the latent or prior distribution $p(z)$ and the posterior approximation $q_\theta(z/x)$ as isotropic Gaussians, and the likelihood $p_\theta(x/z)$ as another isotropic Gaussian (real-valued data) or as a Bernoulli distribution (binary data) (**2.7**).

Figure 2.1 shows a high level representation of a VAE.

$$\begin{aligned} p(z) &= \mathcal{N}(0, \mathbb{I}) \\ q_\theta(z/x) &= \mathcal{N}(z; \mu_\theta(x), \sigma_\theta(x)^2 \mathbb{I}) \\ p_\phi(x/z) &= \mathcal{N}(x; \mu_\phi(z), \sigma_\phi(z)^2 \mathbb{I}) \end{aligned} \tag{2.7}$$

VAEs and representation learning

VAEs inherently have representation learning properties, where data features are automatically encoded in the latent representation $z = (z_1, z_2, ..., z_D)$. VAEs provide a particular case of representation learning called disentanglement. Although there has been different attempts to formally define disentanglement[70–73], it can be generally understood as an orthogonal space where variation in some dimensions z_j translates to variation of a semantic data feature. **Figure 2.2**[74] shows an example of disentanglement in VAEs, the first two columns correspond to the original image and its reconstruction; each of the remaining rows corresponds to a sweep of values on the dimension z_j while maintaining the remaining dimensions constant $z_{\setminus j}$. We can see that each dimension z_j correspond to a semantic feature in the image.

β-VAE[75,76] takes a further look into VAEs and disentangled representations. It focuses on the latent regularization term as a source of disentanglement and adds a weight hyperparameter β that further encourages it (**2.8**). However, larger penalization of the latent regularization term

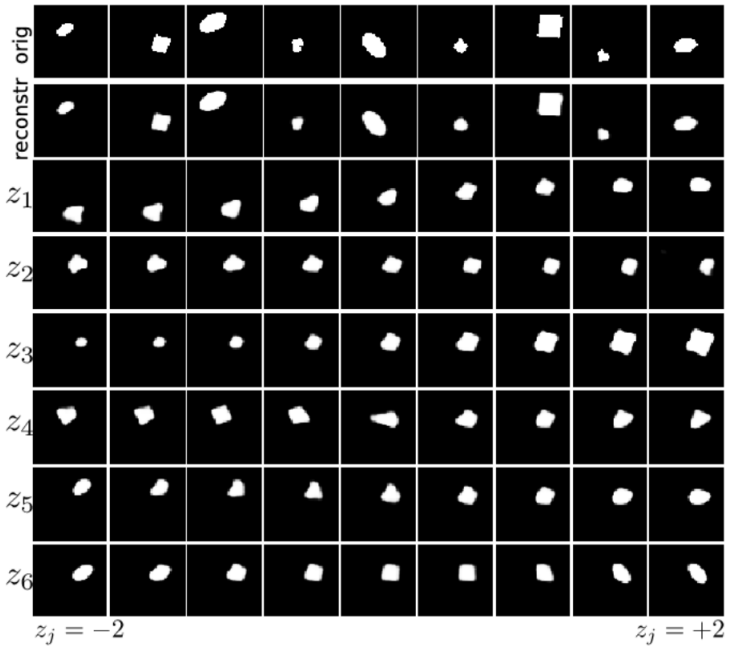


Figure 2.2: Example of disentanglement in β-VAE[75]. The first two rows correspond to the original image and its reconstruction; each of the remaining rows corresponds to sweep values on the dimension z_j while maintaining the remaining constant $z_{\setminus j}$. We can see that each dimension z_j corresponds to a semantic feature in the image. Figure taken from the publication *Disentangling by Factorising*[74]

leads to poorer reconstructions:

$$\mathcal{L}_{\beta\text{-VAE}} = \beta \underbrace{D_{KL}(q_\theta(z/x)||p_\phi(z))}_{\text{Latent Regularization}} - \underbrace{\mathbb{E}_{z\sim q_\theta(z/x)} log p_\phi(x/z)}_{\text{Reconstruction}};\ \beta > 1 \tag{2.8}$$

Hoffman *et al.* provides a further study on the latent regularization term, showing that it can be decomposed into the mutual information between latent and data variables $I(x;z)$ and a marginal latent regularization:

$$D_{KL}(q_\theta(z/x)||p_\phi(z)) = I_{x,z\sim p_{data}(x)q_\theta(z|x)}(x;z) + D_{KL}(q_\theta(z)||p_\phi(z)) \tag{2.9}$$

As we will cover in the GAN section, the mutual information term $I(x;z)$ is equal to the one used in the Information Maximization GAN (InfoGAN). Maximizing this term in the GAN model leads to better disentanglement results. This evidence supports the reasoning that penalizing both terms equally leads to poor reconstructions. Instead, we can penalize the marginal latent regularization that encourages independence between dimensions z_j and therefore disentanglement.

Factor-VAE[74] adds a term that directly enforces independence between dimensions in the latent distribution. This is know as the Total Correlation (TC) term[77] (**2.10**):

$$\mathcal{L}_{\text{Factor-VAE}} = -\mathbb{E}_{z\sim q_\theta(z/x)} log p_\phi(x/z) + D_{KL}(q_\theta(z/x)||p_\phi(z)) + \gamma D_{KL}(q(z)||\prod_j q(z_j)) \tag{2.10}$$

β-**TC-VAE**[78] unifies all previous ideas under one framework. Chen *et al.* analyze how penalty on the Total Correlation term induces disentangled representations by forcing the model to find statistically independent factors in the data distribution:

$$\begin{aligned}\mathcal{L}_{\beta\text{-TCVAE}} = -\mathbb{E}_{z\sim q_\theta(z/x)} log p_\phi(x/z) + \underbrace{I_{x,z\sim p_{data}(x)q_\theta(z|x)}(x;z)}_{\text{Mutual information}} \\ +\beta \underbrace{D_{KL}(q(z)||\prod_j q(z_j))}_{\text{Total Correlation}} + \underbrace{\sum_j D_{KL}(q(z_j)||p(z_j))}_{\text{Dimension-wise KL divergence}}\end{aligned} \tag{2.11}$$

2.1.2 Generative Adversarial Networks

Generative Adversarial Networks (GANs)[79,80] are deep learning models that aim to reproduce a target data distribution p_{data}. GANs are built from two building blocks, a generator G_{θ_g} and a discriminator D_{θ_d}.

The generator G_{θ_g} maps latent vectors z to samples $\tilde{x}$ that resemble data samples x. The generator's distribution $p_g(x)$ tries to implicitly approximate $p_{data}(x)$ through a map from the latent distribution $z \sim p_z(z)$. The discriminator D_{θ_d} distinguishes between real x and fake $\tilde{x}$

samples. It outputs the probability of a sample being real rather than fake, labeling zero for fake and one for real. **Figure 2.3** shows a high level representation of a GAN.

The GAN's loss function is defined as a min-max competition between the generator G_{θ_g} and the discriminator D_{θ_d}. While the generator will aim to minimize the loss by creating samples that resemble the real ones, the discriminator will try to maximize it by correctly classifying real and fake samples (**2.12**):

$$\min_{G_{\theta_g}} \max_{D_{\theta_d}} V(D_{\theta_d}, G_{\theta_g}) = \mathbb{E}_{x \sim p_{data}(x)}[\log D_{\theta_d}(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D_{\theta_d}(G_{\theta_g}(z)))] \tag{2.12}$$

The generator G_{θ_g} and discriminator D_{θ_d} are optimized through stochastic gradient descent. It is relevant to mention that the gradient updates on the generator depend on the discriminator (**2.13**):

$$\begin{gathered} \nabla_{\theta_g} \frac{1}{N} \sum_{i=1}^{N} log(1 - D_{\theta_d}(G_{\theta_g}(z_i)) \\ \nabla_{\theta_d} \frac{1}{N} \sum_{i=1}^{N} [log(D_{\theta_d}(x_i)) + log(1 - D_{\theta_d}(G_{\theta_g}(z_i)))] \end{gathered} \tag{2.13}$$

Algorithm 1 describes the GAN optimization process. The discriminator is trained for k gradient updates for each generator update. This approach is motivated by the idea of providing less noisy updates to the generator, allowing the discriminator to firstly optimize on differentiating real and fake samples. This process is repeated for a number of training iterations until the GAN finally converges.

Finding the optimal point for the min-max optimization of the loss $V(D_{\theta_d}, G_{\theta_g})$ gives a theoretical background on how GANs learn to reproduce the original data distribution p_{data} and their relation to other deep generative models.

Assuming a fixed generator G_{θ_g}, we can find the optimal discriminator value $D^*_{\theta_d}$ that maxi-

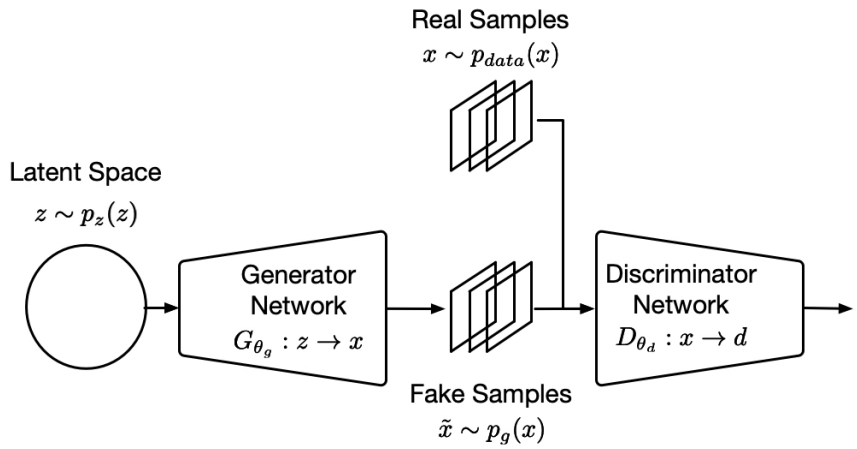


Figure 2.3: High level description of a *GAN*'s architecture.

Algorithm 1 GAN training algorithm. k is a hyperparameter that denotes the number of gradient updates of the discriminator for one of the generator. N denotes the batch size.

for number of training iterations **do**
  for k **do**
    Sample latent vectors $z \sim p_z(z)$
    Sample real data samples $x \sim p_{data}(x)$
    Update discriminator's weights:

$$\nabla_{\theta_d} \frac{1}{N} \sum_{i=1}^{N} [log(D_{\theta_d}(x_i)) + log(1 - D_{\theta_d}(G_{\theta_g}(z_i)))]$$

  end for
  Sample latent vectors $z \sim p_z(z)$
  Update generator's weights:

$$\nabla_{\theta_g} \frac{1}{N} \sum_{i=1}^{N} log(1 - D_{\theta_d}(G_{\theta_g}(z_i)))$$

end for

mizes the loss:

$$\begin{gathered} V(D,G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))] = \\ \int_x [p_{data}(x) log(D(x)) + p_g(x) log(1 - D(x))] dx \\ \frac{\partial V(D,G)}{\partial D} = 0 \rightarrow p_{data}(x) \frac{1}{D(x)} - p_g \frac{1}{1 - D(x)} = 0 \\ D^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} \end{gathered} \tag{2.14}$$

We can now define the loss function $V(D_{\theta_d}, G_{\theta_g})$ given the optimal value for the discriminator $D^*_{\theta_d}$. In **2.15**, D_{KL} denotes Kullback–Leibler divergence and D_{JS} denotes the Jensen-Shannon divergence, both of them are different methods of measuring similarity between two probability distributions:

$$\begin{gathered} V(D^*,G) = \int_x [p_{data}(x) log(D^*(x)) + p_g(x) log(1 - D^*(x))] dx = \\ \int_x [p_{data}(x)(-2log(2)) + p_g(x)(-2log(2))] dx + \\ \int_x [p_{data}(x) log(\frac{2p_{data}(x)}{p_{data}(x) + p_g(x)}) + p_g(x) log(\frac{2p_{data}(x)}{p_{data}(x) + p_g(x)})] dx = \\ -2log2 + 2[\frac{1}{2} D_{KL}(p_{data} || \frac{p_{data}(x) + p_g(x)}{2}) + \frac{1}{2} D_{KL}(p_g || \frac{p_{data}(x) + p_g(x)}{2})] = \\ -2log2 + D_{JS}(p_{data} || p_g) \end{gathered} \tag{2.15}$$

Minimizing the loss function with the optimal discriminator is, in fact, equivalent to maximizing the similarity between the real and generated distributions. Its optimal point $V(D^*, G^*) = -2log2$ happens when the generated distribution matches the original target distribution $p_g = p_{data}$.

We can now draw relationships between GANs and other deep generative models that maximize likelihood. Maximum likelihood models can be seen as minimizing the Kullback–Leibler divergence between a target data distribution p_{data} and the model's p_ϕ:

$$\begin{gathered}\phi_* = argmax_\phi \prod_{i=1}^{M} p_\phi(x_i) = argmax_\phi log \prod_{i=1}^{M} p_\phi(x_i) = argmax_\phi \sum_{i=1}^{M} logp_\phi(x_i) \\ lim_{m\to\infty} argmax_\phi \sum_{i=1}^{M} logp_\phi(x_i) \simeq argmin_\phi - \int_x p_{data}(x) logp_\phi(x) dx = \\ argmin_\phi [\int_x p_{data}(x) logp_{data}(x) - \int_x p_{data}(x) logp_\phi(x)] dx = \\ argmin_\phi D_{KL}(p_{data}(x)||p_\phi(x))\end{gathered} \tag{2.16}$$

The Kullback–Leibler divergence is not a symmetrical function $D_{KL}(p_{data}(x)||p_\phi(x)) \neq D_{KL}(p_\phi(x)||p_{data}(x))$ in contrast to the Jensen-Shannon that is symmetrical. By minimizing $D_{KL}(p_{data}(x)||p_\phi(x))$, maximum likelihood models penalize cases where the model's density does not cover the data distribution's ($\forall x \ / \ p_{data}(x) > 0$ and $p_\phi \to 0$). This ensures that the model's approximation covers the diversity of samples of the original data. In contrast, the Jensen-Shannon D_{JS} is composed of $D_{KL}(p_{data}(x)||p_\phi(x))$ and $D_{KL}(p_\phi(x)||p_{data}(x))$, ensuring sample diversity and the probability of obtaining those samples.

Training Issues

In practice, GAN training is non-trivial and can lead to shortcomings when reproducing the original data distribution. The min-max competition between the generator G_{θ_g} and the discriminator D_{θ_d} requires balanced training between them where the discriminator is required to provide stable gradients to the generator. The training process can destabilize causing the GAN to not converge and leading to poor quality of samples or lack of generalization in the generator's distribution p_g.

GAN training difficulties can be summarized into three ideas: vanishing gradients, unstable gradients, and collapse[81,82]. Improving and stabilizing the training process has been an active field of research after the model's publication.

Vanishing gradients. As we covered in the introduction to GANs, the gradients to the generator are dependent on the discriminator as well (**2.17**):

$$\nabla_{\theta_g} \frac{1}{N} \sum_{i=1}^{N} log(1 - D_{\theta_d}(G_{\theta_g}(z_i)) \tag{2.17}$$

At the beginning of training, it is easy for the discriminator to correctly classify real and fake samples, causing the discriminator to quickly saturate and provide small gradients. *Arjovsky et at.*[81] provided the theoretical and experimental background for the vanishing of the generator's gradients when the discriminator approaches its optimal value.

Unstable gradients. As an alternative to the vanishing gradients, a new heuristic loss function was proposed for the generator[80] (**2.18**). In this case, the generator maximizes the probability of the discriminator being mistaken. However, the GAN optimization is no longer a zero-sum game and does not provide the same theoretical guarantees:

$$\begin{aligned} \max_{D_{\theta_d}} V_{Dis}(D_{\theta_d}, G_{\theta_g}) &= \mathbb{E}_{x\sim p_{data}(x)}[\log D_{\theta_d}(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D_{\theta_d}(G_{\theta_g}(z)))] \\ \max_{G_{\theta_g}} V_{Gen}(D_{\theta_d}, G_{\theta_g}) &= \mathbb{E}_{z\sim p_z(z)}[\log(D_{\theta_d}(G_{\theta_g}(z)))] \end{aligned} \tag{2.18}$$

Although it solves the vanishing gradient issue, it introduces a two new problems[81]. First, the gradients to the generator became noisy and unstable. Second, the learning objective becomes counter intuitive. In **2.19**, we can see how the gradients on generator loss can be expressed as the difference of the Kullback–Leibler divergence $D_{KL}(p_{data}(x)||p_\phi(x))$ and Jensen-Shannon divergence $D_{JS}(p_{data}||p_g)$; translating into generating high quality images but dropping sample diversity:

$$\nabla_{\theta_g}\mathbb{E}_{z\sim p_z(z)}[-log(D^*_{\theta_d}(G_{\theta_g}(z)))] = \nabla_{\theta_g}[D_{KL}(p_{data}(x)||p_g(x)) - 2D_{JS}(p_{data}||p_g)] \tag{2.19}$$

Collapse. GAN training also requires a balance in capacity between the generator and discriminator. If the generator has a larger capacity than the discriminator, the GAN is prone to collapse. Collapse or mode collapse is the process in which the generator specializes in maximizing the discriminator misclassification instead of learning the target data distribution. Generally, this results in the generator dropping some sample diversity or producing samples that do not resemble the original data[82].

Evaluation

GAN evaluation metrics quantify the model's ability to learn and reproduce the original data distribution. Their goal is to measure the similarity in quality and diversity between real $p_{data}(x)$ and fake data $p_g(x)$ distributions. We will particularly focus on the Fréchet Inception Distance (FID)[83] since it has become the standard metric to measure GAN's performance[84–87]. Alternative metrics include the Inception Score[88], Kernel Inception Distance (KID)[89], or 1-Nearest Neighbor (1-NN)[90]; although these methods are not as predominant or have shown not to be as reliable[83,91].

The FID score measures the similarity between real $p_{data}(x)$ and fake data distributions $p_g(x)$. Instead of doing so in the pixel space, it uses a pre-trained ImageNet Inception Network[92,93] to extract features of each image, reducing sample dimensionality and obtaining vision-relevant

features. Feature samples are then fitted into a multivariate Gaussian distributions, obtaining real $\mathcal{N}(\mu_{data}, \Sigma_{data})$ and fake $\mathcal{N}(\mu_g, \Sigma_g)$ feature distributions. Finally, it uses the Fréchet distance[94] to measure the difference between the two distributions (**2.20**):

$$FID = d^2((\mu_{data}, \Sigma_{data}), (\mu_g, \Sigma_g)) = ||\mu_{data} - \mu_g||_2^2 + Tr(\Sigma_{data} + \Sigma_g - 2(\Sigma_{data}\Sigma_g)^{1/2}) \quad (2.20)$$

The standard procedure to evaluate GAN performance is to first divide the original dataset into training and test sets. The training set is then used to train the GAN and the test set is used to evaluate the model's performance. After training, a given number of samples are generated from the GAN (e.g. $10,000$) and the same amount are taken from the test set. Finally, these two sample sets are used to compute the FID score.

Milestone models

Since its introduction, modeling distributions of images has become the mainstream application for GANs, firstly proposed by *Radford at al.*[95]. In this section, we will cover some of the milestone models which can be divided into the following trends:

- Optimization objectives and training stabilization.
- Image resolution and variety scalability.
- Representation learning properties.

Optimization objectives and training stabilization

Initial research on GANs focused on overcoming collapse and training instability. In general, these models propose more efficient methods to bound gradient updates and to measure differences between real and generated distributions. Milestone models include the Relativistic GAN, Wasserstein GAN, and Spectral Normalization GAN.

Relativistic GANs (RGANs)[96] introduce a simple yet effective change that stabilizes training and improves convergence. Jolicoeur-Martineau *et al.* argue that the probability of real samples $D_{\theta_d}(x)$ should decrease as the probability of fake samples $D_{\theta_d}(\tilde{x})$ being real increases. This reasoning is motivated by the fact that the Jensen-Shannon divergence is minimized when $D_{\theta_d}(x) = D_{\theta_d}(\tilde{x}) = \frac{1}{2}$ (**2.21**):

$$\begin{gathered} D_{JS}(p_{data}||p_g) = -2log2 + \mathbb{E}_{x\sim p_{data}(x)}[log(D_{\theta_d}(x))] + \mathbb{E}_{z\sim p_z(z)}[log(1 - D_{\theta_d}(\tilde{x}))] \\ \text{where } \tilde{x} = G_{\theta_g}(z) \end{gathered} \quad (2.21)$$

In RGAN, the discriminator measures the probability of the real samples being more realistic than the fake ones $D(x, \tilde{x}) = \sigma(C(x) - C(\tilde{x}))$. When the features of real and fake samples are similar the probability approaches $D(x, \tilde{x}) = \frac{1}{2}$, minimizing the Jensen-Shannon divergence. **2.22**

shows RGAN's loss function for the generator and discriminator, where σ denotes the sigmoid function and $C(x)$ denotes a neural network:

$$\begin{gathered}\min_{D_{\theta_d}} -\mathbb{E}_{x\sim p_{data}(x),z\sim p_z(z)}[log(\sigma(C(x)-C(\tilde{x}))] \\ \min_{G_{\theta_g}} -\mathbb{E}_{x\sim p_{data}(x),z\sim p_z(z)}[log(\sigma(C(\tilde{x})-C(x))] \\ \text{where } \tilde{x}=G_{\theta_g}(z)\end{gathered} \tag{2.22}$$

Wasserstein GANs (WGANs)[97,98] propose a new loss function based on optimal transport. The new loss function stabilizes training and provides a better measure of distance between distributions. Optimal transport studies the most efficient manner of moving density so one distribution would become another targeted one. In particular, WGANs use the Wasserstein distance, which finds the infimum transportation plan to transform the distribution $p_g(x)$ into $p_{data}(x)$. The discriminator is used as a critic function that helps to determine the distance between distributions and it is required to be 1-Lipschitz continuous. In practice, this requirement translates into bounding the discriminator's gradients with penalties. **2.23** shows WGAN's loss, where λ denotes the gradient penalty coefficient and ϵ denotes a hyperparameter to determine the mixing of the real and fake samples:

$$\begin{gathered}\min_{G_{\theta_g}}\max_{D_{\theta_d}\in 1-Lip}\mathbb{E}_{x\sim p_r}[D_{\theta_d}(x)]-\mathbb{E}_{z\sim p_g(z)}[D_{\theta_d}(G_{\theta_g}(z))]+\lambda\mathbb{E}_{\hat{x}}\left[\left(\left\|\nabla_{\hat{x}}D(\hat{x})\right\|_2-1\right)^2\right] \\ \hat{x}=\epsilon x+(1-\epsilon)\tilde{x};\ \tilde{x}=G(z) \text{ and } x\sim p_{data}(x)\end{gathered} \tag{2.23}$$

Spectral Normalization GAN (SNGAN)[86] also focuses on Lipschitz continuity to bound gradient updates and stabilize training. Lipschitz continuity can be guaranteed by normalizing weights W at each layer of the neural network. Weight normalization consists on dividing the weight matrix by its largest singular value $\sigma(W)$ (**2.24**). At each iteration, SNGAN normalizes its weights and applies the gradient updates after the fact:

$$\begin{gathered}\bar{W}_{\text{SN}}(W) := W/\sigma(W) \\ \sigma(\bar{W}_{\text{SN}}(W))=1.\end{gathered} \tag{2.24}$$

SNGAN uses the Hinge loss (**2.25**), which is equivalent to minimizing the Kullback–Leibler divergence between the generator's and data distributions $D_{KL}(p_g||p_{data})$. Minimizing $D_{KL}(p_g||p_{data})$ ensures that the GAN covers the original sample diversity:

$$\begin{gathered}\min_{D_{\theta_d}}\mathbb{E}_{x\sim p_{data}(x)}[log\, D_{\theta_d}(x)]+\mathbb{E}_{z\sim p_z(z)}[log(1-D_{\theta_d}(G_{\theta_g}(z)))] \\ \min_{G_{\theta_g}} -\mathbb{E}_{z\sim p_z(z)}[D_{\theta_d}(G_{\theta_g}(z))]\end{gathered} \tag{2.25}$$

Image resolution and variation scalability

Stabilizing training enabled to develop deeper models, achieving higher image resolutions and using bigger datasets. Representative examples of these models include the Self-Attention GAN, BigGAN, ProGAN, and Multi-Scale Gradients GAN.

Self-Attention GANs (SAGANs)[87] leverage the improvements introduced by SNGAN. It applies spectral normalization not only on the discriminator but also on the generator and introduces self-attention layers in both architectures. The convolutional self-attention mechanism allows each pixel to measure the relative spatial relevance to other pixels, improving image quality generation. SAGAN was the first GAN to converge on ImageNet[92] without conditioning.

BigGAN[84] built on the SAGAN model. It established a baseline architecture for deeper GANs that can be trained on large datasets such as ImageNet. It also provided an empirical study showing that larger batch sizes and feature map dimensionality improve FID results. In addition, the authors suggest including orthogonal regularization on the weights to limit gradient updates. **2.26** shows the orthogonal regularization of a weight matrix W, where $||.||_F$ denotes the Frobenious norm and β denotes the hyperparameter for the penalty coefficient:

$$R_\beta(W) = \beta \left\| W^\top W \odot (\mathbf{1} - I) \right\|_{\mathrm{F}}^2 \tag{2.26}$$

Progressive Growing of GANs (ProGANs) [99] was the first model to achieve image resolutions of 1024×1024. This model proposes an iterative training process where the GAN is progressively trained at $2\times$ multiple resolutions (e.g. $4, 8, 16, 32, 64, 128, 256, 512$). Although it allows to achieve higher resolutions, the training process is costly. It requires multiple training iterations.

Multi-Scale Gradients GANs (MSG-GANs)[100] will later solve the limitation of progressive training and still achieve high-resolution images. MSG-GANs introduce bypass paths between the generator and discriminator at different convolutional layers. The bypass paths enable gradients to flow at different resolutions from the discriminator to the generator. This architecture change removes the need for iterative training.

GANs and representation learning

Like VAEs, GANs can also offer representation learning properties in their latent distributions $p_z(z)$. These models put constrains on the latent distribution or introduce architectural changes so representations z relate to features in the generated data $\tilde{x}$. Milestone models include the Adversarial Autoencoder, InfoGAN, StyleGAN, ALAE, or BiBigGAN.

Disentanglement and GANs. Contrary to VAEs, GANs do not inherently show representation learning properties and when they do, most of these models do not show disentanglement. As previously introduced, disentanglement is understood as an orthogonal space where variation in some vector dimensions translates to variation of a semantic data feature.

Information Maximization GANs (InfoGANs)[101] are the exemplary GAN model for disentanglement. InfoGANs introduce representation learning by maximizing the information between latent codes c and generated samples $\tilde{x} = G_{\theta_g}(z, c)$ (**Figure 2.4**). Directly maximizing $I(c, \tilde{x})$ is not possible since we do not have access to the true posterior $p(c/x)$. Instead, InfoGAN defines a variational lower bound $L_I(G_{\theta_g}, Q_{\theta_q})$ through the approximation of the posterior Q_{θ_q}. We can maximize $L_I(G_{\theta_g}, Q_{\theta_q})$ and implicitly maximize $I(c, \tilde{x})$. **2.27** shows the InfoGAN's loss where $V(G_{\theta_g}, D_{\theta_d})$ denotes the original GAN loss and λ denotes a hyperparameter coefficient

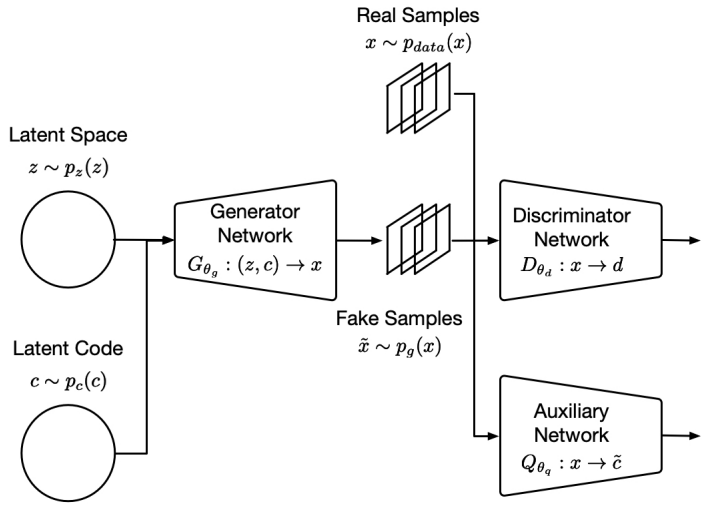
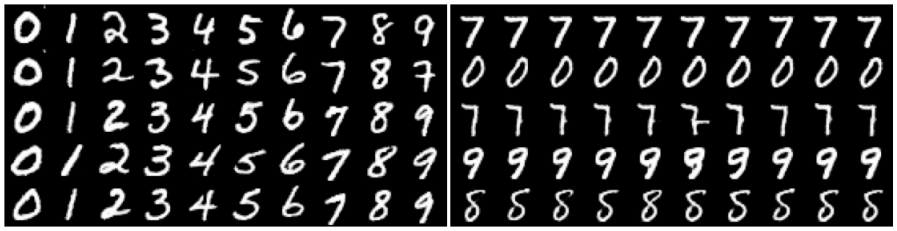
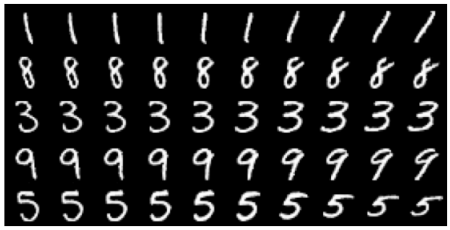


Figure 2.4: High level description of *InfoGAN*'s architecture.

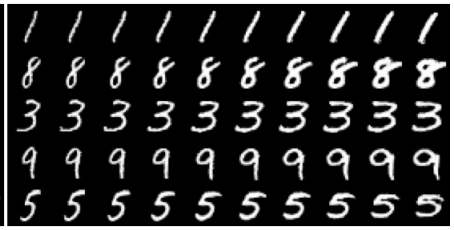


(a) Varying c_1 on InfoGAN (Digit type)

(b) Varying c_1 on regular GAN (No clear meaning)



(c) Varying c_2 from -2 to 2 on InfoGAN (Rotation)



(d) Varying c_3 from -2 to 2 on InfoGAN (Width)

Figure 2.5: Example of disentanglement in InfoGAN[101]. In these examples one dimension values (c_i) are changed while the rest remain constant (c_π)., five different example of this in each subfigure (a,b,c,d). Variations in c_1 correspond to transformations in digit type (a), variations in c_2 correspond to rotation (c), and variations on c_3 to width. In addition, we can see that performing the same operations a regular GAN does not show disentanglement (b). Figure taken from the publication *InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets*[101]

for the variational regularization L_I:

$$\begin{gathered}\min_{G_{\theta_g}, Q_{\theta_q}} \max_{D_{\theta_d}} V(G_{\theta_g}, D_{\theta_d}) - \lambda L_I(G_{\theta_g}, Q_{\theta_q}) \\ L_I(G_{\theta_g}, Q_{\theta_q}) = \mathbb{E}_{c \sim p_c(c)}[log\tilde{c}] + H(c) \;\leq\; I(c, \tilde{x}) \\ \text{where } \tilde{x} \sim G_{\theta_g}(z, c) \text{ and } \tilde{c} = Q_{\theta_q}(\tilde{x})\end{gathered} \tag{2.27}$$

As covered in **Section 2.1.1** and shown by Hoffman *et al.* in VAEs, maximizing mutual information between images and latent codes induces disentanglement in the latent space. **Figure 2.5** shows an example of disentanglement in InfoGANs, variations on different dimensions of the latent codes translate into high-level features in the images such as digit type, rotation, or width. For comparison, we can see how a regular GAN does not the same effect when values changes are introduced in one dimension c_i while the remain are constant (c_π).

The following GAN milestone models hold representation learning in their latent spaces. However, they lack of disentanglement properties.

Adversarial Autoencoders (AAEs)[102] use adversarial learning to regularize the latent distribution and enforce representation learning (**Figure 2.6**). Similarly to the Variational Auto-Encoder (VAE), it uses an auto-encoder where each real sample x is mapped to a latent representation $z \sim \mathcal{N}(\mu(x), \sigma^2(x))$. Instead of using the Kullback–Leibler divergence, it uses the Jensen-Shannon divergence. In this case, the representation's distribution is implicitly regularized by minimizing the difference against an isotropic Gaussian $\mathcal{N}(0, \mathbb{I})$ (**2.28**):

$$\begin{gathered}\min_{E_{\theta_e}, DE_{\theta_{de}}} \underbrace{\mathbb{E}_{x \sim p_{data}}[\mathbb{E}_{z \sim \mathcal{N}(\mu, \sigma^2)}[-log(\tilde{x})]]}_{\text{Reconstruction}} + \underbrace{\mathbb{E}_{z \sim \mathcal{N}(0, \mathbb{I})}[\log D_{\theta_d}(z)] + \mathbb{E}_{z \sim \mathcal{N}(\mu, \sigma^2)}[\log(1 - D_{\theta_d}(z))]}_{\text{Adversarial}} \\ \min_{D_{\theta_d}} \underbrace{\mathbb{E}_{x \sim p_{data}}[\mathbb{E}_{z \sim \mathcal{N}(\mu, \sigma^2)}[-log(\tilde{x})]]}_{\text{Reconstruction}} - \underbrace{\mathbb{E}_{z \sim \mathcal{N}(0, \mathbb{I})}[\log D_{\theta_d}(z)] - \mathbb{E}_{z \sim \mathcal{N}(\mu, \sigma^2)}[\log(1 - D_{\theta_d}(z))]}_{\text{Adversarial}} \\ \text{where } \mu, \sigma = E_{\theta_e}(x) \text{ and } \tilde{x} = DE_{\theta de}(z)\end{gathered} \tag{2.28}$$

Big Bidirectional GANs (BigBiGANs)[103, 104] propose to jointly distinguish latent z and data x samples between real and fake. The model introduces an encoder E_{θ_e} that finds the latent $\tilde{z}$ for each real data x and modifies the discriminator to take in pairs of representations and data (**Figure 2.7**). It also uses the Hinge loss as a learning objective (**2.29**). Interestingly, jointly distinguishing latent and data samples introduces representation learning in the latent distribution $z \sim p_z(z)$ and

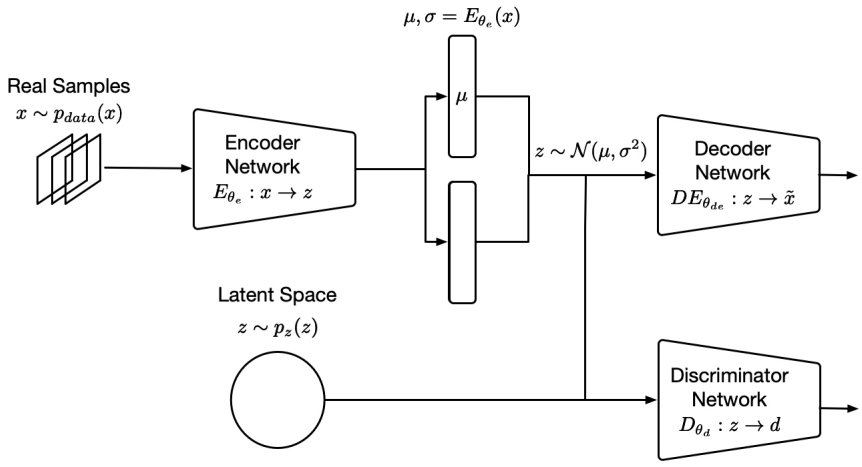


Figure 2.6: High level description of *AAE*'s architecture.

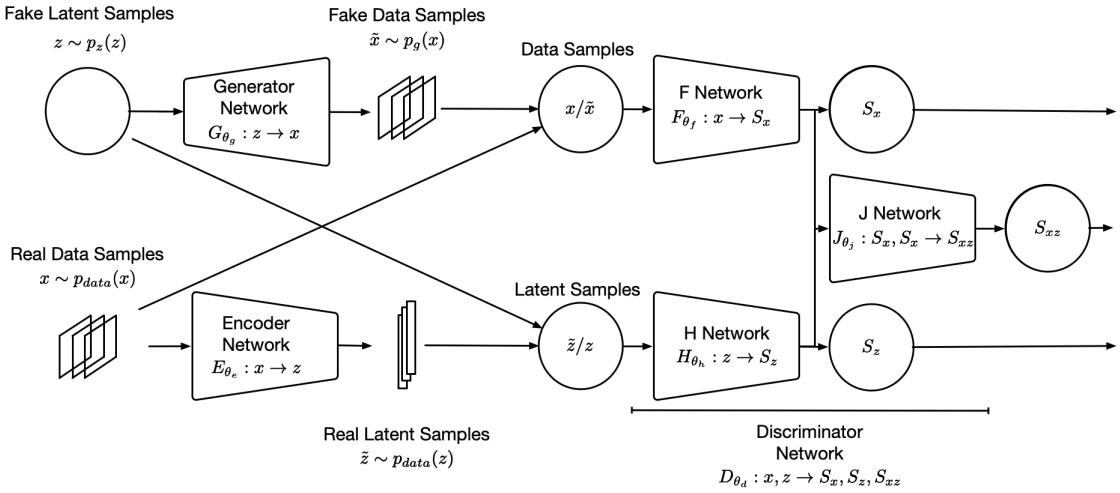


Figure 2.7: High level description of *BigBiGAN*'s architecture.

its approximation $\tilde{z} = E_{\theta_e}(x);\ x \sim p_{data}(x)$:

$$
\begin{aligned}
\ell_{G,E}(x, z, y) &= y(S_x(x) + S_z(z) + S_{xz}(x, z);\ y \in -1, +1 \\
\ell_D(x, z, y) &= h(yS_x(x)) + h(yS_z(z)) + h(yS_{xz}(x, z));\ y \in -1, +1 \\
&\min_{G_{\theta_g}, E_{\theta_e}} \mathbb{E}_{x \sim p_{data}(x)}[\ell_{G,E}(x, \tilde{z}, +1)] + \mathbb{E}_{z \sim p_z(z)}[\ell_{G,E}(\tilde{x}, z, -1)] \\
&\min_{D_{\theta_d}} \mathbb{E}_{x \sim p_{data}(x)}[\ell_D(x, \tilde{z}, +1)] + \mathbb{E}_{z \sim p_z(z)}[\ell_D(\tilde{x}, z, -1)] \\
&\text{where } \tilde{z} = E_{\theta_e}(x);\ \tilde{x} = G_{\theta_g}(z); h(t) = max(0, 1 - t)
\end{aligned}
\tag{2.29}
$$

StyleGAN[85,105,106] takes a different approach to other GANs with representation learning properties. Its main features are architectural changes rather than constrains in the latent distribution. First, it uses a mapping network M_{θ_m} that allows the generator to optimize an alternative latent space $w = M_{\theta_m}(z)$ (**Figure 2.8**). Second, it introduces the Adaptative Instance Normaliza-

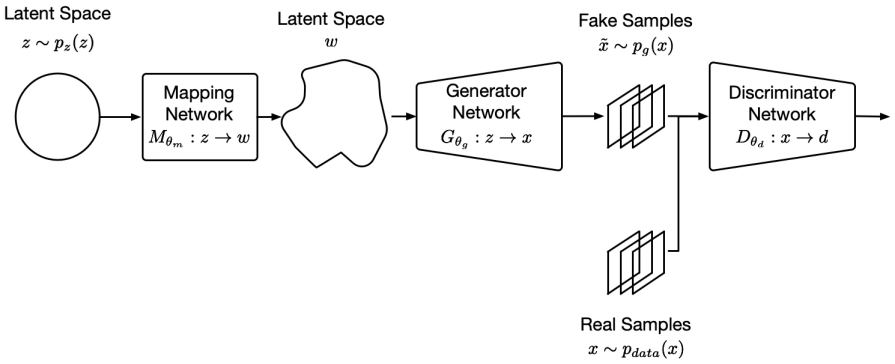


Figure 2.8: High level description of *StyleGAN*'s architecture.

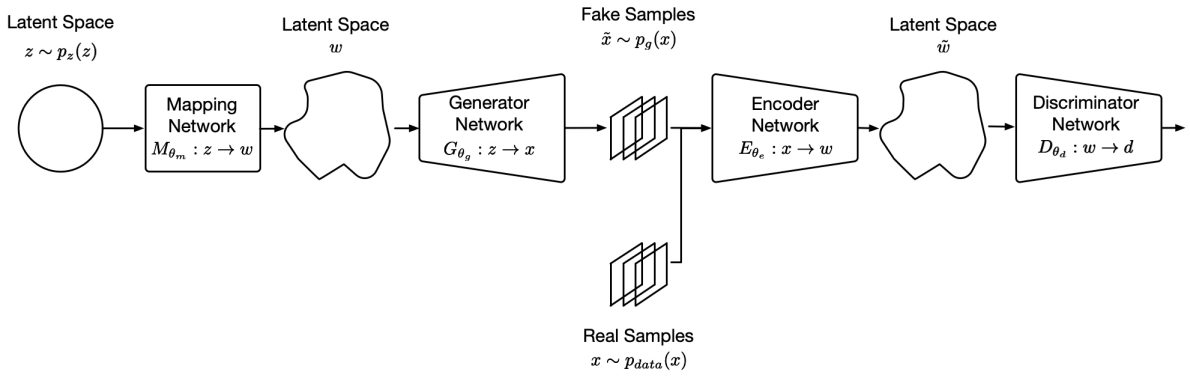


Figure 2.9: High level description of *ALAE*'s architecture.

tion (AdaIN) layer. The AdaIN layer batch normalizes and scales each individual sample $\tilde{x}$ based on its latent representation w, allowing to provide specific features to each generated instance.

Adversarial Latent Autoencoders (ALAEs)[107] combine ideas from StyleGAN and AAEs. ALAEs use the mapping network M_{θ_m}, AdaIN layer, and adversarial learning in the latent distribution. **2.30** shows ALAEs' loss function. The first term performs adversarial learning of the latent samples w and the second term calculates the mean squared error (MSE) between the latent sample w and its reconstruction $\tilde{w}$. **Figure 2.9** shows a high level representation of the model:

$$\min_{M_{\theta_m}, G_{\theta_g}} \max_{E_{\theta_e}, D_{\theta_d}} \mathbb{E}_{x \sim p_{data}(x)}[\log D_{\theta_d}(E_{\theta_e}(x))] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D_{\theta_d}(E_{\theta_e}(G_{\theta_g}(M_{\theta_m}(z)))))]$$

$$\min_{E_{\theta_e}, G_{\theta_g}} \mathbb{E}_{z \sim p_z(z)}[||w - \tilde{w}||_2^2] \tag{2.30}$$

$$\text{where } w = M_{\theta_m}(z),\ \tilde{x} = G_{\theta_g}(w),\ \tilde{w} = E_{\theta_e}(\tilde{x})$$

2.1.3 Performance Comparison of Deep Generative Models

Deep generative models are usually evaluated by their generative process, assessing their ability to reproduce the targeted distribution (**Section 2.1.2**). If the model has representation learning properties, its evaluation either targets disentanglement or its ability to generate samples with

certain features. They are not systematically evaluated on how applicable the representations are in alternative tasks. This is in contrast to self-supervised learning where the standard procedure is to measure performance on ImageNet linear classification. From the reviewed models, only BigBiGAN reports performance in such manner and it will be later compared to self-supervised approaches in **Section 2.2.5**.

Although VAEs and GANs can learn representations of data, the generative process places an additional burden that may not be a requisite for representation learning. As we will see in **Section 2.2**, other unsupervised methods that purely focus on feature extraction outpace GANs and VAEs in learning representations.

2.2 Self-Supervised Learning

During the last recent years, self-supervised learning has risen as an alternative to learning data representations without annotations. Unlike generative models, these approaches lack a generative process; which can be computationally expensive and may not be a requisite for representation learning. Instead, self-supervised learning focuses on creating useful representations of data based on its features; without relying on labels or manual annotations. Representations can then be used in downstream tasks such as classification or object detection.

Self-supervised learning has been successfully applied in a range of different fields, from natural language processing (NLP)[108,109], video[110–112], audio[113,114], or time-series[115–117] to images[118–123]. In this section we will focus on self-supervised learning for image representations.

The premise of self-supervised learning is that we can extract features from an image by training a model on a pretext task, features should then generalize to solve other classification or detection tasks. Convolutional neural networks (CNNs) or Vision Transformers (ViTs) are used as a feature extractors, also referred to as backbone network.

Initial works focused on defining pretext tasks that forced the backbone network to learn visual representations. These early approaches include predicting the relative position of patches in an image[124], predicting the rotation of an image[125], or rearranging the order of a random permutation of patches from an image[126]. However, representations learned from the pretext task do not necessarily align with optimal performance on downstream tasks. Noorozi *et al*[126] analyzed performance of learned representations at each convolutional layer. The results show that the last layer specializes in solving the pretext task and other layers may show better performance on a downstream task.

Motivated by previous research, current state-of-the-art approaches share a common setup. It is based on two ideas: (A) learned representations should capture image features and the relationship between different samples, and (B) learned representations should be robust against non-meaningful transformations such as color distortions[127], rotations, or zoom variations. **Figure 2.10** shows a high level representation of such setup. These methods start by applying random

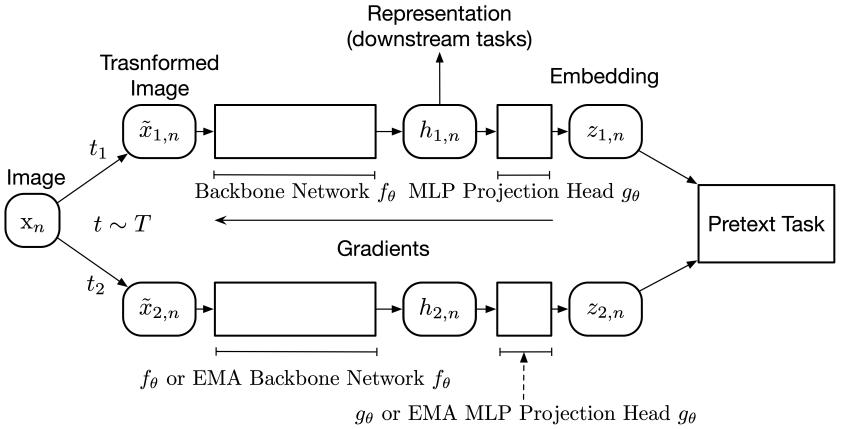


Figure 2.10: High level description of common setup for self-supervised learning models.

transformations $t \sim T$ to a batch of images x_n and running them through a backbone network f_θ and MLP projection head g_θ. The obtained embeddings z_n are then used to solve the specified pretext task. The introduction of an MLP projection head is motivated by the idea of isolating the h_n representations from the pretext task[126], allowing the z_n embeddings to specialize on it. Optimization of the networks varies based on the pretext task.

These models are evaluated primarily in two downstream tasks. Firstly, on linear classification by using representations from the frozen backbone network. Performance is usually evaluated by using 100%,10%, and 1% of the training set from ImageNet[92] or CIFAR10/100. Secondly, on object detection by finetuning the entire backbone network; common datasets include VOC07[128] or COCO[129].

A recurrent pitfall of self-supervised models is known as trivial solutions, collapse, or representation collapse, where representations remain with a constant value for different images. The approaches to solving this issue vary depending on the pretext task.

Self-supervised methods can be further differentiated based on their pretext tasks, including contrastive learning, distillation, clustering, and redundancy reduction.

2.2.1 Contrastive Learning

Contrastive learning defines image pairs in order to maximize similarity between representations of positive pairs compared to negative pairs, positive pairs are defined as different transformations of a reference image while negative pairs as transformations between the reference image and any other.

In general, contrastive learning methods achieve state-of-the-art results and are simple to implement. However, their main drawback is the requirement of good negative pairs to prevent collapse[130–132], translating in either memory banks[119,133] or very large batch sizes[118] which demand a large number of GPUs.

In this section we will cover the reference models: *SimCLR*, *MoCo*, and *Relational Reason-*

ing.

A Simple Framework for Contrastive Learning of Visual Representations (SimCLR)

SimCLR[118] uses a reference image x_n to define positive and negative pairs. Positive pairs are defined as two sampled transformations of the reference image $\{\{\tilde{x}_{1,n}, \tilde{x}_{2,n}\}; \tilde{x}_{1,n} = t_1(x_n),\ \tilde{x}_{2,n} = t_2(x_n),\ t \sim T\}$ while negative pairs are defined as sampled transformations of the reference image and any other in the batch $\{\{\tilde{x}_{1,n}, \tilde{x}_{1,\setminus n}\}$, and $\{\tilde{x}_{1,n}, \tilde{x}_{2,\setminus n}\}; \tilde{x}_{1,n} = t_1(x_n),\ \tilde{x}_{2,n} = t_2(x_n),\ t \sim T$ and $[x_n]_{n=1}^N\}$. This process results in a ratio of $2N-2\ :\ 1$ between negative and positive pairs. Transformed images are run through the backbone network f_θ to obtain representations $\{h_{i,n} = f_\theta(\tilde{x}_{i,n}); i = \{1,2\}\}$; the representations are then pushed through an MLP projection head to obtain the embeddings used in the contrastive task $\{z_{i,n} = g_\theta(z_{i,n}); i = \{1,2\}\}$.

The loss is defined as a contrastive prediction, where the similarity of the positive pairs is maximized against the negative logarithm of the softmax function (**2.31**). The contrastive loss function is also know as Information Noise Contrastive Estimation (infoNCE)[134]. Cosine similarity is used as similarity measure and τ denotes the temperature parameter. The temperature parameter τ controls the output sharpness of the softmax function:

$$
\begin{aligned}
\mathcal{L} &= \frac{1}{2N}\sum_{n=1}^{N}[\ell(1,2,n) + \ell(2,1,n)] \\
\ell_{i,j,n} &= -\log\frac{\exp\left(\text{sim}\left(z_{i,n}, z_{j,n}\right)/\tau\right)}{\sum_{m=1}^{2N}\exp\left(\text{sim}\left(z_{i,n}, z_{j,m}\right)/\tau\right) + \mathbb{I}_{[m\neq n]}\exp\left(\text{sim}\left(z_{i,n}, z_{i,m}\right)/\tau\right)} \\
\text{sim}(u,v) &= u^\top v/\|u\|\|v\|\ ;\ \mathbb{I}_{[m\neq n]} = \begin{cases} 0, & \text{if } m = n \\ 1, & \text{if } m \neq n \end{cases}
\end{aligned}
\tag{2.31}
$$

SimCLR provides a simple and effective method for representation learning. Nonetheless, it requires very large batch sizes to achieve good performance, putting a higher burden on computational requirements.

Momentum Contrast for Unsupervised Visual Representation Learning (MoCo)

MoCo[119,133] provides an alternative to the large batch sizes required by *SimCLR*. It introduces two new ideas, a memory bank that stores embeddings and exponential momentum average (EMA) of the backbone network f_θ and MLP projection head g_θ.

The momentum networks f_ξ and g_ξ are the exponential moving average (EMA) of the backbone network f_θ and MLP projection head g_θ, respectively. **2.32** defines how weights from f_ξ and g_ξ are updated, m controls how fast the weights change where higher values correspond to slower changes. The weights of the backbone network f_θ and MLP projection head g_θ are

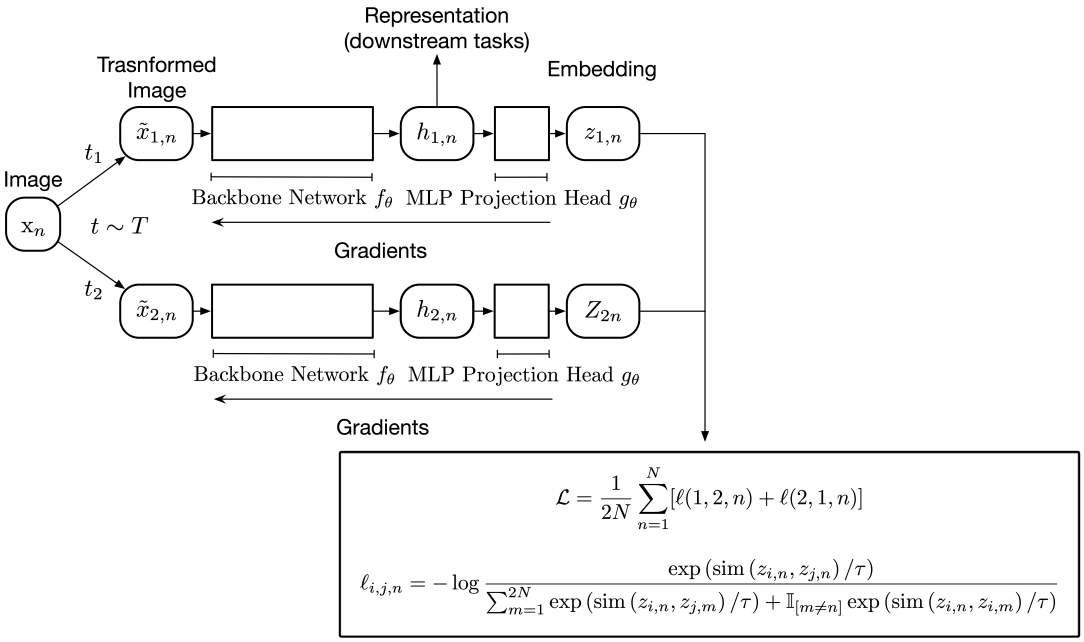


Figure 2.11: High level description of *SimCLR*'s architecture.

updated through regular gradient back-propagation. The motivation behind using a momentum networks f_ξ and g_ξ, insted of the original networks f_θ and g_θ, is to avoid abrupt changes in the embeddings that could cause the model to collapse:

$$\xi \leftarrow m\xi + (1-m)\theta \tag{2.32}$$

MoCo also uses a memory bank to store the z embeddings. The z embeddings are obtained by running images through the momentum backbone network f_ξ and momentum MLP projection head g_ξ. The size of the memory bank K can be considerably larger than the batch size (e.g. 32×), allowing to achieve higher ratios $K:1$ between negative and positive pairs without large batch sizes.

MoCo applies two sampled transformations for each image x_n in the batch $\{\tilde{x}_{1,n} = t_1(x_n),\ \tilde{x}_{2,n} = t_2(x_n),\ t \sim T\}$. One of the transformations $\tilde{x}_{2,n}$ is pushed through the EMA networks f_ξ and g_ξ, storing the obtained embedding into the memory bank $z_{2,n} = g_\xi(f_\xi(\tilde{x}_{2,n}))$. The other transformation $\tilde{x}_{1,n}$ is pushed through the backbone network f_θ and MLP projection head g_θ, obtaining the embedding $z_{1,n} = g_\xi(f_\xi(\tilde{x}_{1,n}))$. The $z_{1,n}$ embeddings are directly used in the contrastive loss with all the memory bank's embeddings z_k. *MoCo* uses the infoNCE function in order to maximize similarity between positive pairs against negative pairs (**2.33**). Dot product is used as similarity

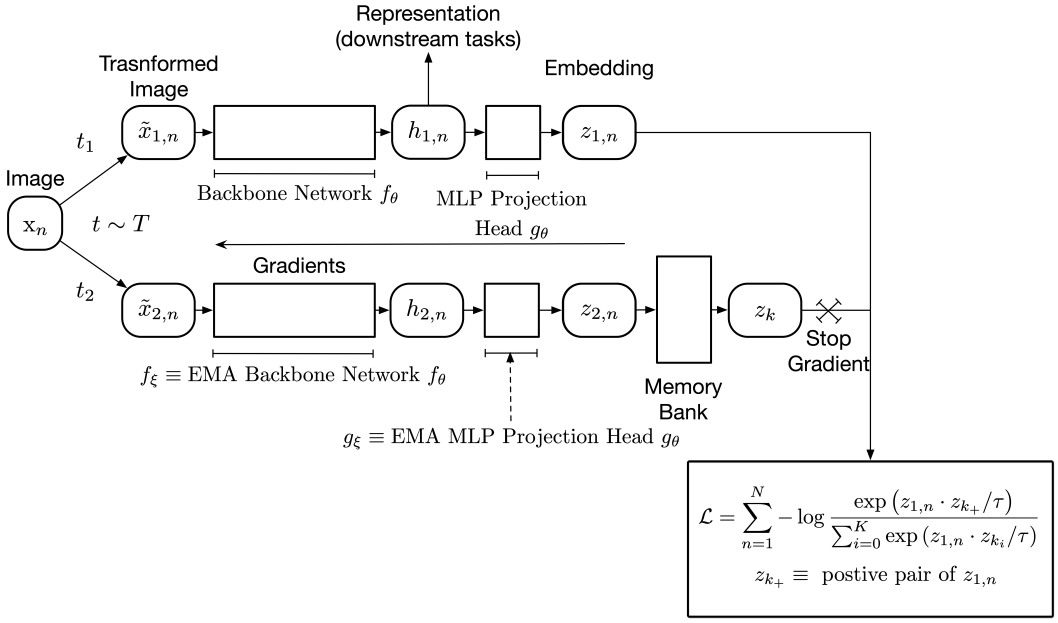


Figure 2.12: High level description of *MoCo*'s architecture.

measure, τ denotes the temperature parameter, and K defines the size of the memory bank:

$$\mathcal{L} = \sum_{n=1}^{N} -\log \frac{\exp\left(z_{1,n} \cdot z_{k_+}/\tau\right)}{\sum_{i=0}^{K} \exp\left(z_{1,n} \cdot z_{k_i}/\tau\right)} \qquad z_{k_+} \equiv \text{ postive pair of } z_{1,n} \tag{2.33}$$

By using a memory bank of representations, *MoCo* allows to separate the batch size from obtaining a large number of negative pairs. Regardless, this approach still requires a large amount of GPU RAM and two different forward passes of the images.

Self-Supervised Relational Reasoning for Representation Learning (Relational Reasoning)

Relational Reasoning[135] also defines a contrastive learning setting as a pretext task, using positive and negative pairs. It uses a relational network r_ϕ as a learnable probability similarity metric that characterizes the relationship between pairs.

Relational Reasoning applies K random transformations $\{[t_i]_{k=1}^{K};\ t \sim T\}$ to each image in the batch. Each transformed image $\tilde{x}_{k,n}$ is then run through the backbone network f_θ to obtain representations $z_{k,n} = f_\theta(\tilde{x}_{k,n})$. Afterward, an aggregation function $a(\cdot,\cdot)$ concatenates features of representations to define positive pairs (intra-reasoning) $a(h_{i,n}, h_{j,n})$ and negative pairs (inter-reasoning) $a(h_{i,n}, h_{j,\setminus n})$. Pairs are then run through the relational network r_ϕ to obtain the probability similarity $y = r_\phi(a(h_{i,n}, h_{j,n}))$. The loss is defined as a binary classification problem, where the relation score y defines the probability estimation of the representation membership over the

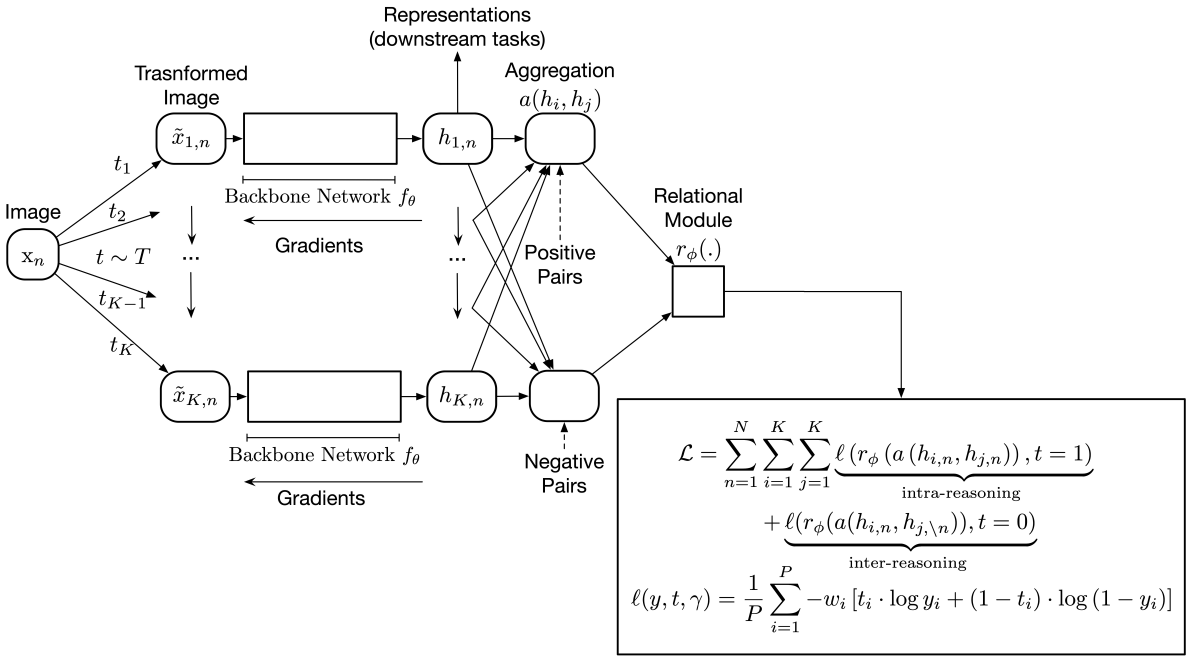


Figure 2.13: High level description of *Self-Supervised Relational Reasoning for Representation Learning* architecture.

presentation pairs P (**2.34**). Positive pairs (intra-reasoning) are labeled as 1 while negative pairs (intra-reasoning) are labeled as 0. w_i is a scaling factor where $\gamma \geq 0$.

$$\begin{aligned}
\mathcal{L} &= \sum_{n=1}^{N}\sum_{i=1}^{K}\sum_{j=1}^{K} \underbrace{\ell\left(r_\phi\left(a\left(h_{i,n}, h_{j,n}\right)\right), t=1\right)}_{\text{intra-reasoning}} + \underbrace{\ell(r_\phi(a(h_{i,n}, h_{j,\setminus n})), t=0)}_{\text{inter-reasoning}} \\
\ell(y,t,\gamma) &= \frac{1}{P}\sum_{i=1}^{P} -w_i\left[t_i \cdot \log y_i + \left(1-t_i\right)\cdot\log\left(1-y_i\right)\right] \\
w_i &= \frac{1}{2}\left[\left(1-t_i\right)\cdot y_i + t_i \cdot \left(1-y_i\right)\right]^\gamma
\end{aligned} \tag{2.34}$$

While *SimCLR* and *Relational Reasoning* are both contrastive methods and rely on a large number of negative pairs, the ability of *Relational Reasoning* on generating a larger number of image transformations K creates a different percentage in negative and postive pairs. *SimCLR* typically has a ratio of 98% negative to 2% positive pairs, *Relational Reasoning* has 50% ratio for both. The authors theorize that the larger set of positive pairs allow the realtional reasoning approach to more effectively create representations.

2.2.2 Distillation

Distillation methods also maximize similarity between postive samples, but contrary to contrastive learning, distillation approaches do not use negative pairs. Instead, they use teacher-student solutions in order to produce representations and avoid model collapse. Teacher-student

approches translate into different versions of architecture imbalances.

These methods provide on-par performance with contrastive learning. Nonetheless, they require two independent forward passes and additional memory costs for the teacher network weights.

Bootstrap Your Own Latent (BYOL)

BYOL[121] uses a similar setup as *MoCo*. However, *BYOL* does not use a memory bank as it does not use negative pairs. The student network is composed of a backbone network f_θ and MLP projection head g_θ plus a predictor network p_θ, while the teacher network is composed of the exponential momentum average (EMA) of the backbone network f_ξ and MLP projection head g_ξ.

The momentum networks f_ξ and g_ξ are the exponential moving average (EMA) of the backbone network f_θ and MLP projection head g_θ, respectively. **2.35** defines how weights from f_ξ and g_ξ are updated. m controls how fast the weights change where higher values correspond to slower changes. The weights of the backbone network f_θ and MLP projection head g_θ are updated through regular gradient back-propagation:

$$\xi \leftarrow m\xi + (1-m)\theta \tag{2.35}$$

BYOL applies two sampled transformations to each image in the batch $\{\tilde{x}_{1,n} = t_1(x_n),\ \tilde{x}_{2,n} = t_2(x_n),\ t \sim T\}$. One of the transformations $\tilde{x}_{2,n}$ is pushed through the teacher networks f_ξ and g_ξ, obtaining the corresponding embedding $z_{2,n} = g_\xi(f_\xi(\tilde{x}_{2,n}))$. The other transformation $\tilde{x}_{1,n}$ is pushed through the student networks f_θ and g_θ, obtaining the embedding $z_{1,n} = g_\theta(f_\theta(\tilde{x}_{1,n}))$. Afterward, the student's embedding $z_{1,n}$ is used to predict the teacher's embedding $z_{2,n}$ through the prediction network p_θ. *BYOL* uses the mean squared error (MSE) between the normalized versions of the teacher's embedding $z_{2,n}$ and the prediction from the student's embedding $q_{1,n} =$

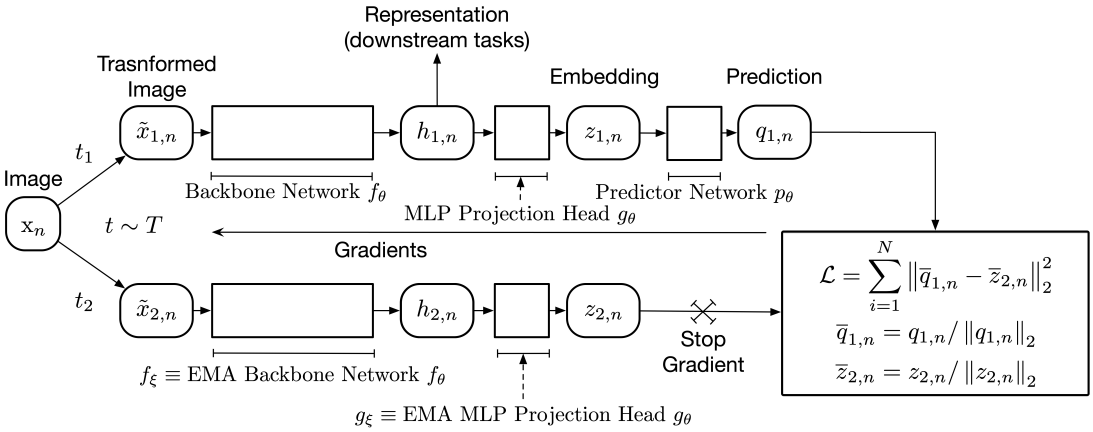


Figure 2.14: High level description of *BYOL*'s architecture.

$p_\theta(z_{1,n})$ (**2.36**):

$$\begin{aligned} \mathcal{L} &= \sum_{i=1}^{N} \left\| \overline{q}_{1,n} - \overline{z}_{2,n} \right\|_2^2 \\ \overline{q}_{1,n} &= q_{1,n} / \left\| q_{1,n} \right\|_2 \\ \overline{z}_{2,n} &= z_{2,n} / \left\| z_{2,n} \right\|_2 \end{aligned} \tag{2.36}$$

As *BYOL* does not use negative pairs, it is less impacted by batch size compared to contrastive methods. Interestingly, *BYOL*'s performance drops to random when batch normalization is removed[136], i.e. *BYOL* collapses. This finding suggests that batch normalization implies some form of implicit contrastive learning, where each instance of the batch is normalized based on the mean and standard deviation of all samples of the batch.

Simple Siamese Representation Learning (SimSiam)

SimSiam[120] uses a similar approach to *BYOL*. In fact, it reduces the sources of asymmetry between teacher and student networks. *SimSiam* uses the backbone network f_θ and MLP projection head g_θ for the student networks and the backbone network f_ξ and MLP projection head g_ξ for the teacher networks, the different between them is that only the student networks are updated through stochastic gradient descent. After each epoch, the weights of the student networks are copied into the teacher.

As *BYOL*, *SimSiam* applies two sampled transformations to each image $\{\tilde{x}_{1,n} = t_1(x_n),\ \tilde{x}_{2,n} = t_2(x_n),\ t \sim T\}$. One of the transformations $\tilde{x}_{2,n}$ is pushed through the the teacher networks f_ξ and g_ξ, obtaining the corresponding embedding $z_{2,n} = g_\xi(f_\xi(\tilde{x}_{2,n}))$. The other transformation $\tilde{x}_{1,n}$ is pushed through the student networks f_θ and g_θ, obtaining the embedding $z_{1,n} = g_\theta(f_\theta(\tilde{x}_{1,n}))$. Afterward, the student's embedding $z_{1,n}$ is used to predict the teacher's embedding $z_{2,n}$ through the prediction network p_θ. *SimSiam* uses the negative cosine similarity between the teacher's

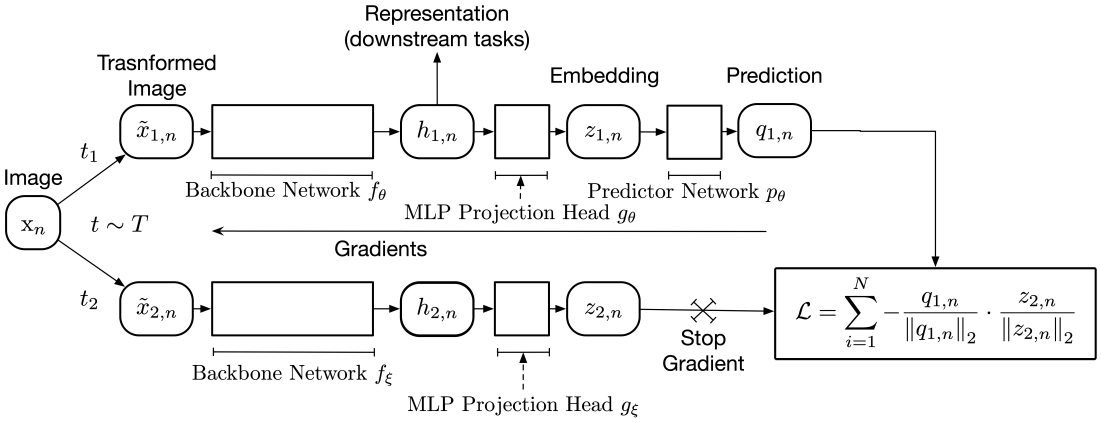


Figure 2.15: High level description of *SimSiam*'s architecture.

embedding $z_{2,n}$ and the prediction from the student's embedding $q_{1,n} = p_\theta(z_{1,n})$ (**2.37**):

$$\mathcal{L} = \sum_{i=1}^{N} -\frac{q_{1,n}}{\|q_{1,n}\|_2} \cdot \frac{z_{2,n}}{\|z_{2,n}\|_2} \tag{2.37}$$

As in the case of *BYOL*, *SimSiam* is less sensitive to batch size compare to contrastive learning methods.

2.2.3 Clustering

Self-supervised clustering frameworks create groups of representations in a direct manner. The main idea is that different transformations of the same image are more likely to be assigned to the same cluster, similar to positive pairs in contrastive learning.

Collapse is represented by all sample images being assigned to one cluster. It is avoided with methods that control the clustering process such as Sinkhorn-Knopp algorithm[122, 137]. Clustering frameworks also require to specify the expected number of clusters.

These methods do not require negative samples and they converge faster than contrastive learning. The clustering objective imposes harder constraints. They also have less computational requirements, yet the clustering process can be costly.

DeepCluster[138] and *SwAV*[122] are the most representative clustering frameworks.

DeepCluster

DeepCluster[138] can be seen as an initial step on self-supervised clustering approaches, although its setup is different from most self-supervised methods. It does not use any kind of image transformations. Instead, it iterates through two steps: (1) clustering of image representations through K-Means and (2) training of a convolutional neural network (CNN) by using cluster membership of image representations as labels.

DeepCluster is based on the assumption that even the representations extracted from a randomly initialized CNN provide guidance on possible cluster separations. During the clustering process, all images x_n are pushed through the backbone network f_θ to obtain representations $h_n = f_\theta(x_n)$. K-Means[139] is used to find the set of clusters K that minimizes **2.38** where C is the centroid's matrix, y_n is the vector of cluster membership for each representation h_n, and D is the dimensionality of representations:

$$\min_{C} \frac{1}{N} \sum_{n=1}^{N} \min_{y_n \in \{0,1\}^k} \|h_n - C y_n\|_2^2 \text{ where } C \in \mathbb{R}^{D \times K}; h_n \in \mathbb{R}^{D \times 1} \tag{2.38}$$

After the clustering process, the backbone network f_θ and MLP network g_θ are trained to

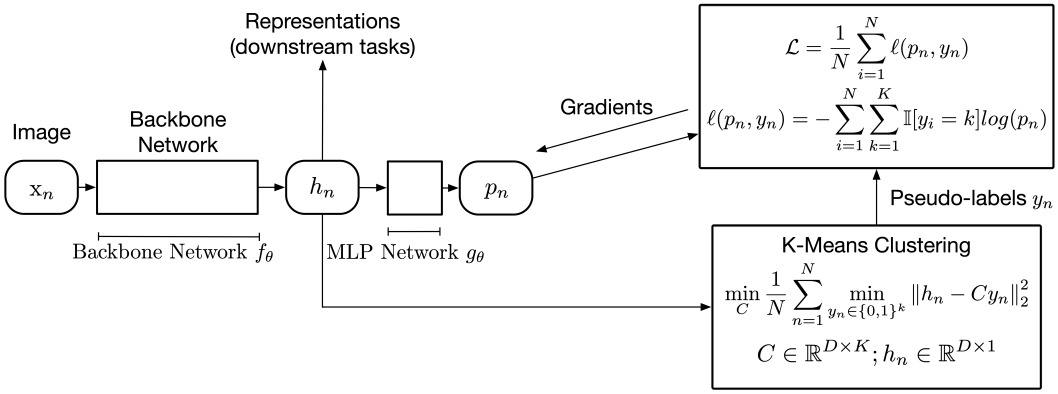


Figure 2.16: High level description of *DeepCluster*'s architecture.

predict probability cluster membership p_n for each of the image representations h_n. In this step, *DeepCluster* minimizes the cross entropy loss between the predictions $p_n = g_\theta(f_\theta(x_n))$ and pseudo-labels y_n (**2.39**):

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}\mathbb{I}[y_i = k]log(p_n) \tag{2.39}$$

DeepCluster uses two methods to avoid collapse. First, when a cluster gets no assignations, its centroid gets replaced by another non-empty and randomly selected cluster's centroid plus a small random perturbation. Representations previously assigned to the non-empty cluster are then reassigned between these two clusters. Second, to avoid the backbone network f_θ on specializing in differentiating a small subset of clusters, images are uniformly sampled across the different clusters.

Finally, the authors showed that some over segmentation is beneficial. *DeepCluster* achieves its best performance on ImageNet[92] when using $10,000$ clusters, while ImageNet contains $1,000$ different classes.

Swapping Assignments between multiple Views of the same image (SwAV)

SwAV[122] is based on the premise that different transformations of the same image are more likely to belong to the same cluster.

SwAV performs a soft clustering solution. It uses a set of trainable cluster prototypes $C = \{c_1, ..., c_K\}$ and codes $Q = \{q_1, ..., q_N\}$. The cluster prototypes c_k are vectors with equal dimensionality to embeddings z_n that represent each of the clusters. Codes q_n are vectors of K dimensionality where each dimension defines the probability of the embedding z_n belonging to a cluster k.

Each training iteration can be divided into two steps. First, the computation of the soft-assignment codes Q (**2.40**). Second, the computation of the loss and optimization of cluster

prototypes C and networks f_θ and g_θ through stochastic gradient descent (**2.41**).

SwAV starts by applying two sampled transformations to each image in the batch $\{\tilde{x}_{1,n} = t_1(x_n),\ \tilde{x}_{2,n} = t_2(x_n),\ t \sim T\}$. Transformed images are run through the networks f_θ and g_θ to obtain embeddings $\{z_{i,n} = g_\theta(f_\theta(\tilde{x}_{i,n});\ i = \{1,2\}\}$.

At this stage, *SwAV* computes the soft-assignment codes $Q = \{q_1, ..., q_N\} \in \mathbb{R}_+^{K\times N}$ for each batch embedding $Z = \{z_1, ..., z_N\}$. It maximizes the codes Q that account for the similarity between embeddings Z and clusters prototypes C. The constraints in **2.40** enforce that on average each cluster is selected at least N/K times in the batch, where $\mathbb{1}_K$ denotes a vector of ones in dimension K and $\mathbb{1}_N$ denotes a vector of ones in dimension N. Codes Q are iteratively found through the Sinkhorn-Knopp algorithm[137]:

$$\begin{aligned} &\max_{Q\in\mathbb{Q}} Tr(Q^T C^T Z) + \epsilon H(Q); \\ &H(Q) = -\sum_{i,j} Q_{i,j} log(Q_{i,j}) \\ &\mathbb{Q} = \{Q \in \mathbb{R}_+^{K\times N} | Q\mathbb{1}_N = \frac{1}{K}\mathbb{1}_K, Q^T\mathbb{1}_K = \frac{1}{N}\mathbb{1}_N\} \end{aligned} \tag{2.40}$$

Afterward, *SwAV* evaluates the loss by swapping codes and embeddings for each pair with two transformations of the same image $\{q_{i,n}, z_{j,n}\}; i \neq j$. Swapping codes and embeddings enforces for the two transformations to have similar cluster memberships. **2.41** shows *SwAV*'s cross-entropy loss between the computed code $q_{i,n}$ and the probability vector of cluster membership $p_{j,n}^{(k)}$. The probability of cluster membership $p_{i,n}^{(k)}$ is computed over the softmax function of

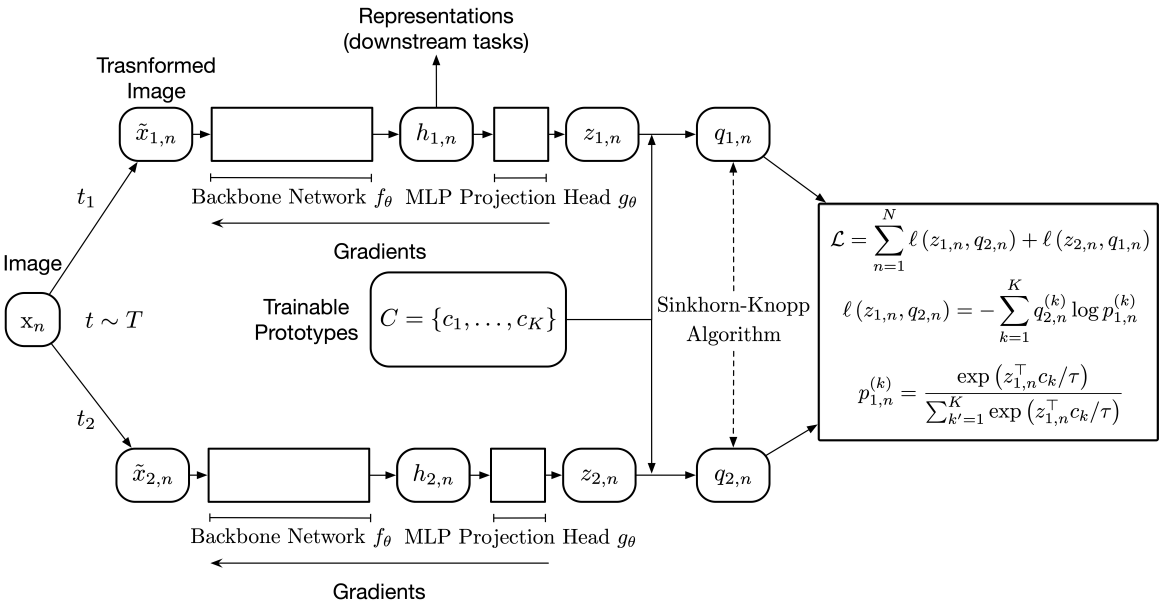


Figure 2.17: High level description of *SwAV*'s architecture.

similarities between cluster prototypes c_k and the embedding $z_{i,n}$:

$$\begin{aligned} \mathcal{L} &= \sum_{n=1}^{N} \ell\left(z_{1,n}, q_{2,n}\right) + \ell\left(z_{2,n}, q_{1,n}\right) \\ \ell\left(z_{1,n}, q_{2,n}\right) &= -\sum_{k=1}^{K} q_{2,n}^{(k)} \log p_{1,n}^{(k)} \text{ where } p_{1,n}^{(k)} = \frac{\exp\left(z_{1,n}^{\top} c_k / \tau\right)}{\sum_{k'=1}^{K} \exp\left(z_{1,n}^{\top} c_k / \tau\right)} \end{aligned} \tag{2.41}$$

SwAV uses two ideas to avoid collapse and to allow a flexible clustering process. First, it enforces equal partition of samples N across K clusters where no cluster has more than N/K assignations. Second, it uses soft assignations between embeddings and clusters. The soft-assignment solution creates a flexible space of codes Q compared to hard-assignments[137], allowing to create more accurate clusters even with high class imbalances. *SwAV* uses an online clustering version of the Sinkhorn-Knopp algorithm to optimize codes Q; enabling the use of any dataset size.

SwAV can be trained with large and small batches. Nevertheless, in order to compute codes Q and effectively avoid collapse, it requires storing embeddings z_n when the number of clusters is much larger than the batch size.

2.2.4 Redundancy Reduction

Barlow Twins[123] introduced a new approach to self-supervised learning. This model aims to create representations that are invariant to distortions while minimizing the redundant information between feature dimensions in the embeddings.

It uses two different transformations to distort each image in the batch $\{\tilde{x}_{1,n} = t_1(x_n),\ \tilde{x}_{2,n} = t_2(x_n); t \sim T, \text{ and } [x_n]_{n=1}^{N}\}$. Transformed images are run through the networks f_θ and g_θ to obtain embeddings $z_{i,n} = g_\theta(f_\theta(\tilde{x}_{i,n}))$. Right after, embeddings are batch normalized for each transformation set $\{Z_i \in \mathcal{R}^{NxD}; i = \{1,2\}\}$ and it calculates $C \in [-1,1]^{DxD}$ as the cross-correlation matrix computed between transformation sets $\{Z_1, Z_2\}$ along the batch dimension (**2.42**). D denotes the dimensionality of the embeddings $z_{i,n}$:

$$C_{ij} = \frac{\sum_{n=1}^{N} z_{1,n,i} z_{2,n,j}}{\sqrt{\sum_{n=1}^{N} \left(z_{1,n,i}\right)^2} \sqrt{\sum_{n=1}^{N} \left(z_{2,n,i}\right)^2}} \tag{2.42}$$

The loss is defined by two terms. The invariance term that attempts to create representations invariant to distortions and the redundancy reduction term where off-diagonal features of the correlation matrix are minimized to prevent any redundant information between them (**2.43**). The parameter λ is used as a weighting factor to balance the difference in number between the

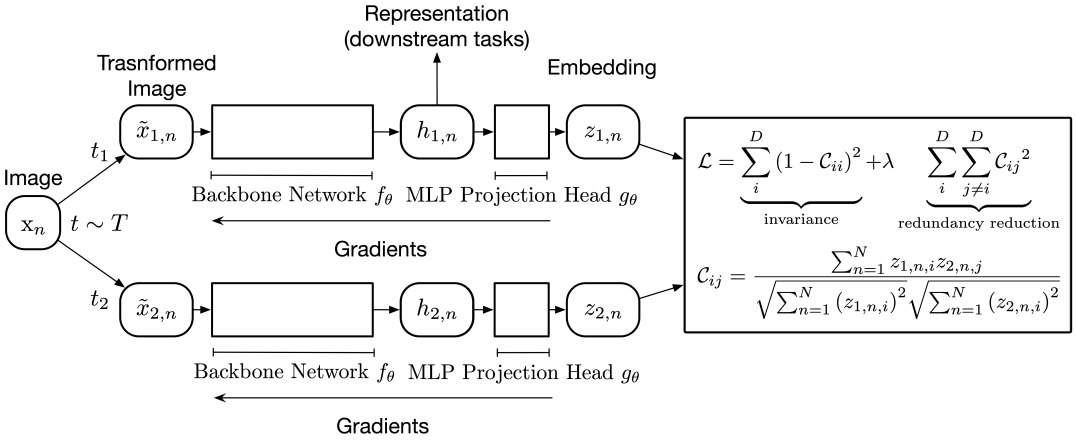


Figure 2.18: High level description of *Barlow Twins*' architecture.

diagonal and off-diagonal terms:

$$\mathcal{L} = \underbrace{\sum_{i}^{D} \left(1 - \mathcal{C}_{ii}\right)^2}_{\text{invariance}} + \lambda \underbrace{\sum_{i}^{D} \sum_{j \neq i}^{D} \mathcal{C}_{ij}^2}_{\text{redundancy reduction}} \tag{2.43}$$

Barlow Twins avoids collapse by forcing decorrelation between features of the representations and by batch normalizing the representations before computing the cross-correlation matrix. It benefits from higher dimensional representations, in contrast to *SimCLR* or *BYOL* that quickly saturate performance. In addition, *Barlow Twins* is robust against batch sizes, providing on-par results with other self-supervised approaches and releasing the user from high computational requirements[118, 121, 133].

2.2.5 Performance Comparison of Self-Supervised Methods

Self-supervised models are primarily evaluated on two downstream tasks: (1) by using representations on a linear classification, including datasets such as ImageNet[92] or CIFAR10/100, and (2) by fine tuning the backbone network on object detection, common datasets include VOC07[128] or COCO[129]. In order to compare these methods and their performance, we will focus on linear classification over ImageNet.

Evaluation on linear classification can be done either by providing 100% of the training samples or a subset of them, usually 1% or 10%. The former is commonly known as fully supervised learning while the latter is referred to as semi-supervised learning.

By focusing on **Table 2.1**, we can conclude that self-supervised learning has closed the gap with supervised models within 7% accuracy. However, their main advantage can be seen in semi-supervised learning. **Table 2.2** shows how self-supervised learning greatly outperforms supervised models, providing representations that generalize even when data annotations are

Method	Top-1	Top-5
Supervised	76.5	-
BigBiGAN	61.3	81.9
SimCLR	69.3	89.0
MoCo	71.1	90.1
SimSiam	71.3	-
BYOL	74.3	91.6
SwAV	75.3	-
Barlow Twins	73.2	91.0

Table 2.1: Top-1 and top-5 accuracies (in %) of relevant self-supervised models on linear classification over ImageNet. This data was taken from each of the model publications[118–123]. Top-1 score checks if the class with the top probability score aligns with the actual label. Top-5 checks if the top 5 probability scores contain the actual label. All models used a ResNet-50[140] as backbone network f_θ. Non provided accuracies are shown as '-'.

Method	Top-1		Top-5	
	1 %	10 %	1 %	10 %
Supervised	25.4	56.4	48.4	80.4
SimCLR	48.3	65.6	75.5	87.8
BYOL	53.2	68.8	78.4	89.0
SwAV	53.9	70. 2	78.5	89. 9
Barlow Twins	55. 0	69.7	79. 2	89.3

Table 2.2: Top-1 and top-5 accuracies (in %) of relevant self-supervised models on linear classification over ImageNet using 1% and 10% of training examples. This data was taken from each of the model publications[118,121–123]. Top-1 score checks if the class with the top probability score aligns with the actual label. Top-5 checks if the top 5 probability scores contain the actual label. All models used a ResNet-50[140] as backbone network f_θ.

scarce. Within self-supervised models, we can establish that *BYOL*, *SwAV*, and *Barlow Twins* show the highest performance, all of them within a difference of 2.1% accuracy. The difference between them is their applicability based on computational requirements, understood dynamics, and convergence guarantees.

In practice, *BYOL* achieves on-par results with other described methods and does not require large batches. However, as a distillation method, it requires asymmetric network solutions. Their learning dynamics have not been fully understood yet, and although there have been some initial attempts on understanding the relevance of batch normalization or stop-gradient; there is no description of how asymmetric architectures link to the optimization of the loss[136,141].

Clustering methods such as *SwAV* are more prone to collapse and require specific constraints to avoid it, e.g. Sinkhorn-Knopp algorithm. Although they do not require large batches, they need to store representations if the number of clusters is much larger than the batch size. On the other hand, these methods converge faster than other approaches.

Finally, *Barlow Twins* is robust against batch size as *SwAV* and *BYOL* and does not require asymmetries or any kind of memory storage. Its main advantage is that it can provide competing

results by constructing a simple and intuitive learning objective.

Chapter 3

PathologyGAN: Learning deep representations of cancer tissue

3.1 Introduction

In this chapter, we propose to use Generative Adversarial Networks (GANs) for unsupervised representation learning of cancer tissue. We argue that GAN-based models could be promising to achieve such goal for the following reasons:

- By being able to generate high-fidelity images, GANs could learn the most relevant descriptions of tissue morphologies.
- The latent representations learned by GANs could help us quantify differences in tissue architectures free from supervised information.

We introduce PathologyGAN, a generative adversarial network that can generate high-fidelity tissue images from a structured and interpretable latent space. We assess the quality of the generated images through the Fréchet Inception Distance (FID) score using convolutional features of a pre-trained Inception-V1 network. In addition, we evaluate the quality of the images by asking pathologists to classify tissue images between real and fake.

We explore the structure and interpretability of PathologyGAN's latent representations. We show how distinct regions of the latent space relate to tissue and cellular information in the generated images. Furthermore, we perform linear interpolations and simple arithmetic vector operations in the latent space and show how they translate into tissue features in generated images.

Finally, we study the impact of the GAN's architectural features on the representation learning properties of the model and the impact of the dataset size on the model's ability to converge and generalize the original distribution of images.

3.2 Methods

3.2.1 PathologyGAN

We developed PathologyGAN using BigGAN[84] as a baseline architecture, and introduced changes that empirically improved the Fréchet Inception Distance (FID) and the structure of the latent space.

BigGAN has been shown to be a successful GAN in replicating datasets with a diverse number of classes and large amounts of samples, such as ImageNet with approximately $1M$ samples and $20K$ classes. For this reason, we theorize that such a model will be able to learn and replicate the diverse tissue morphologies contained in whole slide images (WSIs), being able to handle the large number of tiles resulting from dividing the WSIs.

We followed the same architecture as BigGAN, employed Spectral Normalization in both generator and discriminator, self-attention layers, and we also use orthogonal initialization and regularization as mentioned in the original paper.

We make use of the Relativistic Average Discriminator[96], where the discriminator's goal is to estimate the probability of the real data being more realistic than the fake data. We take this approach instead of following the Hinge loss[142] as the GAN objective. We find that this change makes the model convergence faster and produces higher quality images (**Section 3.3.1**). Images using the Hinge loss did not capture the morphological structure of the tissue (**Appendix A.1**). The discriminator and generator loss functions are formulated as in **3.1** where $p_{data}(x)$ is the distribution of real data, $p_g(x)$ is the distribution for the fake data, $C_{\theta_d}(x)$ is the non-transformed discriminator output or critic, and σ denotes the sigmoid function:

$$
\begin{aligned}
L_{Dis} &= -\mathbb{E}_{x\sim p_{data}(x)}\left[\log\left(D_{\theta_d}(x)\right)\right] - \mathbb{E}_{\tilde{x}\sim p_g(x)}\left[\log\left(1 - D_{\theta_d}(\tilde{x})\right)\right], \\
L_{Gen} &= -\mathbb{E}_{\tilde{x}\sim p_g(x)}\left[\log\left(D_{\theta_d}(\tilde{x})\right)\right] - \mathbb{E}_{x\sim p_{data}(x)}\left[\log\left(1 - D_{\theta_d}(x)\right)\right], \\
D_{\theta_d}(x) &= \sigma\left(C_{\theta_d}(x) - \mathbb{E}_{\tilde{x}\sim p_g(x)}[C_{\theta_d}(\tilde{x})]\right), \\
D_{\theta_d}(\tilde{x}) &= \sigma\left(C_{\theta_d}(\tilde{x}) - \mathbb{E}_{x\sim p_{real}(x)}[C_{\theta_d}(x)]\right), \\
&\text{where } \tilde{x} = G_{\theta_g}(w),\ w = M_{\theta_m}(z),\ \text{and } z \sim \mathcal{N}(0, \mathbb{I})
\end{aligned}
\tag{3.1}
$$

Additionally, we introduce elements from StyleGAN[85] with the purpose of allowing the generator to freely optimize the latent space and find high-level features of cancer tissue. First, a mapping network M_{θ_m} composed of four dense ResNet layers[143], placed after the latent vector $z \sim \mathcal{N}(0, \mathbb{I})$, with the purpose of allowing the generator to find the latent space $w = M_{\theta_m}(z)$ that better disentangles the latent factors of variation. Second, style mixing regularization, where two different latent vectors z_1 and z_2 are run into the mapping network and fed at the same time to the generator, randomly choosing a layer in the generator and providing w_1 and w_2 to the different halves of the generator (e.g. on a generator of ten layers and being six the randomly selected

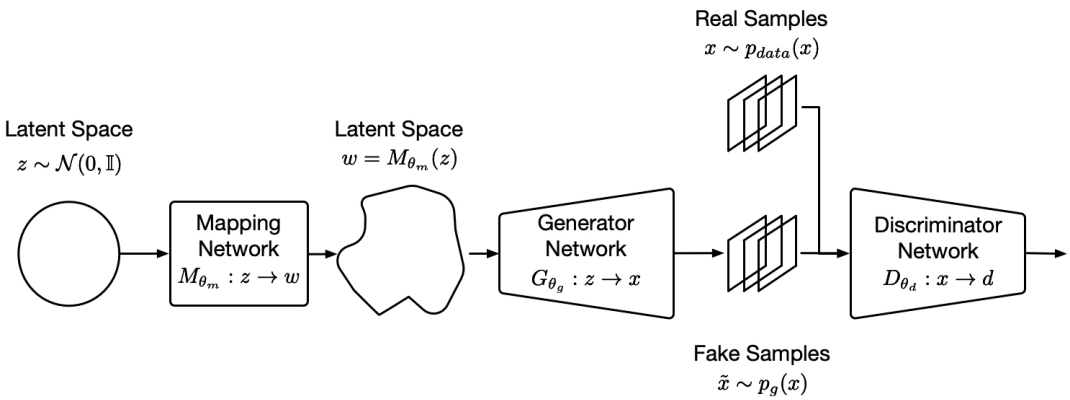


Figure 3.1: High level description of PathologyGAN's architecture.

layer, w_1 would feed layers one to six and w_2 seven to ten). Style mixing regularization encourages the generator to localize the high-level features of the images in the latent space. Third, we also use adaptive instance normalization (AdaIN)[85] on our models, providing the entire latent vectors w.

We use the Adam optimizer[144] with $\beta_1 = 0.5$ and the same learning rates of 0.0001 for both generator and discriminator. The discriminator takes 5 steps for each of the generator. Each model was trained on an NVIDIA Titan RTX 24 GB for approximately 72 hours. We trained our model on the breast and colorectal cancer datasets (**Section 1.5**) for 45 and 130 epochs, respectively.

3.2.2 Evaluation metric on PathologyGAN

The Fréchet Inception Distance (FID)[83] is a common metric used to measure GANs performance, it quantifies the GAN's ability to learn and reproduce the original data distribution. The goal of the FID score is to measure the similarity in quality and diversity between real $p_{data}(x)$ and generated data $p_g(x)$.

Instead of measuring the distance between the real and generated distributions in the pixel space, it uses a pretrained ImageNet Inception Network[93] to extract features of each image, reducing the dimensionality of the samples and obtaining vision-relevant features. Feature samples are fitted into a multivariate Gaussian distribution obtaining real $\mathcal{N}(\mu_{data}, \Sigma_{data})$ and generated $\mathcal{N}(\mu_g, \Sigma_g)$ feature distributions. Finally, it uses the Fréchet distance[94] to measure the difference between the two Gaussian distributions (**3.2**):

$$FID = d^2((\mu_{data}, \Sigma_{data}), (\mu_g, \Sigma_g)) = ||\mu_{data} - \mu_g||_2^2 + Tr(\Sigma_{data} + \Sigma_g - 2(\Sigma_{data}\Sigma_g)^{1/2}) \quad (3.2)$$

We evaluate our model by calculating the FID score from 10K generated images and another randomly sampled $10K$ real images of the test set.

We focus on using FID as it is a common GAN evaluation method[84–87] that reliably captures

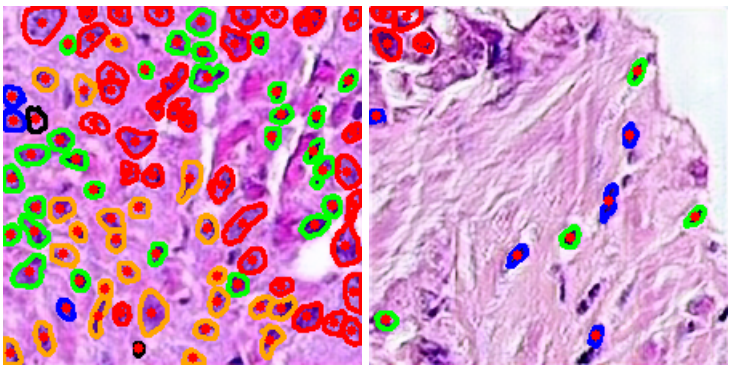


Figure 3.2: Hover-Net identifies different cell types in our generated images. Neoplastic epithelial cells are highlighted in red, non-neoplastic epithelial in orange, inflammatory in green, and connective cells in blue.

differences between the real and generated distributions.

3.2.3 Cell type quantification on generated images

In our results, we use counts of different cell types as a means to measure the representation learning properties of the model. The motivation behind this approach is to ensure that our model captures meaningful and faithful representations of the tissue. We use this information to label each generated tissue image according to the cell type density, allowing us to visualize the representation learning properties of the model's latent space. We use Hover-Net on generated samples from a model trained on the breast cancer dataset. Hover-Net is a tool that identifies neoplastic epithelial, non-neoplastic epithelial, inflammatory, connective, and dead cells. **Figure 3.2** provides an example of how HoverNet identifies these types of cells in our generated images, where neoplastic epithelial cells are highlighted in red, non-neoplastic epithelial in orange, inflammatory in green, and connective cells in blue.

Finally, we create 8 different classes that account for counts of the different cell types in the tissue image, and consecutively we label each generated image with the corresponding class, allowing us to more clearly visualize the relation between cell type density in generated images and the model's latent space.

3.2.4 Tissue type assignation on generated images

The NCT colorectal cancer dataset provides a label along each 224×224 tissue sample and we make use of this information to label each generated image with a type of tissue, assigning a label to each generated image according to the 10-nearest neighbors of real images in the Inception-V1 feature space.

In **Figure 3.5** we show generated images on the left column and the nearest real neighbors on the remaining columns. We present the different tissue types: tumor (I), stroma (J), normal mucosa (K), muscle (L), mucus (M), lymphocytes (N), debris (O), and adipose (P). From this

example, we can conclude that distance in feature space gives a reliable reference to find the tissue type of generated images.

3.2.5 Pathologists' classification test

To demonstrate that the generated images can sustain the scrutiny of clinical examination, we asked expert pathologists to take a test, setup as follows:

- 50 Individual images - Pathologists were asked to rate all individual images from 1 to 5, where 5 meant the image appeared the most real.

We chose fake images in two ways, half of them were hand-selected and the other half were selected based on their smallest Euclidean distance to real images in the convolutional feature space (Inception-V1). Hand-selecting half of the fake images is motivated to ensure tissue diversity in the tests, certain types of cells such as lymphocytes are easier for the GAN to replicate and these generated images generally have small distance values to real images. All the real images are randomly selected between the three closest neighbors of the fake images. **Figure 3.3** and **Figure 3.4** show breast and colorectal cancer examples of tests taken by the pathologists.

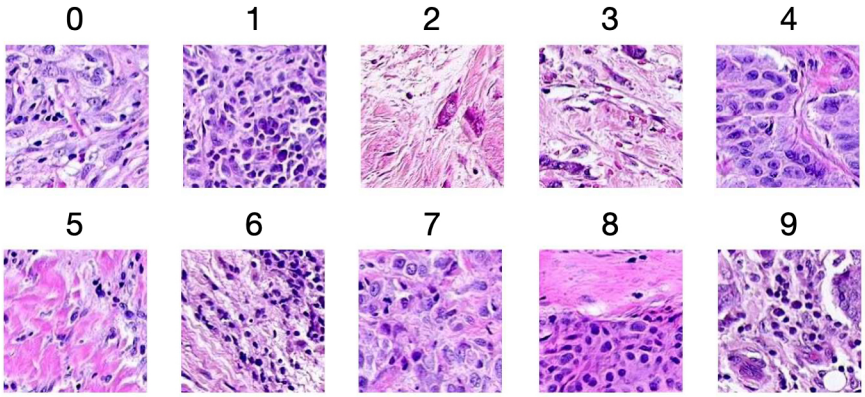


Figure 3.3: Pathologists test example of breast cancer tissue.

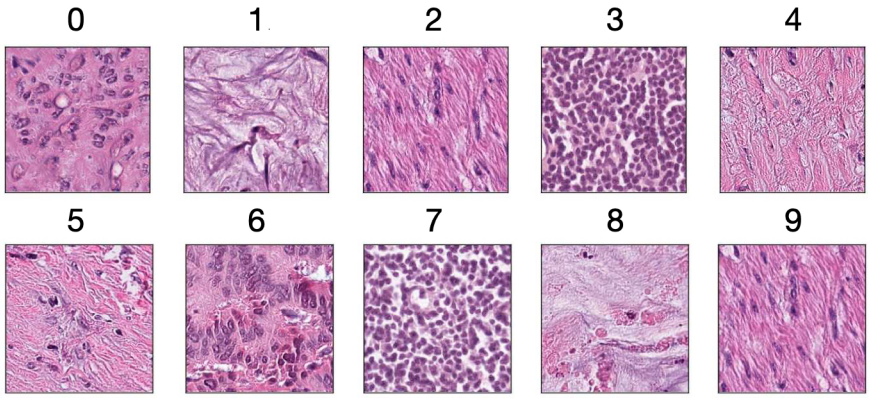


Figure 3.4: Pathologists test example of colorectal cancer tissue.

3.3 Results

3.3.1 Image quality analysis

We study the fidelity of the generated images and their distribution in relation to the original data through measures of FID score and by visualizing the closest neighbors between generated and real images. As explained in **Section 3.2.3**, we calculate Fréchet Inception Distance (FID) with the usual convolutional features of a pre-trained ImageNet Inception-V1.

Table 3.1 shows FID scores for a BigGAN with a hinge loss, BigGAN with the relativistic average discriminator loss, and PathologyGAN. BigGAN with a hinge loss is not able to reproduce features found in tissue and often suffers batch effect artifacts (**Appendix A.1**), in both cancer tissue types this model shows a high value of FID. BigGAN with the relativistic average discriminator loss shows the advantage of this loss approach showing a low FID value for both tissue types. PathologyGAN shows similar FID values to the previous model, it includes not only the relativistic average discriminator loss but also the mapping network and style mixing regularization for its representation learning properties. These results suggest that our model can capture cellular and morphological features found in different cancer tissues. As an alternative comparison value, we provide FID scores for different GAN models and datasets (ImageNet, FFHQ). It is worth mentioning that FID values are not directly comparable because it compares the distribution of features and these may vary for different datasets. Nevertheless, it can provide some coarse references to compare. Our previous FID values are similar to ImageNet models such as BigGAN[84] and SAGAN[87] with FIDs of 7.4 and 18.65, or FFHQ StyleGAN[85] with an FID of 4.40.

Model	FID Colorectal	FID Breast
BigGAN - Hinge Loss	87.55±3	98.5±22
BigGAN - Relativistic Loss	30.20±6	15.54±2
PathologyGAN	32.05±3	16.65±3

Table 3.1: PathologyGAN's FID performance evaluation. We report mean and standard deviations computed over three different random initializations for the breast and colorectal cancer datasets. BigGAN with a hinge loss is not able to reproduce features found in tissue and often suffers batch effect artifacts, in both cancer tissue types this model shows a high value of FID. BigGAN with the relativistic average discriminator loss shows the advantage of this loss approach showing a low FID value for both tissue types. PathologyGAN shows similar FID values to the previous model, it includes not only the relativistic average discriminator loss but also the mapping network and style mixing regularization for its representation learning properties. These results suggest that our model is able to capture cellular and morphological features found in different cancer tissues.

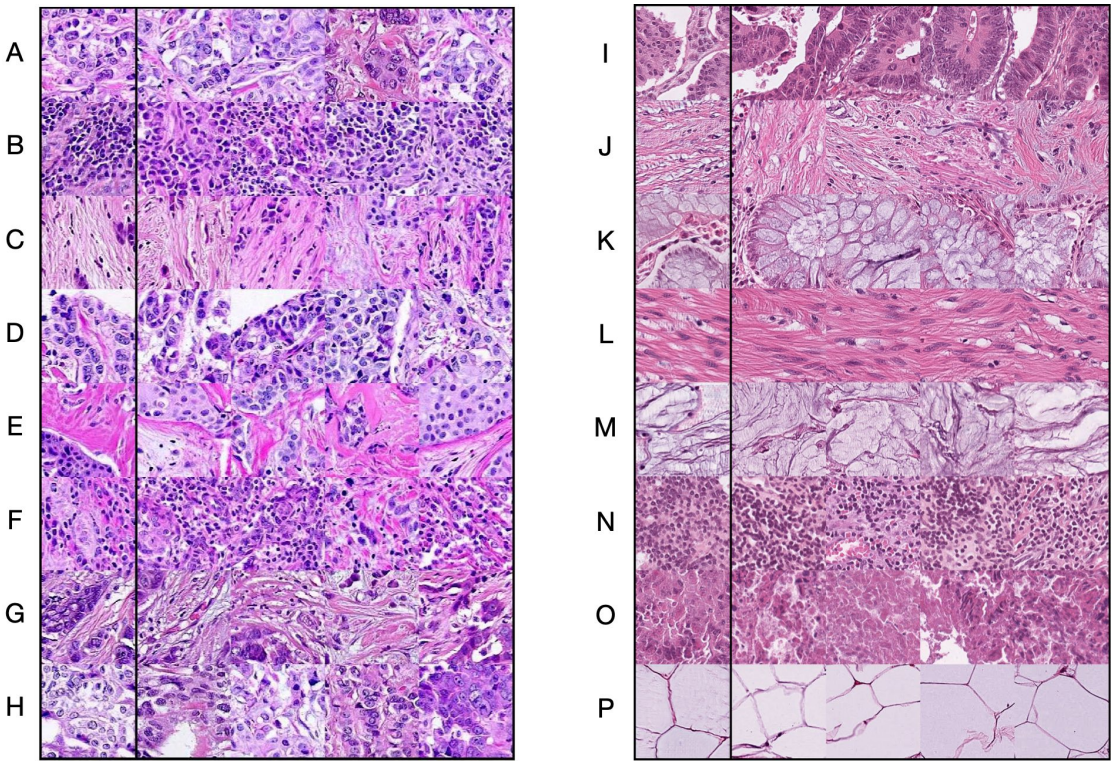


Figure 3.5: Nearest neighbors in Inception-V1 feature space for breast cancer (A-H) and colorectal cancer (I-P). For each row, the first column image corresponds to a generated tissue samples from PathologyGAN, the remaining columns are the closest real images in Inception-V1 feature space.

Additionally, in **Figure 3.5** we present samples of generated images (first column) and their closest real neighbors in Inception-V1 feature space, the images are paired by rows. (A-H) correspond to different random samples of breast cancer tissue. In the case of colorectal cancer, we provide examples of different types of tissue: tumor (I), stroma (J), normal mucosa (K), muscle (L), mucus (M), lymphocytes (N), debris (O), and adipose (P). We can see that generated and real images hold the same morphological characteristics.

3.3.2 Pathologists' results

Figure 3.6 shows the receiver operating characteristic (ROC) for pathologists' performance over real/fake classification for breast (A) and colorectal cancer tissue (B). We show how their performance is close to random classification. In addition, **Table 3.2** shows the mean and standard deviation of the pathologists' ratings for the fake and real images and Cohen's kappa score for inter-observer agreement between pathologists, where values of 1 shows complete agreement and 0 shows no agreement above chance. Colorectal and breast cancer tissue ratings for real and fake show almost equal value, suggesting that pathologists find no reliable artifact or feature to distinguish between real and fake. However, kappa scores for real and fake images and cancer types show no clear agreement above chance. It is relevant to mention that pathologists' usual procedure is to work with larger images and resolution, common feedback from pathologists was

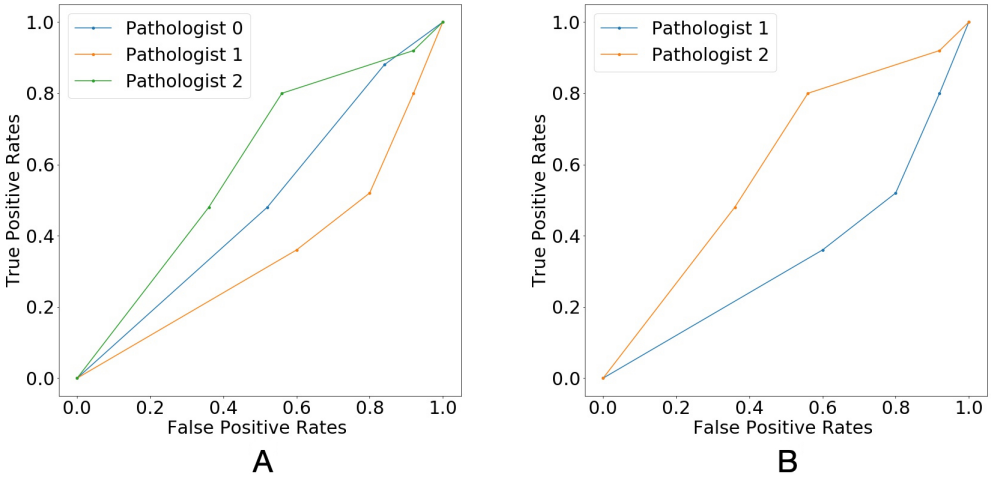


Figure 3.6: Receiver operating characteristic (ROC) curve of pathologists' performance over real versus fake classification for breast (A) and colorectal cancer tissue (B). The near-random classification performance from expert pathologists suggests that generated tissue images do not present artifacts that give away the tissue as generated.

	Rating (1-5) Mean/Std	Kappa Score Path. Agreement
Colorectal Cancer		
Fake	3.7±1.2	0.17
Real	3.6±1.3	0.05
Breast Cancer		
Fake	4.13±0.95	0.13
Real	4.13±0.98	0.26

Table 3.2: Pathologists' ratings and kappa scores on PathologyGAN's images. We report mean and standard deviations for pathologists' ratings on real and fake images. In addition, we report Cohen's kappa scores to account for pathologists' agreement on image ratings. Colorectal and breast cancer tissue ratings for real and fake show almost equal value, suggesting that pathologists find no reliable artifact or feature to distinguish between real and fake. However, kappa scores for real and fake images and cancer types show no clear agreement above chance. We theorize that different pathologists may have leaned on different features to make the unreliable rating and therefore found no agreement on the classification.

that the generated fake images were of such a quality that at the 224×224 resolution, they were not able to differentiate between real and fake tissue. We theorize that different pathologists may have leaned on different features to make the unreliable rating and therefore found no agreement on the classification.

3.3.3 Analysis of latent representations

In this section we focus on the PathologyGAN's latent space, exploring its representation learning properties. Here we will provide examples of its impact on linear interpolations and vector operations on the latent space w, as well as visualizations of the latent space w.

Figure 3.7 and **Figure 3.8** capture how the latent space w has a structure that shows a direct relationship with tissue properties. To create these figures, we generated $10K$ images and labeled them accordingly to **Section 3.2.3** and **Section 3.2.4,** along with each tissue image we also have the corresponding latent vector w and we used UMAP[145] to project them to a two-dimensional space.

Figure 3.7 reveals the relationship between the number of different cell types in tissue and regions of the latent space. We show examples of neoplastic epithelial, non-neoplastic epithelial, inflammatory, and connective cell density where class 0 corresponds to the lowest density of the cell type and cell 7 to the highest. **Figure 3.8** displays how the distinct regions of the latent space generate different kinds of tissue. These examples provide evidence of a structured latent space according to tissue cellular characteristics and tissue type.

We also found that linear interpolations between two latent vectors w have better feature transformations when the mapping network and style mixing regularization are introduced. **Figure 3.9** shows linear interpolations in latent space w between tissue images with none or few tumor cells and high number of tumor cells. (A, C) correspond to a model with a mapping network and style mixing regularization and (B, D) to a model without those features, we can see that transitions on (A, C) include an increasing population of cancer cells rather than the fading effect observed in images of (B, D). This result indicates that (A, C) better translates interpolations in the latent space, as real cells do not fade away.

In addition, we performed linear vector operations in w that translated into semantic image features transformations[95]. **Figure 3.10** provides examples of six vector operations that result in feature alterations in the images. By performing these linear vector operations, we make the underlying assumption that PathologyGAN's latent space is an Euclidean space. However, more complex vector operation and feature transformations might break this assumption. Nevertheless, this evidence shows further support for the relation between a structured latent space and tissue characteristics.

Finally, we explored how linear interpolation and vector operations translate into the individual points in the latent space. In **Figure 3.11** we provide examples of interpolations from stroma to tumor in colorectal cancer, and tumor to lymphocytes in breast cancer. Through the intermediate vectors, we show that gradual transitions in the latent space translate into smooth feature transformations, in these cases increase/decrease of tumor cells or lymphocytes. Alternatively, **Figure 3.12** shows how results from vector operations fall into regions of the latent space that correspond to the expected tissue type or a combination of features such as tumor and lymphocytes. With these figures, we visualize the meaningful representations of cells and tissue types,

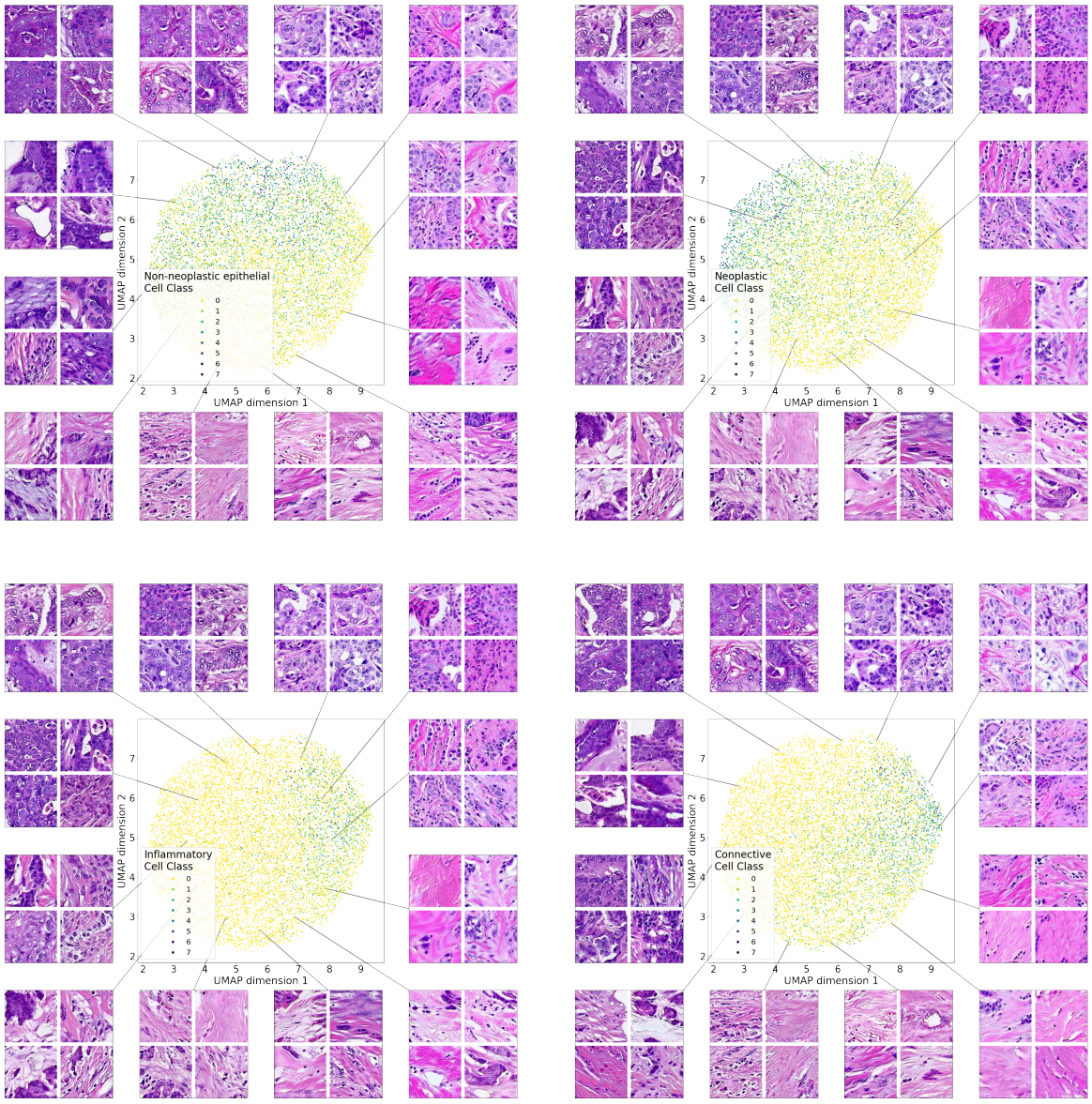


Figure 3.7: Latent space of PathologyGAN trained on breast cancer tissue from Netherlands Cancer Institute (NKI) and Vancouver General Hospital (VGH) dataset. Uniform Manifold Approximation and Projection (UMAP) representations of generated tissue samples. We label each image and representation pair based on neoplastic epithelial, non-neoplastic epithelial, inflammatory, and connective cell density where class 0 corresponds to the lowest density of the cell type and cell 7 to the highest. Different regions of the latent space defined by the representations correspond to different densities of cell types.

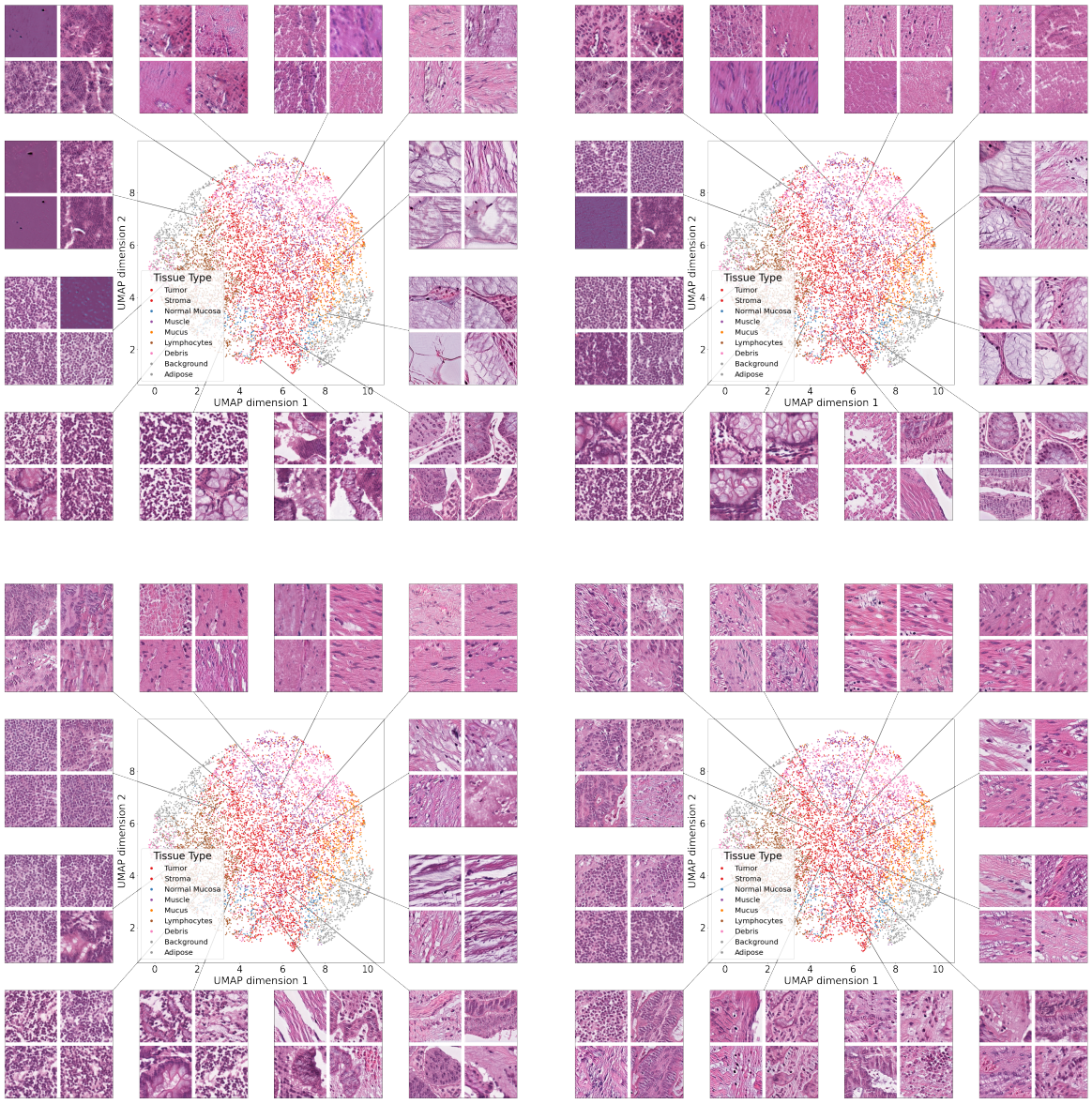


Figure 3.8: Latent space of PathologyGAN trained on colorectal cancer tissue from National Center for Tumor (NCT) dataset. Uniform Manifold Approximation and Projection (UMAP) representations of generated tissue samples, each generated image is labeled with the type of tissue. Different regions of the latent space generate distinct kinds of tissue.

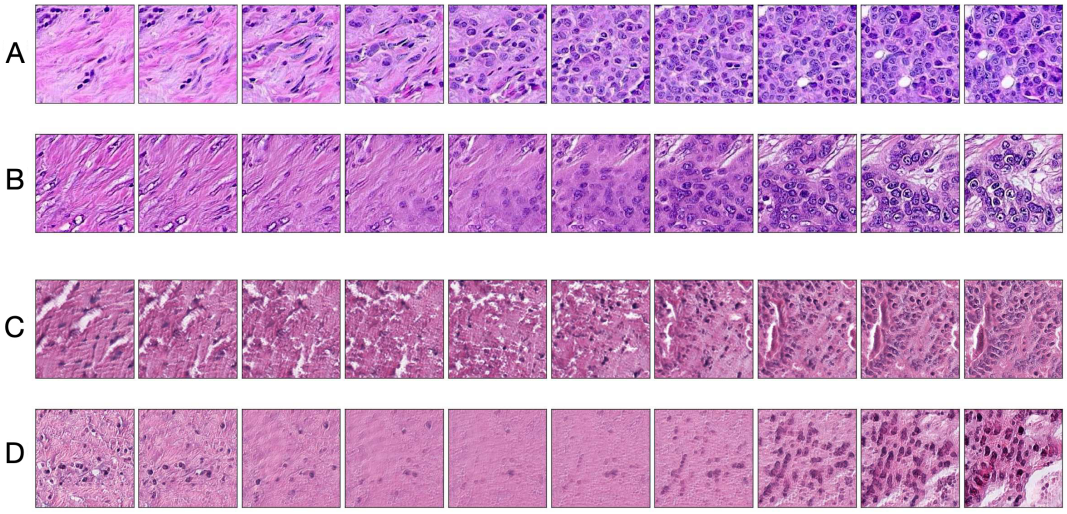


Figure 3.9: Linear interpolation in the latent space w from few or none tumor cells (left end) to more tumor cells (right end) on breast cancer (A, B) and colorectal cancer (C, D). (A, C) PathologyGAN model interpolations with a mapping network and style mixing regularization. (B, D) PathologyGAN model interpolations without a mapping network and style mixing regularization. (A, C) includes an increasing population of epithelial cells rather than a fading effect from model (B, D), this shows that model (A, C) better translates high-level features of images from latent space vectors.

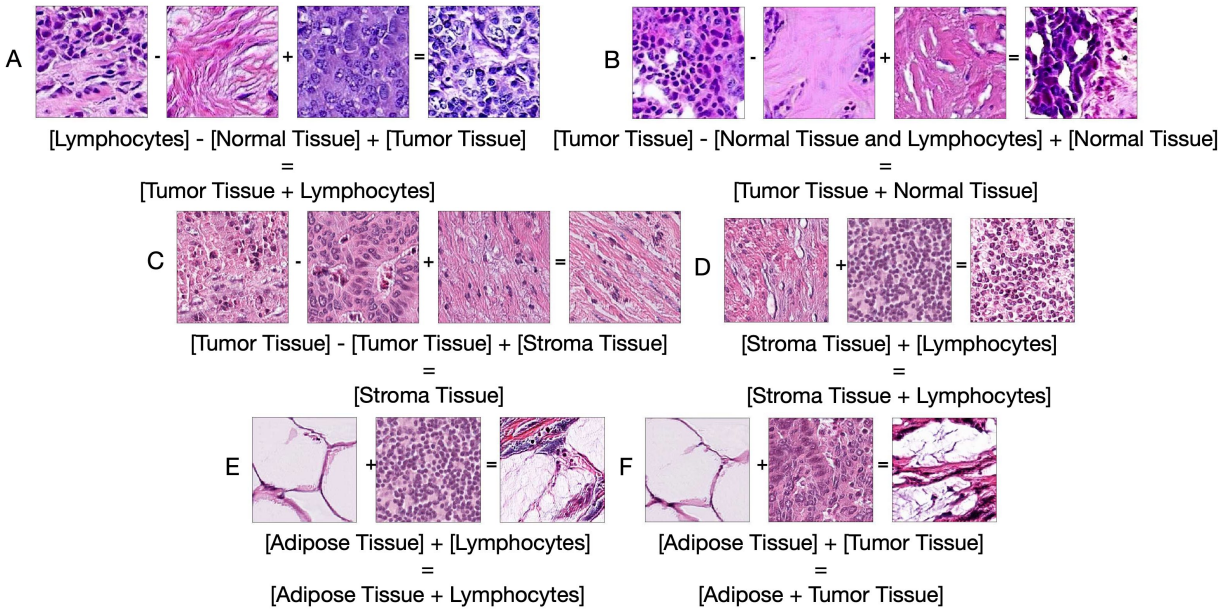


Figure 3.10: Linear vector operations on the latent space w translate into image feature transformations. We gather latent vectors that generate images with different high-level features and perform linear operations on the vectors before we feed the generator, resulting in semantic translations of the characteristics of the images. We perform the arithmetic operations (A-B) on breast cancer tissue and (C-F) on colorectal cancer tissue.

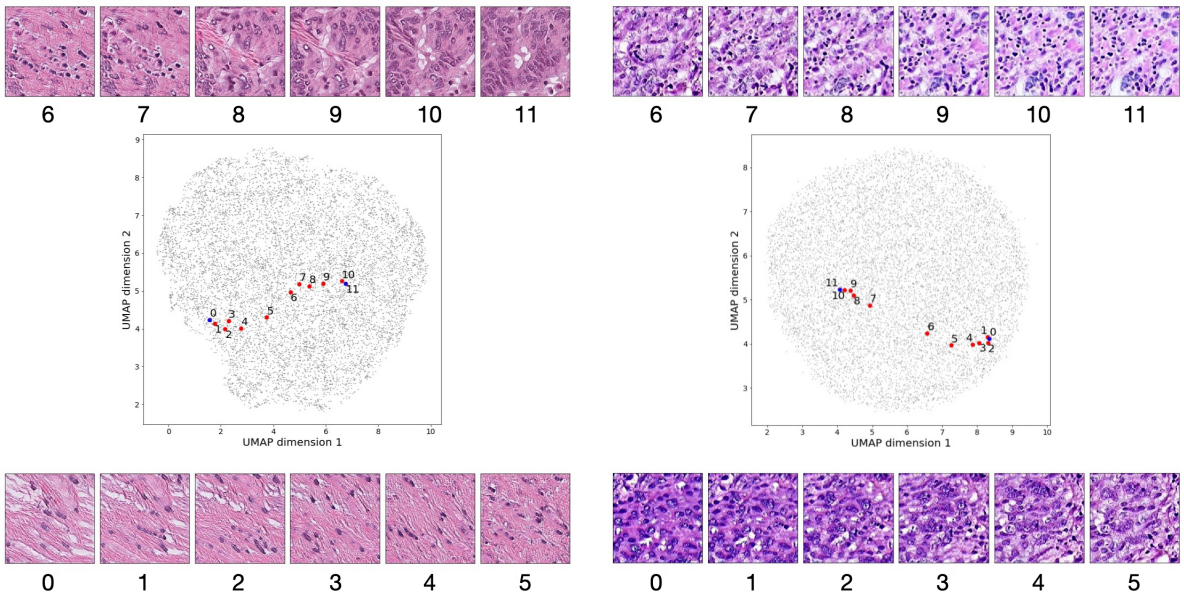


Figure 3.11: Uniform Manifold Approximation and Projection (UMAP) of linear interpolations in PathologyGAN's latent space w and associated images. Colorectal cancer (left) shows a transition between stroma and tumor, while brest cancer (right) shows a transition between tumor and lymphocytes. Starting vectors are colored in blue while intermediate points of the interpolations are colored in red. Through the intermediate vectors, we show that gradual transitions in the latent space translate into smooth feature transformations, increase/decrease of tumor cells, or increase of lymphocyte counts.

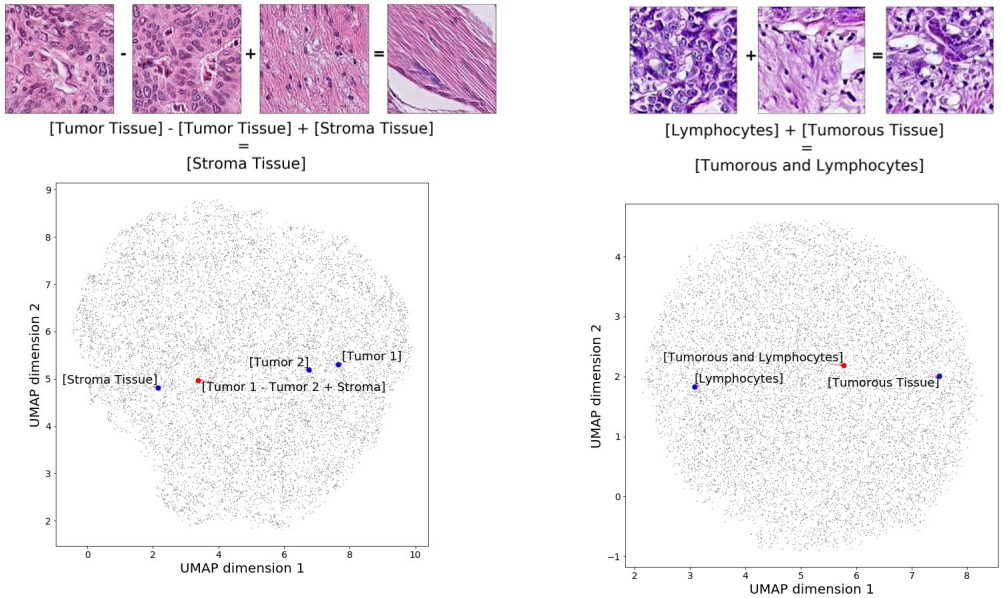


Figure 3.12: Uniform Manifold Approximation and Projection (UMAP) of linear vector operations of representations and associated tissue images. Original vectors are colored in blue while the results are colored in red. We show colorectal cancer examples (left) and breast cancer examples (right). After vector operations, the results fall into regions of the latent space that correspond to the tissue type (left) or a combination of features. E.g. tumor, lymphocytes, or stroma cells.

and also the interpretability of the latent space. **Appendix A.4** contains additional examples of these visualizations.

3.3.4 Mapping network and style mixing regularization impact on representations

We measure the impact of introducing a mapping network and using style mixing regularization on the model's latent space and its representation learning properties. In order to do so, we compare visualizations of the latent space w for two PathologyGANs, one using these features and another without them. We include comparisons for datasets, breast and colorectal cancer.

Figures 3.13-3.16 capture the clear difference in the latent space ordering with respect to cell type density and tissue type in the generated image. Without a mapping network and style mixing regularization, the latent space w shows a random placement of the vectors subject to tissue characteristics, when these two elements are introduced different regions of the latent space produce images with distinct tissue types and cellular densities.

3.3.5 Dataset size impact on representations

To further understand PathologyGAN's performance with respect to dataset size, we subsample the NCT dataset and created subsets of 5K, 10K, and 20K images.

We measure its ability to generalize and hold meaningful representations in two ways: by measuring the FID against the complete dataset of 100K samples and by exploring the latent space structure.

PathologyGAN shows an ordered latent space based on tissue type even for 5K samples (**Figure 3.17**). However, it is not able to reliably generalize the original dataset distribution. **Table 3.3** shows how 20K samples are enough to achieve a reasonable FID of 41.92.

NCT Datase Size	FID
5K	83.52
10K	56.03
20K	41.92
Complete 100K	32.05

Table 3.3: PathologyGAN's FID performance for different training sample sizes of the NCT cohort.

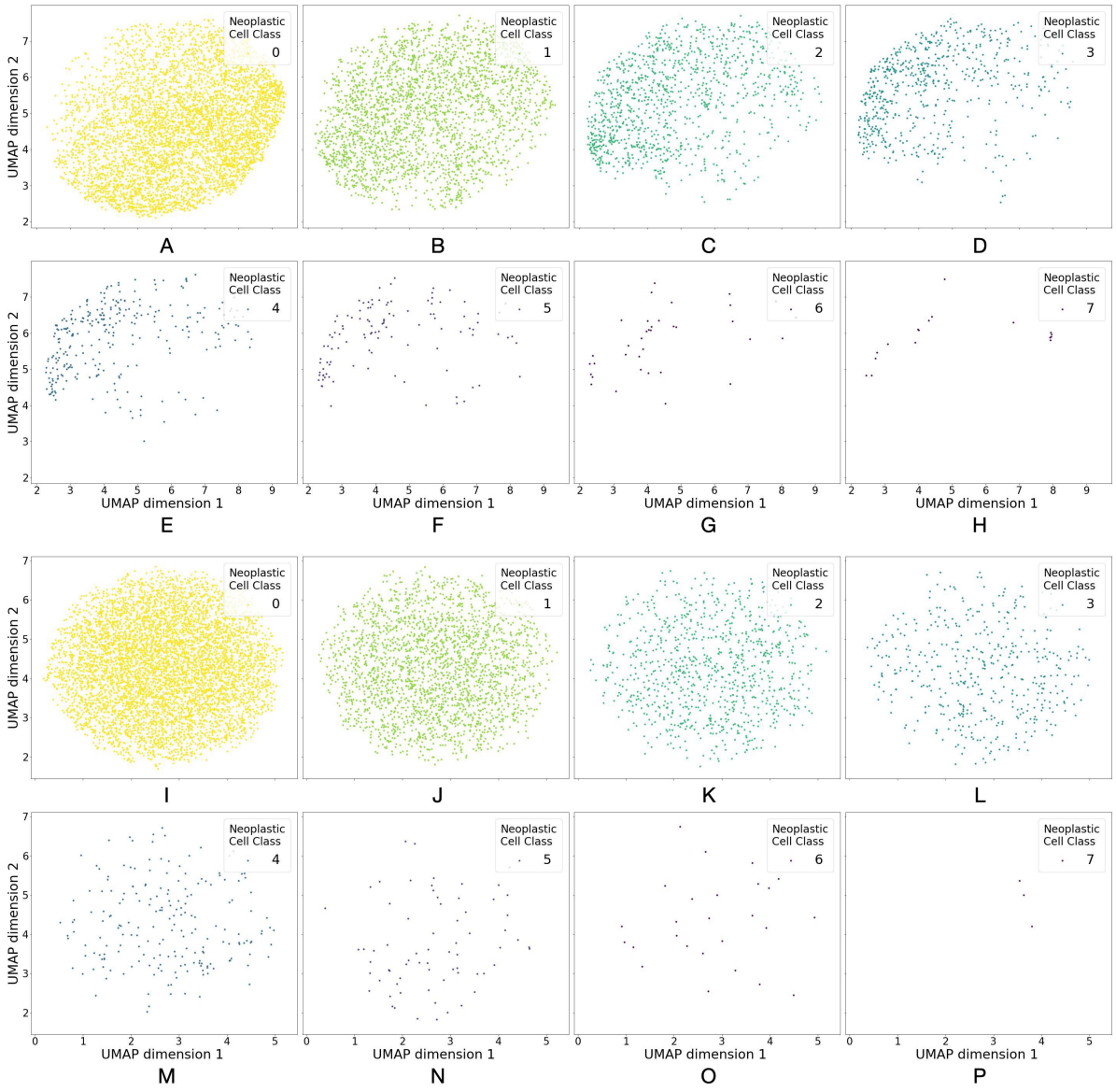


Figure 3.13: Breast cancer tissue: Scatter plot comparison of the latent space w for two different PathologyGAN models. Model (A-H) includes a mapping network and style mixing regularization, model (I-P) does not include them. Each subfigure shows data points only related to one of the classes where each class is subject to the density of neoplastic epithelial cells in the tissue image. Class 0 representations are associated with images with the lowest density of neoplastic epithelial cells (A,I) while class 7 representations are associated with the largest (H,P). In model (A-H), representations with an increasing number of cancer cells in the image correspond to proportionally moving from quadrant IV to II in the 2-dimensional space. In contrast, model (I-P) representations are randomly placed. This figure shows how including the mapping network and style mixing regularization introduces representation learning properties.

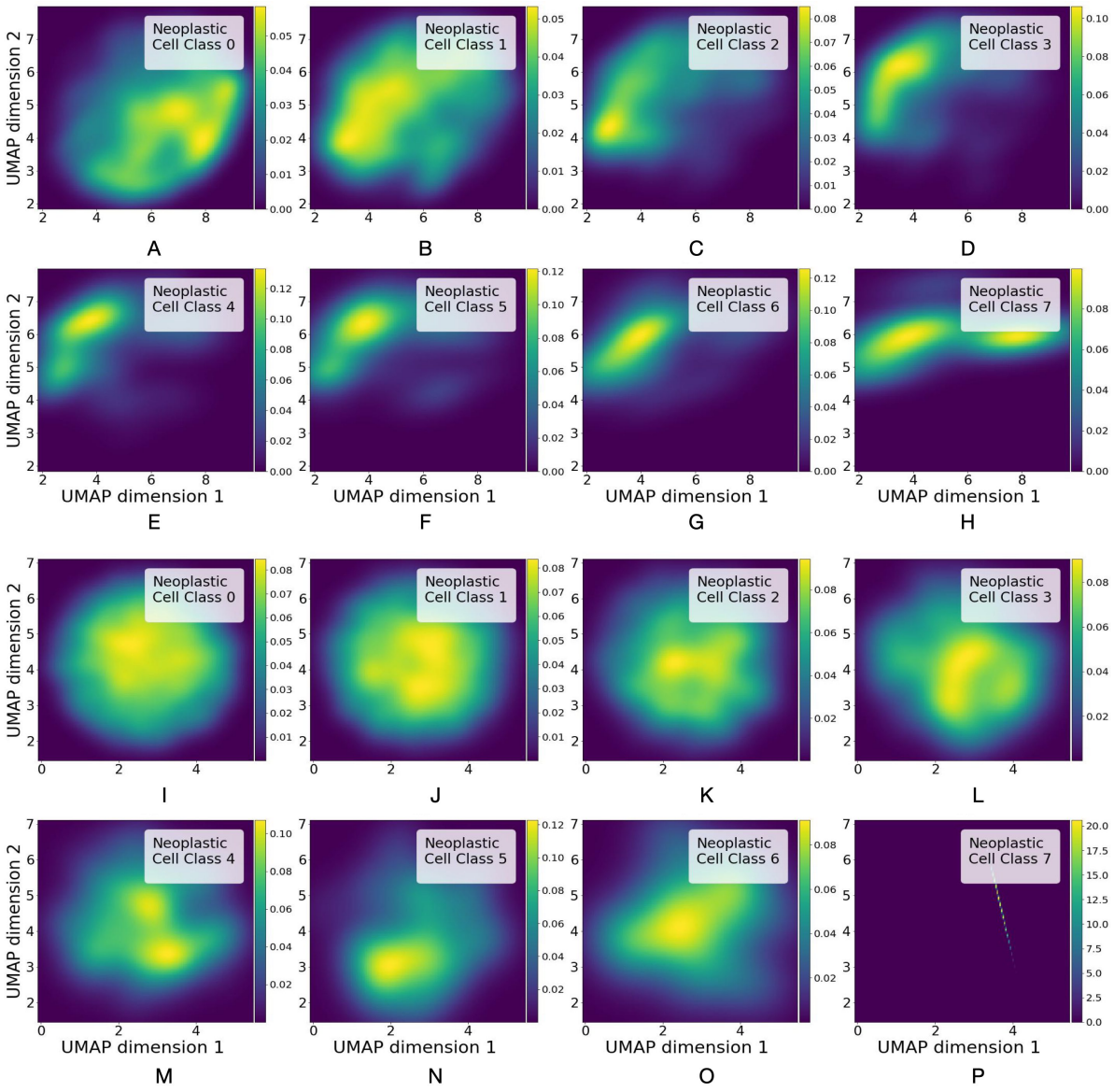


Figure 3.14: Breast cancer tissue: Density plot comparison of the latent space w for two different PathologyGAN models. Model (A-H) includes a mapping network and style mixing regularization, model (I-P) does not include them. Each subfigure shows data points only related to one of the classes where each class is subject to the density of neoplastic epithelial cells in the tissue image. Class 0 representations are associated with images with the lowest density of neoplastic epithelial cells (A,I) while class 7 representations are associated with the largest (H,P). In model (A-H), representations with an increasing number of cancer cells in the image correspond to proportionally moving from quadrant IV to II in the 2-dimensional space. In contrast, model (I-P) representations are randomly placed. This figure shows how including the mapping network and style mixing regularization introduces representation learning properties.

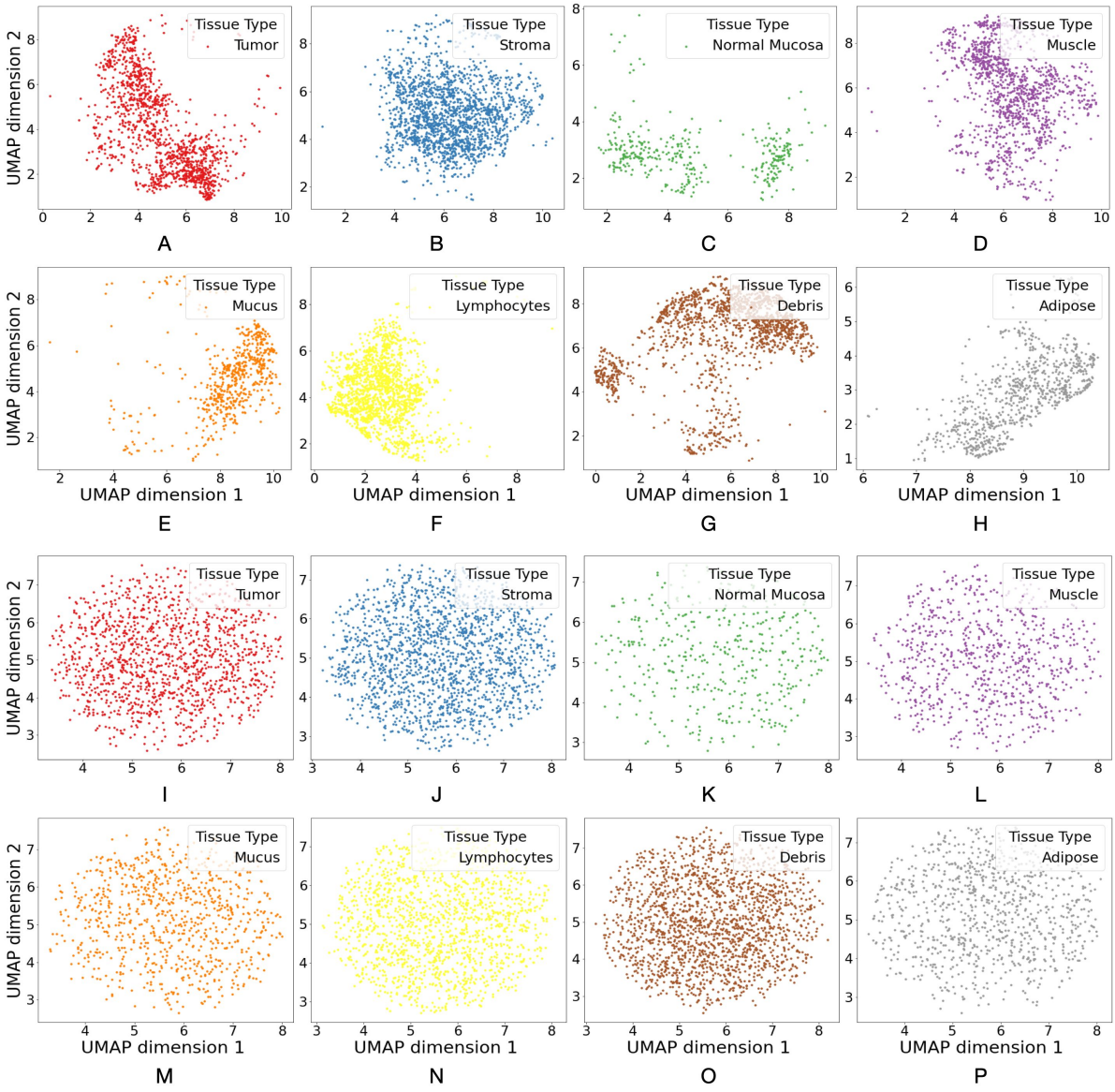


Figure 3.15: Colorectal cancer tissue: Scatter plot comparison of the latent space w for two different PathologyGAN models. Model (A-H) includes a mapping network and style mixing regularization, model (I-P) does not include them. Each subfigure shows data points only related to one type of tissue. Model (A-H) shows how distinct regions of the latent space correspond to different tissue types while in the model (I-P) the representations are randomly placed. This figure shows how including the mapping network and style mixing regularization introduces representation learning properties.

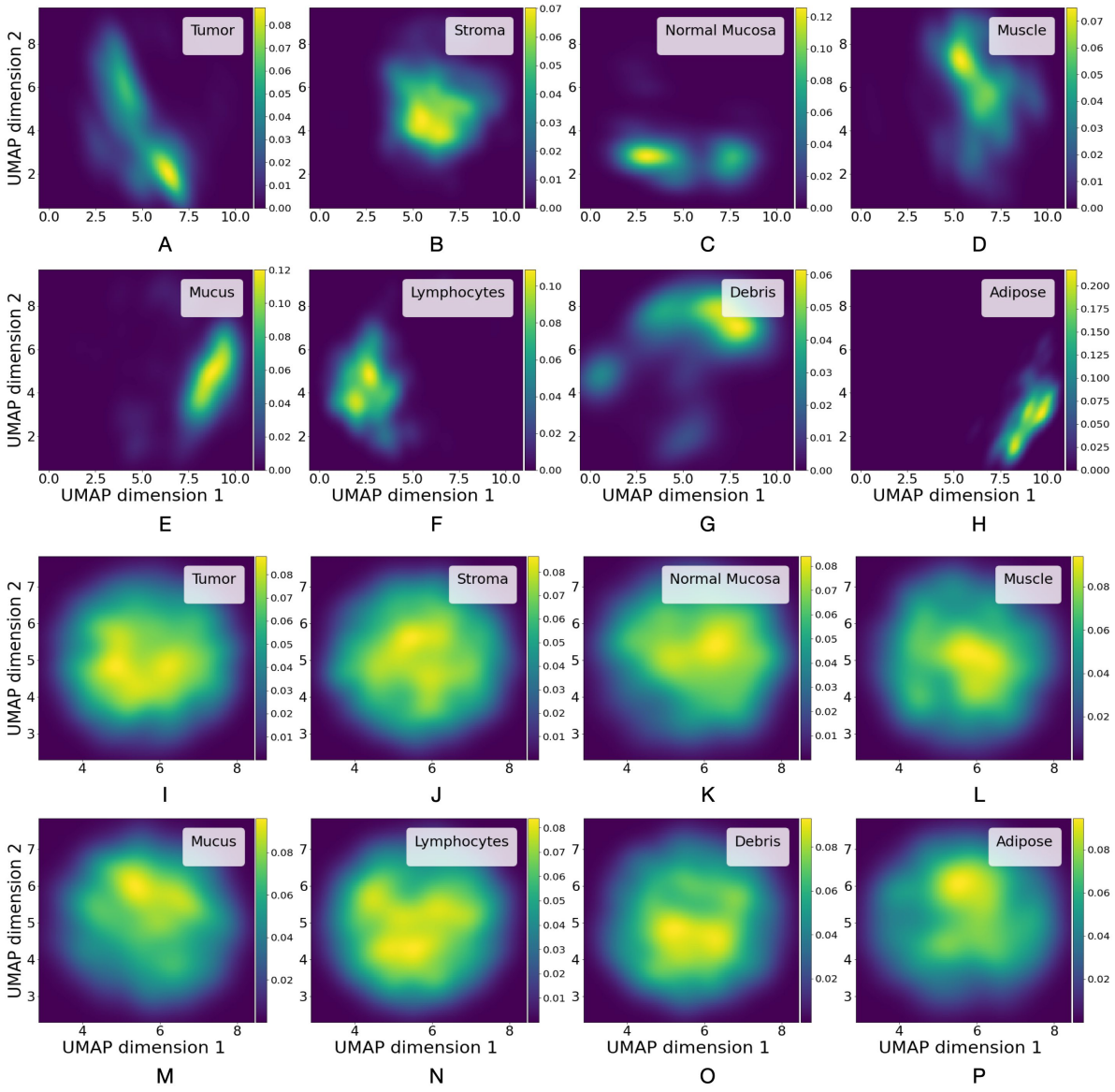


Figure 3.16: Colorectal cancer tissue: Density plot comparison of the latent space w for two different PathologyGAN models. Model (A-H) includes a mapping network and style mixing regularization, model (I-P) does not include them. Each subfigure shows data points only related to one type of tissue. Model (A-H) shows how distinct regions of the latent space correspond to different tissue types while in the model (I-P) the representations are randomly placed. This figure shows how including the mapping network and style mixing regularization introduces representation learning properties.

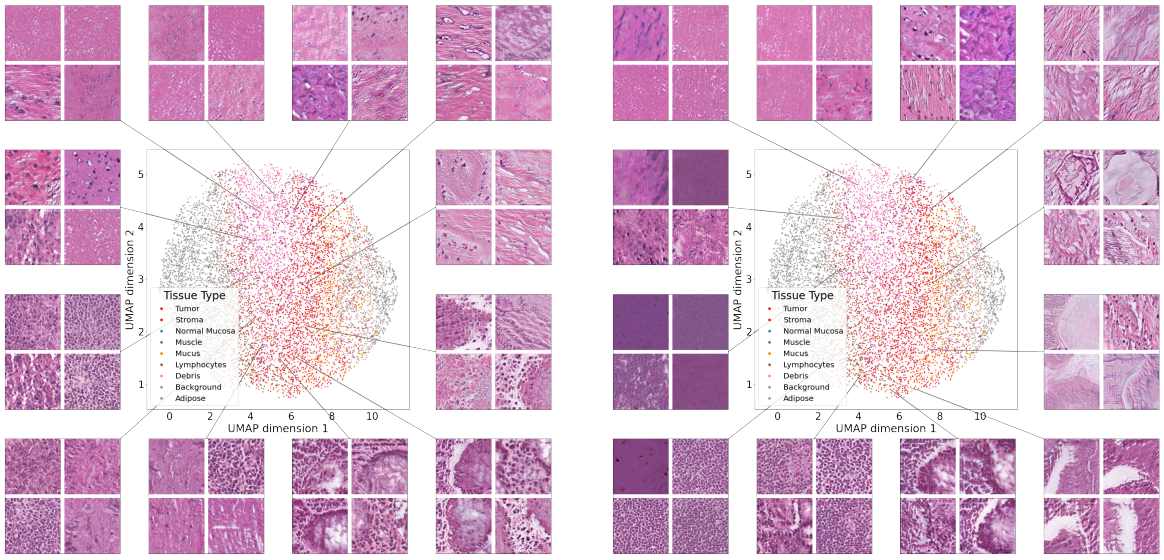


Figure 3.17: PathologyGAN's latent space trained with 5K samples of the NCT dataset. Uniform Manifold Approximation and Projection (UMAP) representations of generated tissue samples, each generated image is labeled with the type of tissue. Even with $5K$ samples, PathologyGAN holds an structured latent space where different regions generate distinct kinds of tissue.

3.4 Conclusion

Throughout this chapter, we have presented a generative adversarial network that can replicate and create representations based on tissue architectures and cellular characteristics. For this reason, our results are focused on testing two features of our model: Image quality of generated tissue and interpretability of tissue representations.

We tested the model's ability to learn and reproduce the distribution of different tissue types and cellular information. The FID results in **Table 3.1** show that PathologyGAN does reproduce these distributions when it is judged by convolutional features (32.05/16.65). We would like to highlight that the FID score measures the distribution difference between real and generated samples, and given the low values, we argue that the model is also capturing the abundance or scarcity of different tissue patterns.

Figure 3.5 shows generated tissue samples and their closest real neighbors in the Inception-V1 convolutional space, allowing us to visually inspect the similarity between generated and real tissue samples. In the colorectal cancer samples, we can see that the different types of tissue, from tumor (I) to adiose (P), clearly resemble the real tissue. Breast cancer tissue samples (A-H) also show the same result where the real and generated samples present the same patterns and shapes.

Furthermore, we tested the generated images of PathologyGAN against pathologists' interpretations. We aimed to test that the generated tissues do not contain any artifacts that give them away as fake through the eyes of professionals. Given the near-random performance between real and fake classification in **Figure 3.6** and that the average rating for real and fake images is almost equal in **Table 3.2**. We conclude that generated samples are realistic enough to pass as

real through the examination of pathologists. We argue this is relevant because the model is able to reproduce tissue patterns to which pathologists are accustomed.

We conclude that the model is able to reproduce the detail and distribution of tissue not only from the convolutional features but also by pathologists' interpretations.

In relation to PathologyGAN's latent space structure and interpretability of its representations, **Figure 3.7** and **Figure 3.8** show how distinct regions of the latent space hold tissue and cellular information directly related to the generated images. **Figure 3.7** shows a clear relationship between regions of the latent space and different cell type densities in the tissue, while **Figure 3.8** links the region of the space with the different tissue types (e.g. tumor, stroma, normal mucosa, muscle, lymphocytes, debris, mucus, adipose, and background). Furthermore, **Figures 3.13-3.16** provide examples of the impact of the mapping network and style mixing regularization on the representation learning properties of the model. We also studied the necessary size of a dataset that can be used to train our model showing that a sample of 20K instances is enough to achieve a reasonable FID and representation learning properties.

In **Figure 3.9** and **Figure 3.11**, we relate linear interpolations between latent vectors and generated tissue attributes, showing that gradual transitions in the latent space transfer into smooth feature transformations, in these cases increase/decrease of tumor cells or increase of lymphocyte counts. Finally, **Figure 3.10** and **Figure 3.12** provide examples of linear vector operations and their translation into tissue characteristic changes, with different tissue changes. These results provide support for the representation learning properties of the model, not only holding meaningful information of cell and tissue types but also an interpretable insight into the representations themselves.

Chapter 4

Adversarial learning of cancer tissue representations

4.1 Introduction

As introduced in the previous chapter, PathologyGAN is a generative adversarial network (GAN) that can generate high-fidelity tissue images from a structured and interpretable latent space. However, it does not offer the ability to project real tissue images to its latent space, only offering latent representations for synthesized tissue.

In this chapter, we will focus on tackling this limitation from PathologyGAN and propose to modify the model by including an encoder that projects images onto the GAN's latent space; allowing to find tissue representations of real images. We show evidence that PathologyGAN's tissue representations carry meaningful information related to tissue and illustrate their applicability in three different scenarios:

- We visualize the tissue representations of real images and show how distinct regions of the latent space enfold different characteristics. Additionally, we show how reconstructed tissue from a latent representation follows the same characteristics as the original image.
- We quantify how informative tissue representations are by training a linear classifier to predict tissue types and compare them to other supervised approaches.
- We demonstrate the suitability of the tissue representations on a multiple instance learning (MIL) task, predicting presence of tumor on WSIs.

Through these applications, we aim to validate that the unsupervised tissue representations from PathologyGAN can be used in different real application tasks and contexts.

4.2 Methods

4.2.1 PathologyGAN and real tissue image representations

We build upon PathologyGAN and introduce an encoder E_{θ_e} that maps back generated tissue to the GAN's latent space. Effectively learning to interpret tissue morphologies and acting as the inverse of the generator G_{θ_g}. Af ter training, the encoder can be used independently to map real tissue samples to their representations in the latent space. **Figure 4.1** captures the high-level architecture of the model. We propose this approach instead of iterative methods that update latent vectors of a pre-trained GAN to project a real image into its latent space as they are considerably slower[146,147]. Iterative projection methods require several iterations per real image compared to one forward pass of the encoder.

The loss functions of PathologyGAN remain unmodified. The discriminator's L_{Dis} and generator's losses L_{Gen} are defined as the Relativistic Average Discriminator[96] where the discriminator's goal is to estimate the probability of the real data being more realistic than the fake data. **4.1** describes PathologyGAN's learning objective where p_{data} is the distribution of real data, $p_g(x)$ is the distribution of synthetic data produced by the Generator G_{θ_g}, $C_{\theta_d}(x)$ is the non-transformed discriminator output or critic, w the transformed latent space with representation learning properties, and σ denotes the sigmoid function:

$$
\begin{aligned}
L_{Dis} &= -\mathbb{E}_{x\sim p_{data}(x)}\left[\log\left(D_{\theta_d}(x)\right)\right] - \mathbb{E}_{\tilde{x}\sim p_g(x)}\left[\log\left(1 - D_{\theta_d}(\tilde{x})\right)\right], \\
L_{Gen} &= -\mathbb{E}_{\tilde{x}\sim p_g(x)}\left[\log\left(D_{\theta_d}(\tilde{x})\right)\right] - \mathbb{E}_{x\sim p_{data}(x)}\left[\log\left(1 - D_{\theta_d}(x)\right)\right], \\
D_{\theta_d}(x) &= \sigma\left(C_{\theta_d}(x) - \mathbb{E}_{\tilde{x}\sim p_g(x)}[C_{\theta_d}(\tilde{x})]\right), \\
D_{\theta_d}(\tilde{x}) &= \sigma\left(C_{\theta_d}(\tilde{x}) - \mathbb{E}_{x\sim p_{real}(x)}[C_{\theta_d}(x)]\right), \\
&\text{where } \tilde{x} = G_{\theta_g}(w),\ w = M_{\theta_m}(z),\ \text{and } z \sim \mathcal{N}(0, \mathbb{I})
\end{aligned}
\tag{4.1}
$$

The encoder loss function L_{Enc} (**4.2**) minimizes the mean square error between latent vectors w and their reconstruction from generated images $w' = E_{\theta_e}(G_{\theta_g}(w))$:

$$
\begin{aligned}
L_{Enc} &= \mathbb{E}_{z\sim P_z}\left[\frac{1}{n}\sum_{i=1}^{n}(w_i - w_i')^2\right] \\
&\textit{where } w' = E_{\theta_e}(G_{\theta_g}(w)),\ w = M_{\theta_m}(z),\ \text{and } z \sim \mathcal{N}(0, \mathbb{I})
\end{aligned}
\tag{4.2}
$$

Although the encoder E_{θ_e} is simultaneously trained with the GAN model, we can separate the model training into two parts: The mapping network M_{θ_m}, generator G_{θ_g}, and discriminator D_{θ_d} that are trained as a GAN, and the encoder E, which is trained to project back the generated tissue images onto the latent space. In practice, the encoder E_{θ_e} learns with the Generator G_{θ_g}. We trained our encoder based on the assumption that the generator is successful in reproducing cancer tissue. Therefore the encoder will learn to project real tissue images if it is able to do so

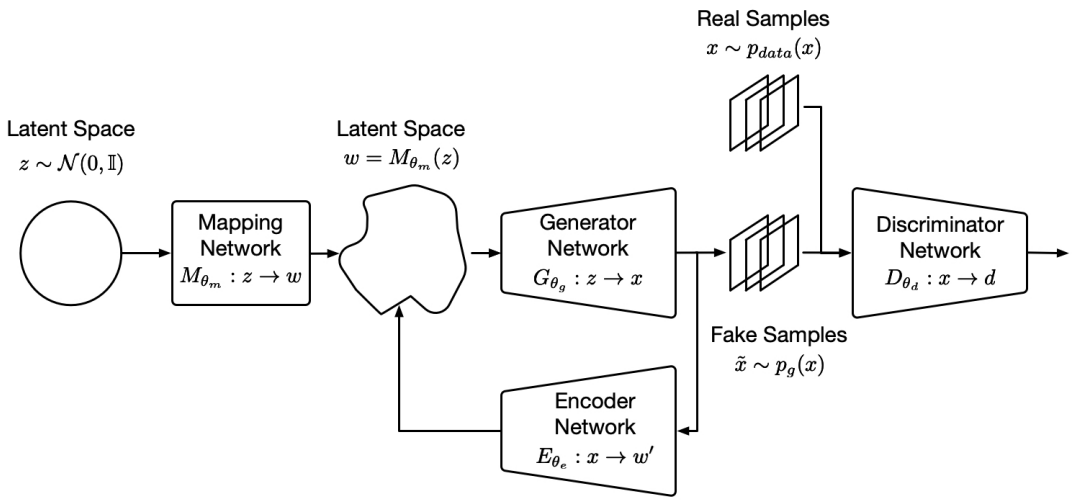


Figure 4.1: High level architecture of proposed modification of PathologyGAN. We modify the model by including an encoder that projects images onto the GAN's latent space.

with generated ones. Based on this logic, we use only generated images to train the encoder.

Additionally, the encoder is only updated when the generator is not trained with style mixing regularization[85]. Style mixing regularization is a technique that promotes disentanglement where the generator G_{θ_g} generates an image from two different latent vectors w_1 and w_2, these vectors are fed at different layers of the generator G_{θ_g}. However, it becomes impractical to train the encoder in these steps because these images have no clear assignation in the latent space w. Therefore, style mixing regularization is only performed 50% of the time in the generator training, so our encoder is updated every two steps per one of the generator.

We trained our model on the breast, colorectal, and lung cancer datasets (**Section 1.5**). In all three cases, we only use training sets to train our GAN model. We later project the real tissue patches of the training and test sets by using our encoder and work with latent representations for the three different applications.

4.3 Results

4.3.1 Visualizations of tissue representations and reconstructions

We first aim to test how interpretable tissue representations of real images are and if they capture enough information to recreate tissue with the same characteristics. We do so by visualizing the tissue representations along with the corresponding real images and by reconstructing images from the representations themselves.

We use the breast cancer and colorectal cancer datasets for these results. As introduced in **Chapter 3**, the breast cancer dataset is annotated with Hover-Net[19], providing quantitative measures of neoplastic and non-neoplastic epithelial, inflammatory, and connective cells. In addition, the colorectal tissue images have an associated label with the corresponding tissue

type. In both cases, we used the training set to train our GAN model and later used the encoder E_{θ_e} to project images from test sets, obtaining tissue representations of real samples.

Figures 4.2-4.6 display PathologyGAN's tissue representations and compares them to pre-trained ImageNet Inception-V1's representations. As previously covered in the **Introduction**, pre-trained models have been shown to capture useful tissue information. In these images, we used UMAP[145] to reduce the dimensionality of tissue representations to 2 dimensions and plot each vector with its associated tissue image and label. **Figures 4.2-4.4** compare tissue representations for breast cancer tissue, while PathologyGAN's representations show a more localized neoplastic and non-neoplastic epithelial cell density, Inception-V1's representations display more confined inflammatory and connective cell density. **Figure 4.6** compares tissue representations for colorectal cancer tissue. In this case, Inception-V1's representations show a clearer distinction between tissue types compared to PathologyGAN's representations. However, PathologyGAN holds generative abilities where Inception-V1 does not. In both datasets, these figures show how PathologyGAN's representations capture tissue characteristics of real images, whether it is tissue type or different cell type densities.

In addition, we compare real images to their synthetic reconstructions. To do so, we use the generator to synthesize images from the tissue representation of each real sample $x_{recon} = G_{\theta_g}(E_{\theta_e}(x_{real}))$. **Figure 4.7** shows the original real images and their associated reconstructions, we use representations of real images (A) to synthesize their reconstructions (B) with the GAN's generator. We provide examples of the colorectal cancer tissue types (0-4) and different cancer cell densities in breast cancer tissue (5-9). We can see that the reconstructions follow the characteristics of the real tissue images. We provide further samples of tissue reconstructions in the **Appendix B.1**.

4.3.2 Tissue type classification over tissue representations

In this task, we aim to quantify how informative the representations are, verifying that the model is able to learn tissue patterns that define tissue type. We train a linear classifier to predict tissue type over representations.

We use the colorectal cancer dataset with each tissue sample and its associated type label. We train our GAN with the training set and again use the encoder E_{θ_e} to project the training and test sets onto the GAN's latent space, obtaining representations for each tissue image. Consecutively, we use the tissue representations of the training set to train a logistic regression (one-versus-rest) and evaluate performance using test set representations.

Table 4.1 show the performance of the logistic regression, reaching an average AUC (one-vs-rest) of 0.976 and multi-class accuracy of 85.43% just from representations. **Table 4.2** provides a comparison between other existing supervised methods, such as Raczkowski *et al.*[63] where a Bayesian deep neural network is trained for tissue type classification achieving an AUC of 0.992 and 92.44% accuracy. In addition, we provide a comparison with a linear classifier trained over

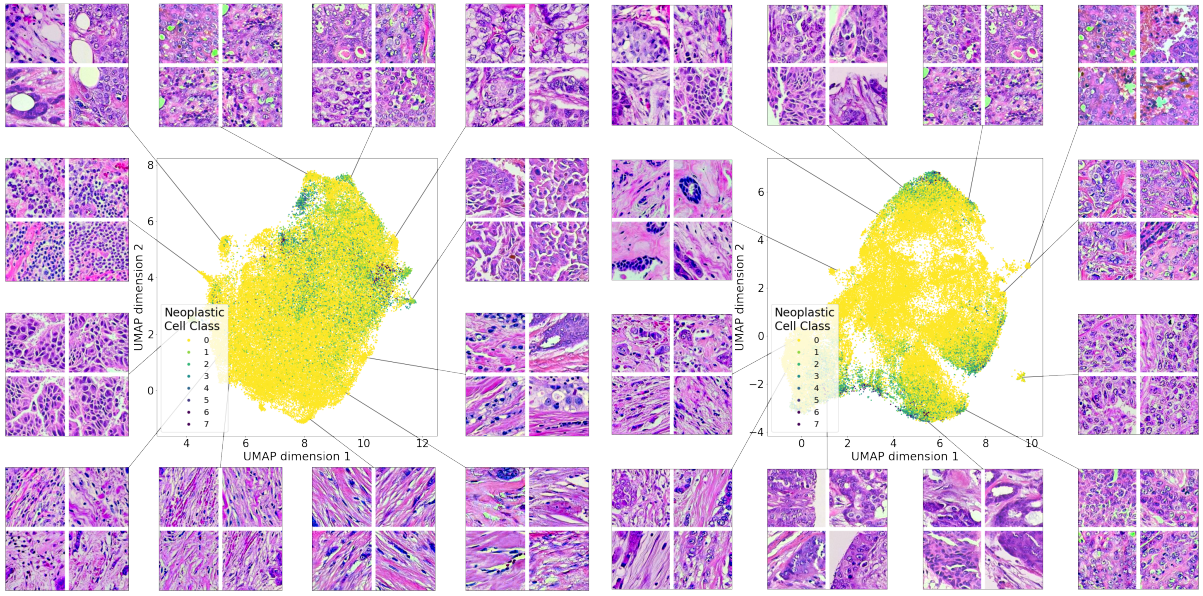


Figure 4.2: Uniform Manifold Approximation and Projection (UMAP) vectors of PathologyGAN's and Inception-V1 breast tissue representations with neoplastic cell annotations. PathologyGAN's representations show more localized neoplastic epithelial cell density. Both representations show how distinct regions of the latent space correspond to different densities to neoplastic cells where class 0 corresponds to lowest density and class 7 to highest density.

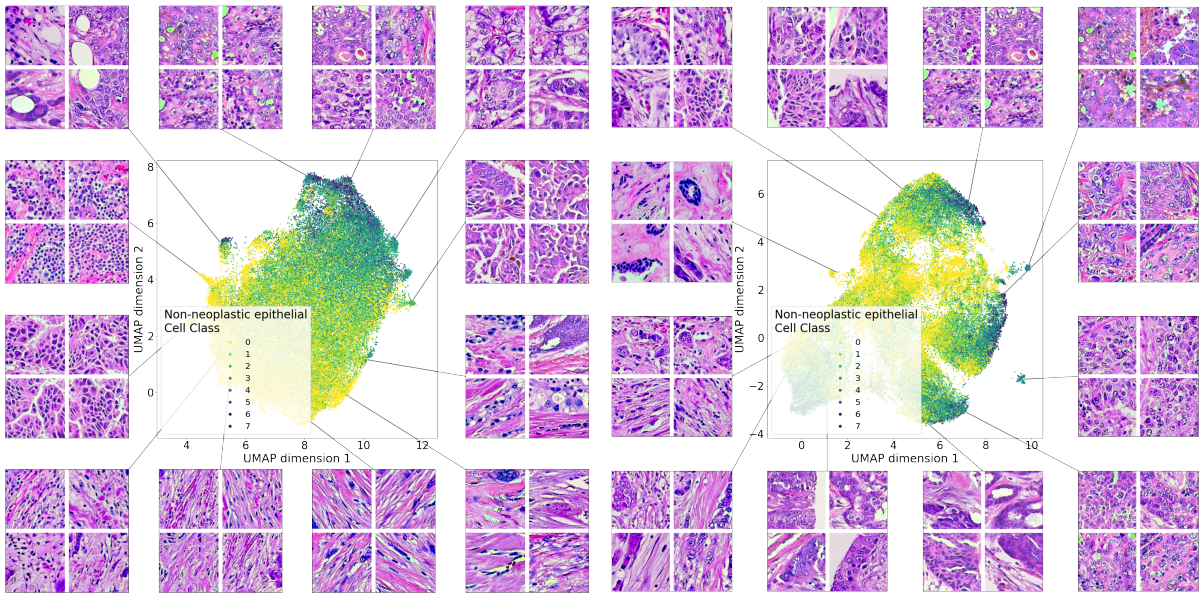


Figure 4.3: Uniform Manifold Approximation and Projection (UMAP) vectors of PathologyGAN's and Inception-V1 breast tissue representations with non-neoplastic cell annotations. PathologyGAN's representations show more localized non-neoplastic epithelial cell density. Both representations show how distinct regions of the latent space correspond to different densities to non-neoplastic cells where class 0 corresponds to lowest density and class 7 to highest density.

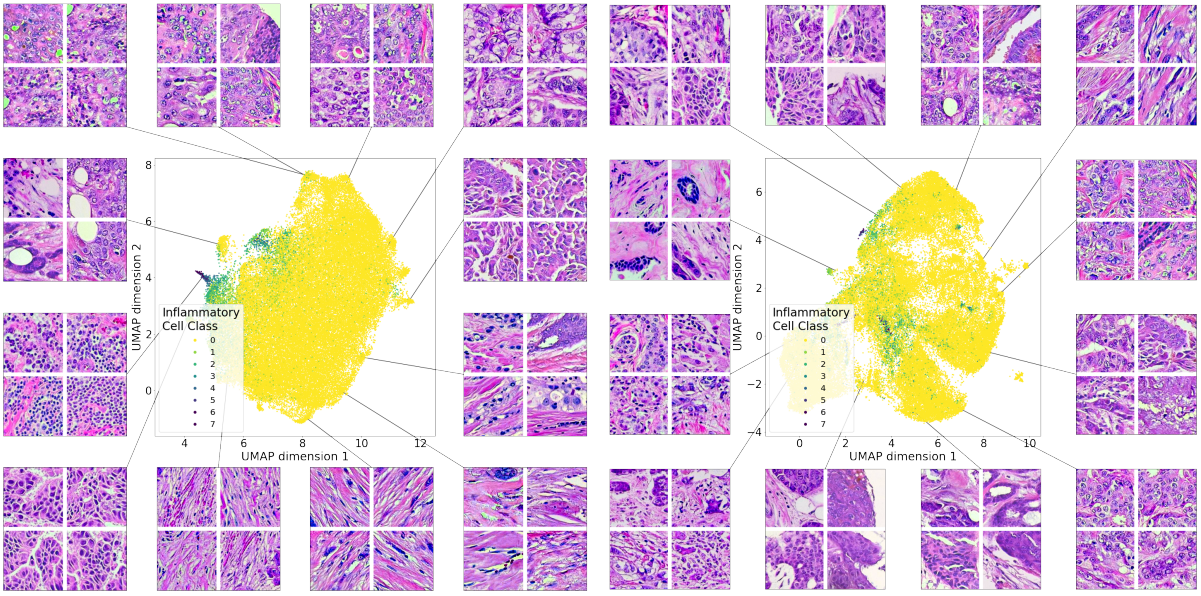


Figure 4.4: Uniform Manifold Approximation and Projection (UMAP) vectors of PathologyGAN's and Inception-V1 breast tissue representations with inflammatory cell annotations. Inception-V1's representations show better isolation of inflammatory cell density while PathologyGAN's are more sparse. Both representations show how distinct regions of the latent space correspond to different densities to inflammatory cells where class 0 corresponds to lowest density and class 7 to highest density.

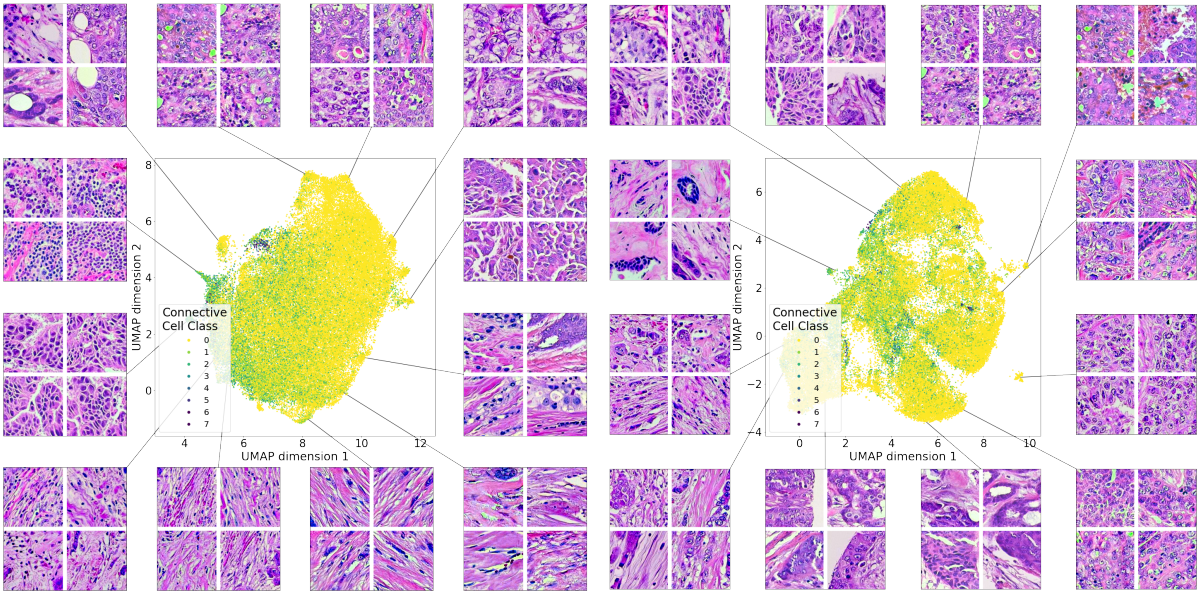


Figure 4.5: Uniform Manifold Approximation and Projection (UMAP) vectors of PathologyGAN's and Inception-V1 breast tissue representations with connective cell annotations. Inception-V1's representations show better isolation of connective cell density while PathologyGAN's are more sparse. Both representations show how distinct regions of the latent space correspond to different densities to connective cells where class 0 corresponds to lowest density and class 7 to highest density.

representations of a pre-tranied ImageNet Inception-V1[148], reaching an AUC of 0.997 and 95% accuracy. Finally, Kather *et al.*[64] provides another performance reference using an RNF-SVM with an AUC of 0.995 and 92.44% accuracy. We provide further details on the Receiver Operating Characteristic (ROC) curve and confusion matrix of the logistic regression performance in **Appendix B.2**.

Tissue representations from our model provide robust information such that a linear classifier can predict tissue types with high accuracy. Despite a small loss of performance when compared to the best-supervised method, our fully unsupervised representations could be an effective so-

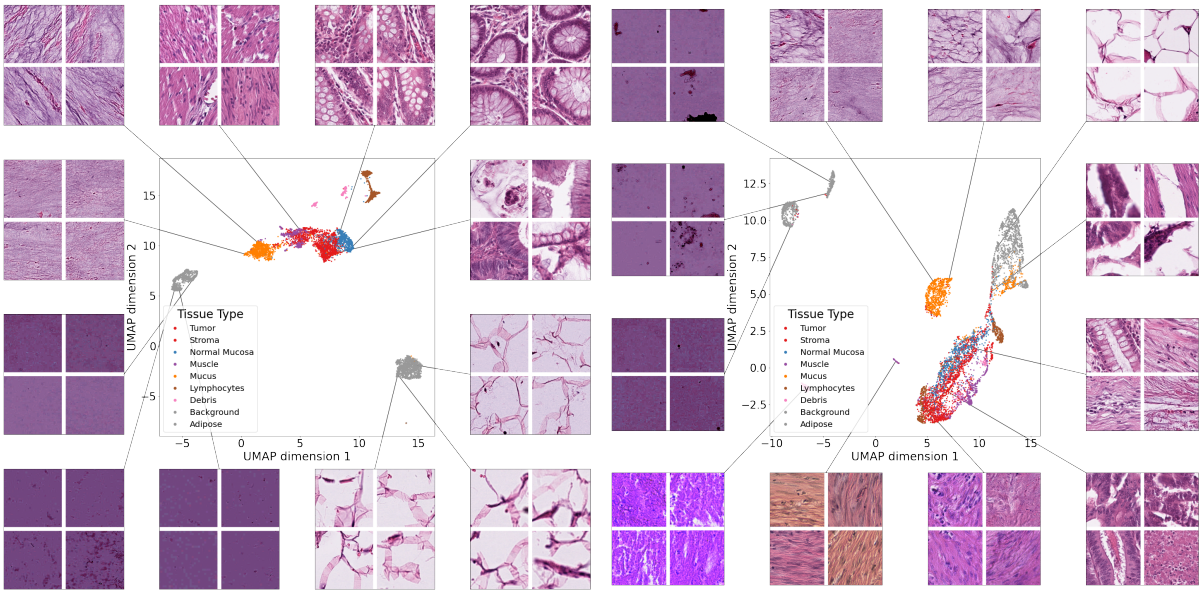


Figure 4.6: Uniform Manifold Approximation and Projection (UMAP) vectors of PathologyGAN's and Inception-V1's colorectal tissue representations with tissue type annotations. Inception-V1's representations show a clearer distinction between tissue types compared to PathologyGAN's representations. In both cases, we observe that real tissue projected to the latent space retain tissue morphology information.

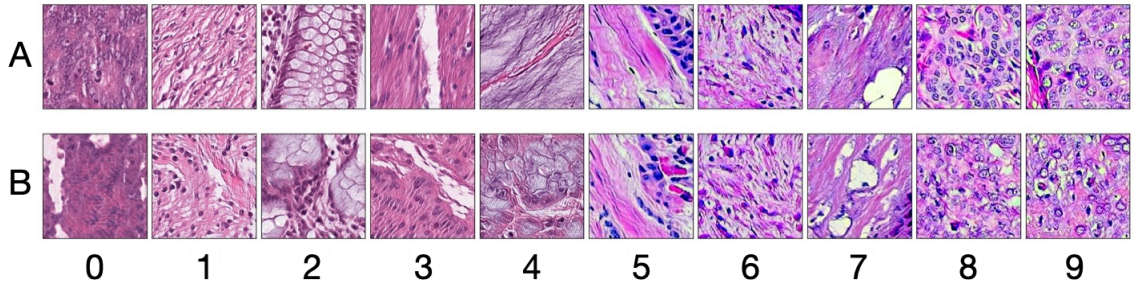


Figure 4.7: Real tissue images and their reconstructions. We take real tissue images and map them to the latent space with our encoder, then we use the generator with the tissue vector representations to generate the image reconstructions. (A) correspond to the real tissue images x_{real} and (B) to the reconstructions $x_{recon} = G(E(x_{real}))$, the images are paired in columns. We present samples of colorectal cancer tissue from NCT (0-4) and breast cancer tissue from VGH and NKI (5-10). We show different examples of tumor(0,8,9), stromal (1,7), normal mucosa (2), muscle (3), and lymphocytes (5,6), the reconstructions follow the real image attributes.

lution when extensive and precise manual annotations cannot be obtained.

4.3.3 Multiple Instance Learning on tissue representations

Finally, we test the reliability of the tissue representations in a weakly supervised setting. We use tissue representations in a multiple instance learning (MIL) task for tumor presence prediction in Whole Slide Images (WSIs), where each WSI has an average, minimum, and maximum of 974, 175, and 14K, tiles, respectively.

We use the lung cancer dataset where each WSI has an associated label tumor or normal subject to the presence of tumor tissue. We divide each WSI into 224×224 patches to train our GAN model and obtain tissue representations, and we later used all tissue representations of the WSI.

In the case of the MIL problem, we have a bag of instances $X = \{x_1, ..., x_n\}$ with an individual label $Y \in \{0, 1\}$. We further assume that each instance of the bag has an associated label $y_i \in \{0, 1\}$ to which we have no access:

$$Y = \begin{cases} 0, & \text{iff } \sum_k y_k = 0 \\ 1, & \text{otherwise} \end{cases}$$

In our case, we have tissue tiles that we translated into tissue representations $W = \{w_1, ..., w_n\}$, and use them to determine the presence of lung tumor $Y = 1$ in the WSI. We used the attention-

Total AUC: 0.976							
Tumor	**Simple Stroma**	**Mucosa**	**Mucus**	**Lymph.**	**Debris**	**Back.**	**Adipose**
0.974	0.929	0.964	0.997	0.994	0.959	1.0	0.998
Total Accuracy: 85.43%							
Tumor	**Simple Stroma**	**Mucosa**	**Mucus**	**Lymph.**	**Debris**	**Back.**	**Adipose**
89%	71%	68%	91%	83%	63%	100%	96%

Table 4.1: Logistic regression Accuracy and AUC on tissue type classification. We use the colorectal cancer tissue images from NCT to train a classifier over tissue representations. A logistic regression (one-vs-rest) is able to reliably find information in the representations to predict tissue types.

Model	**AUC**	**Accuracy**
Ours	0.976	85.43%
Inception-V1[148]	0.997	95.0%
Bayesian DNN[63]	**0.995**	**99.2%**
RBF-SVM[64]	0.976	87.4%

Table 4.2: Performance comparison on tissue type classification between existing supervised methods and tissue representations. These results reflect the baseline performance of supervised deep learning and non-deep learning approaches. Tissue representations on logistic regression without any transformation or projection, and the fact that they are comparable with supervised performance, demonstrate the applicability and information held in them.

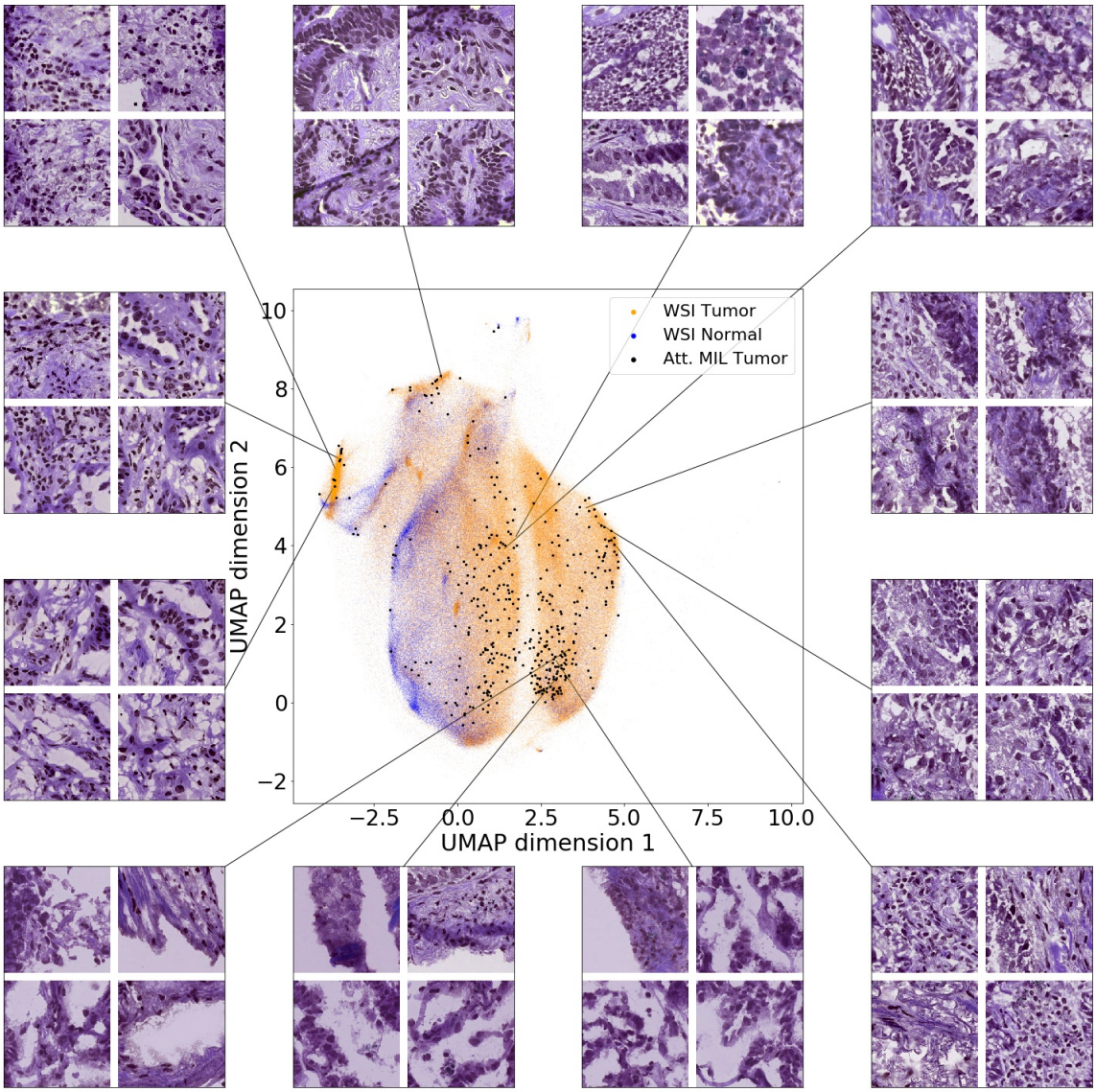


Figure 4.8: Uniform Manifold Approximation and Projection (UMAP) vectors of lung cancer tissue representations. We labeled each patch of the WSI with the corresponding label subject to presence of tumor in the WSI, and highlight representations (black markers) and images where the attention-based deep MIL focuses to predict the outcome. We can see that the MIL framework emphasizes on regions where there is only the presence of tumor patches, relying on the information contained in the tissue representations.

based deep MIL[149] as it assigns a weight to each instance of the bag x_k, allowing us to measure which representations are relevant for the prediction.

The attention-based deep MIL over tissue representations achieves an AUC of 0.98 and accuracy of 94%. These results are comparable to Coudray *et al.*[13], where an Inception-V3 network was tested on the same dataset and purpose achieving an AUC of 0.993 and accuracy of 97.5%.

Figure 4.8 shows a UMAP dimensionality reduction of tissue representations for the test set, we label each representation with the WSI label, tumor or normal depending on the presence of tumor. In addition, we highlight images and representations of the top 0.1% most weighted

representations for the tumor outcome prediction. We can see that the MIL model focuses on regions of the latent space where there is no presence of normal tissue.

Given the accurate prediction of lung tumor presence in the WSIs and the focus of the MIL framework on representations solely present in tumor tissue, these results show that the representations are reliable enough to capture tissue characteristics such as tumor tissue at scale.

4.4 Conclusion

We have presented a new approach for PathologyGAN, a modification that aims to overcome the model's previous shortcomings. PathologyGAN's initial inability to find tissue representations of real images limited its impact on application-related tasks such as tumor detection or tissue classification.

We introduced the proposed solution, including an encoder that effectively learns to inverse the generated images into the GAN's latent space. Once trained, this encoder enables a direct mapping between real images and tissue representations. We further illustrated the applicability of the tissue representations in three different scenarios, providing evidence of the morphological and cellular information captured by the representations.

Through **Figures 4.2-4.6**, we display how PathologyGAN's representations of real images and compared them to pre-trained ImageNet models which have already been shown to capture tissue characteristics in the representations. UMAP dimensionality reductions of the PathologyGAN's representations show a clear relationship with morphological and cellular annotations, exhibiting a structure based on tissue type or cell type density. Interestingly, PathologyGAN's representations show a more localized neoplastic and non-neoplastic epithelial cell density while Inception-V1's display more isolated inflammatory and connective cell density, and a clearer distinction between tissue types.

In addition, **Figure 4.7** shows how tissue representations encode enough information on its morphology and cellularity so the GAN can reconstruct tissue with similar features to the original image. We conclude that tissue representations are not only interpretable but also hold the relevant information needed to synthesize tissue with the same characteristics.

In **Table 4.1** and **Table 4.2**, we also provide evidence of how representations can provide reliable tissue type information to a linear classifier, achieving comparable performance to other fully supervised approaches. PathologyGAN's tissue representations are fully unsupervised, providing an alternative when manual annotations are scarce.

We show how tissue representations can scale to the whole slide image (WSI) level, using them on a multiple instance learning (MIL) task for tumor presence prediction on WSIs. **Figure 4.8** shows how the attention-based deep MIL[149] model focuses on representations that contain tumor tissue, verifying that tissue representations can be used at scale and in real task settings.

As a limitation, PathologyGAN's tissue representations might be susceptible to batch effect

across institutions. PathologyGAN uses Adaptative Instance Normalization (AdaIN) as a building block to generate samples based on each instance representation. AdaIN was first proposed in *neural style transfer* to transform an image based on the content style of another. *i.e.* manipulate color, contrast, or texture from an image based on another. The inherent architecture of AdaIN makes PathologyGAN potentially susceptible to the imaging variations across institutions, also referred to as batch effect[150]. This phenomenon introduces slight variations in resolution, color, contrast, and brightness in WSIs and TMAs due to the imaging process of different institutions. As a result, tissue representations could introduce non-relevant tissue information over tissue feature characteristics, causing the representations not to generalize on a given task across institutions. This possible limitation warrants further study. Although the multiple instance learning (MIL) task for tumor presence prediction was done over several TCGA institutions and it reports on par results with fully supervised approaches, the batch effect could play a bigger role on more complex prediction tasks such as survival.

Overall, we conclude that PathologyGAN's tissue representations show to capture cellular and morphological tissue characteristics from real images. These tissue representations also show to be applicable at scale, providing an alternative when manual annotations are scarce. In addition, it shows an advantage compared to fully supervised and pre-trained models, its ability to generate tissue with the cellular and morphological characteristics encoded in the representations. Nonetheless, the generative process can be an additional burden to representation learning. Alternative approaches such as self-supervised learning can provide methods for learning tissue representations solely from features contained in the image and without the need of labels.

Chapter 5

Histomorphological phenotype learning: a self-supervised methodology for discovery of morphological tissue patterns

5.1 Introduction

In this final chapter, we move away from deep generative models and focus on self-supervised learning. Unlike generative models, these approaches lack a generative process; which can be computationally expensive and may not be a requisite for representation learning. Instead, self-supervised learning focuses on creating useful representations of data based on its features; without relying on labels or manual annotations.

We use self-supervised learning to learn representations of tissue and cluster them based on the tissue patterns found in the image. We refer to these tissue patterns with common characteristics as histomorphological phenotypes. In digital pathology, histomorphological phenotypes are defined as phenotype manifestations of tissue patterns observed by pathologists. They can be characterized by cellular content and behavior as well as spatial patterns found in the tissue image[151], and they provide valuable information about the tumor diagnosis and possible prognosis[5, 13, 26, 152–154].

We introduce Histomorphological Phenotype Learning (HPL), a methodology to automatically extract histomorphological phenotype representations through self-supervised learning and community detection. We present an approach that provides a simple mechanism to relate Histomorphological Phenotype Clusters (HPCs) with different clinical annotations, without the need to retrain the model as would be required by supervised and weakly-supervised end-to-end solutions. In addition, the proposed methodology is easily interpretable, allowing pathologists to scrutinize tissue patterns and their relations to annotations such as cancer type, overall survival, recurrence free survival, or molecular phenotypes.

As proof of the utility, we used our method to discriminate lung cancer types between ade-

nocarcinoma (LUAD) and squamous cell carcinoma (LUSC), and survival analysis for lung adenocarcinoma using WSIs from The Cancer Genome Atlas (TCGA) and the New York University (NYU) School of Medicine as an independent cohort. We provide evidence of how our methodology finds phenotypes that can (A) discriminate between LUAD and LUSC, (B) be prognostic of overall and recurrence free survival, and (C) show correlations between cell types, growth patterns, and omic-based immune signatures.

5.2 Methods

5.2.1 HPL: Histomorphological phenotype learning

HPL discovers histomorphological phenotypes by automatically identifying distinct morphological tissue patterns without using any label or annotation. This approach permits to describe a whole slide image (WSI) or patient as a compositional vector representation where each dimension is the histomorphological phenotype percentage of the total tissue area. WSI and patient compositional vector representations can be used across different kinds of clinical annotations, avoiding the need of model retraining unlike supervised end-to-end solutions. **Figure 5.1** outlines the entire HPL methodology.

The framework starts by segmenting WSIs into 224×224 tiles without overlap. Tiles are filtered out if the tissue in the image does not cover at least 60% of the area. Finally, stain normalization is applied[65] (**Figure 5.1A**).

HPL uses self-supervised learning to capture morphological characteristics found in tissue into vector representations. Through this process, each 224×224 tile image is transformed into a tile vector representation $\{z \in R^D; D = 128\}$. In addition, it aims to create representations that are invariant to color and slight zoom distortions, further mitigating the impact of variations in the imaging processing across institutions[150] (**Figure 5.1B**). HPL uses *Barlow Twins*[123] for self-supervised learning, this method has proven to be on par or outperforming other self-supervised methods while releasing the user from higher computational requirements such as large batches[118] or architecture asymmetries[121,133]; providing competitive state-of-the-art results with less resources.

HPL defines a nearest neighbors graph between z tile vector representations, motivated by the idea that neighboring tile vector representations hold similar morphological features. Next, HPL uses Leiden community detection[155] in the nearest neighbors graph in order to find Histomorphological Phenotype Clusters (HPCs). HPCs are clusters of 224×224 tissue tiles that have common morphological patterns (**Figure 5.1C**).

Defining HPCs allows us to easily describe a WSI or patient (one or more WSIs) by a single vector. Each WSI or patient compositional vector representation (w) has a dimensionality equal to the total number of HPCs (C) and each dimension (w_i) accounts for the percentage of a HPC

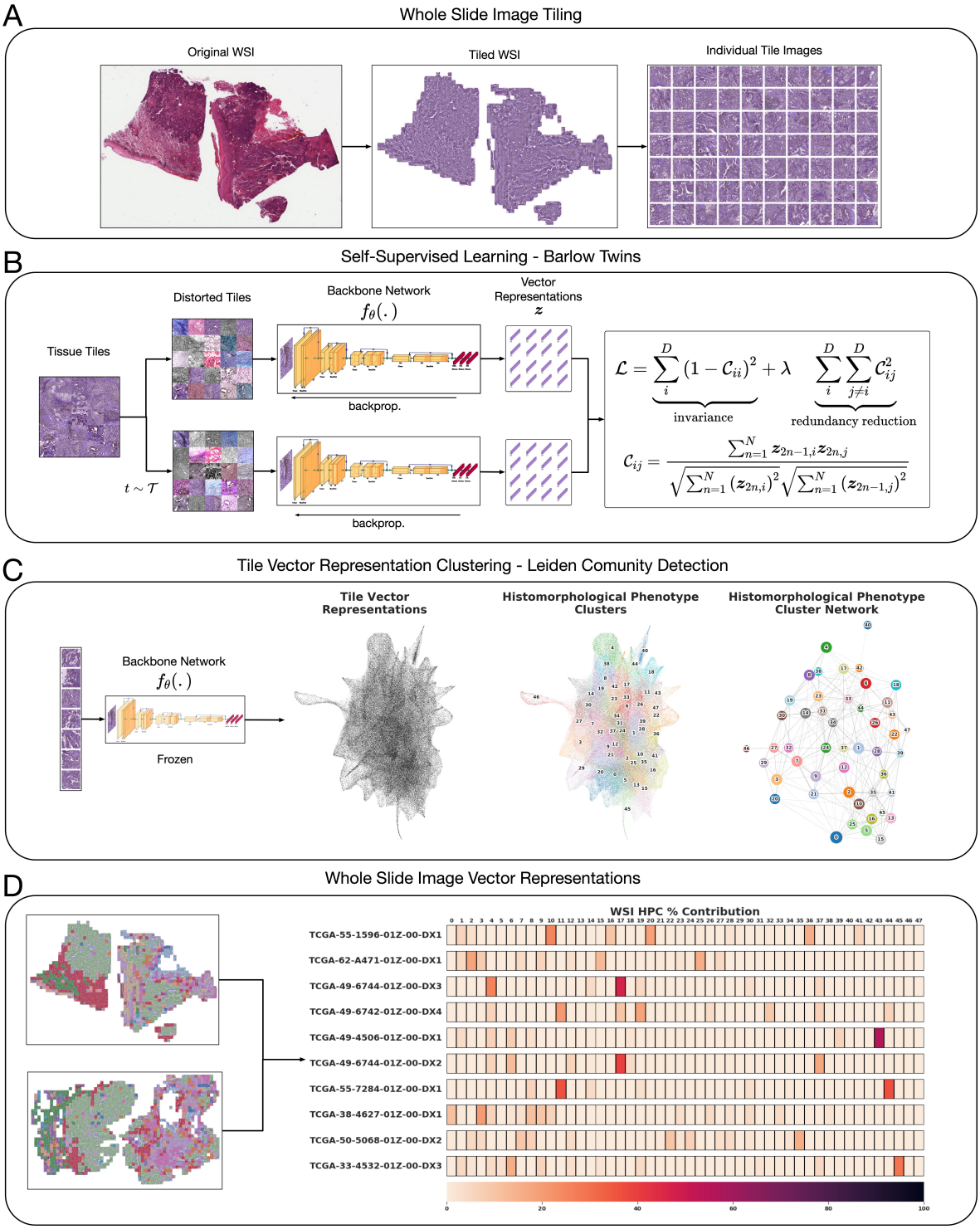


Figure 5.1: Overview of Histomorphological Phenotype Learning (HPL) methodology architecture. **A** Whole slide images (WSIs) are processed for tile extraction and stain color normalization. **B** Self-supervised training of backbone network f_θ in order to create tile vector representations. **C** Tiles are projected into z vector representations using the frozen backbone network f_θ. Continuously, Histomorphological Phenotype Clusters (HPCs) are defined by using Leiden community detection over a nearest neighbor graph of z tile vector representations. **D** WSIs or patients (one or more WSIs per patient) are defined by a compositional vector with dimensionality equal to the number of HPCs, accounts for the percentage of a HPC with respect to the total tissue area. **HPL** creates WSI and patient compositional vector representations that can be easily used in interpretable models such as logistic regression or cox regression, relating tissue phenotypes with clinical annotations.

with respect to the total tissue area (**5.1 - Figure 5.1D**):

$$w = \{w_0, w_1, ..., w_{C-1}\} \text{ where } \{w_i \in [0,1] / \sum_{i=0}^{C-1} w_i = 1\} \tag{5.1}$$

HPL's approach to describing WSIs and patients provides (A) interpretability through tissue patterns that can be easily visualized and analyzed in the context of WSI and by individual tiles, (B) the application of interpretable models such as logistic regression or Cox regression, verifying the statistical significance of phenotypes in cancer type classification or survival, (C) broader association studies with genomic, transcriptomic and other omics profiles, and (D) a description of spatial heterogeneity as well as potential relationship between patterns.

5.2.2 Whole slide image processing

We make use of the publicly available pipeline DeepPATH[13] to divide each WSI into non-overlapping tiles of 224×224 at 5X magnification (with tiles re-sampled to ensure $2.016 \mu m$ per pixel), filter out images with less than 60% tissue in total area, and apply stain normalization[65].

5.2.3 Self-supervised learning

Self-Supervised learning aims to create useful representations of data based on its features which, unlike supervised methods, does so without relying on expensive human annotations. These methods use a pretext task as a mean to capture relevant feature information into each instance representation. Subsequently, these representations can be used in downstream tasks.

A common setup in self-supervised learning is the use of a data augmentation pipeline T in order to create invariant representations against image distortions[127] and a backbone network f_θ to project the images into a latent space $\boldsymbol{z}$, where the different approaches define the pretext task goal.

Barlow Twins[123] aims to create representations that are invariant to distortions while minimizing the redundant information between feature dimensions in the representation. It uses two different transformations to distort each image in the batch $\{\tilde{x}_i = t(x_i),\ \tilde{x}'_i = t'(x_i); t \sim T, t' \sim T \text{ and } [x_n]_{n=1}^N\}$ and runs the images through the backbone network f_θ to obtain latent representations $z \in R^D / D = 128;\ \{z_1, z_2, ..., z_{2N-1}, z_{2N}\} = \{f_\theta(\tilde{x}_1), f_\theta(\tilde{x}'_1), ..., f_\theta(\tilde{x}_N), f_\theta(\tilde{x}'_N)\}$. Right after, the representations are batch normalized for each transformation set $Z \in R^{NxD}; \{Z_{2n}, Z_{2n-1}\}$ and calculates $C \in [-1,1]^{DxD}$ as the cross-correlation matrix computed between $\{Z_{2n}, Z_{2n-1}\}$ along the batch dimension.

The loss is defined by two terms, the invariance term that attempts to create representations invariant to distortions and the redundancy reduction term where off-diagonal features of the correlation matrix are minimized to prevent any redundant information between them (**5.2**). The

parameter λ is used as a weighting factor to balance the difference in number between the diagonal and off-diagonal terms:

$$\mathcal{L} = \underbrace{\sum_{i}^{D} \left(1 - C_{ii}\right)^2}_{\text{invariance}} + \lambda \underbrace{\sum_{i}^{D} \sum_{j \neq i}^{D} {C_{ij}}^2}_{\text{redundancy reduction}} \; ; \; C_{ij} = \frac{\sum_{n=1}^{N} \boldsymbol{z}_{2n-1,i} \boldsymbol{z}_{2n,j}}{\sqrt{\sum_{n=1}^{N} \left(\boldsymbol{z}_{2n,i}\right)^2} \sqrt{\sum_{n=1}^{N} \left(\boldsymbol{z}_{2n-1,j}\right)^2}} \tag{5.2}$$

It is relevant to mention how *Barlow Twins* avoids trivial solutions to the pretext task, also known as collapse in self-supervised learning, where representations for different images hold constant values. This method avoids collapse via redundancy reduction, forcing decorrelation between features of the latent representations, and by batch normalizing the batch vector representations before computing the cross-correlation matrix. *Barlow Twins* is robust against batch size and does not require asymmetries or any kind of memory storage[118,121,133]. Its main advantage is that it can provide on par results with other self-supervised learning methods by constructing a simple and intuitive learning objective while releasing the user from high computational requirements.

The backbone network f_θ is a convolutional neural network (CNN) used as a feature extractor of the tissue images. In our work, we used a CNN composed of several ResNet layers[143] and a self-attention layer[87,156]. This architecture is similar to the one used in Quiros *et al.*[157], where it was shown to be an effective feature extractor.

We trained the self-supervised model using 250K tile images for 60 epochs and batch size of 64. We used a single GPU (NVIDIA Titan Xp 12GB/NVIDIA RTX 24GB) with training times ranging between 48 to 72 hours. Our approach allows to train the self-supervised model on a reasonable amount of time without the need of a large GPU cluster.

Once the backbone network f_θ was trained, it was frozen and only used to project tissue tiles into representations.

5.2.4 Clustering tile vector representations

Leiden community detection[155] has been shown to be a successful algorithm on discovering well connected communities in graph structures. Derived as an improvement from Louvain algorithm[158], it uses a heuristic method to find partitions in graphs.

The algorithm starts by assigning each node in the graph to a different community, then it iterates through the steps described below until there are no further changes in the network. Firstly, for each node the algorithm will evaluate the gain of modularity when the node is moved from the current community to a neighboring one; if there's a positive gain, the node is kept in the new community. After an initial pass, this process is repeated for nodes that changed community until it reaches a local maxima, finding a partition of the graph. Secondly, the previous partition

is refined by possibly further splitting some of the previously defined communities. This is done by randomly merging a node to a community if it increases modularity, which allows to further explore the partition space and avoid badly connected communities, a shortcoming from Louvain algorithm[155]. Finally, nodes in each community are aggregated and communities are treated as nodes in the next iteration.

Modularity is defined by **5.3**, where e_c is the number of edges in a community c, $\frac{K_c^2}{2m}$ is the expected number of edges, K_c is the sum of the degrees of the nodes in a community c, and m is the total number of edges in the network. γ is the resolution parameter where higher values lead to more communities and lower values to fewer communities

$$\mathcal{H} = \frac{1}{2m} \sum_c \left(e_c - \gamma \frac{K_c^2}{2m} \right) \tag{5.3}$$

In order to find Histomorphological Phenotype Clusters (HPCs) from self-supervised tissue representations, we create a graph by using K nearest neighbors ($K = 250$) over $200,000$ randomly sampled vectors from the training set, and use Leiden community detection to define clusters. Subsequently, we assign clusters to vector representations of additional sets by using again K nearest neighbors between vectors of the training set and each of the vectors of the additional set.

We performed the clustering process an initial time in an effort to automatically identify and remove representations of background and artifact tiles. Once these representations were removed, we run the clustering process an additional time so these non informative tile representations do not interfere in the number of tissue communities found from the Leiden algorithm.

5.2.5 Whole slide image and patient vector representations

We defined whole slide image and patient (one or more WSIs) representations as a vector with dimensionality equal to the number of Leiden clusters C (i.e. Histomorphological Phenotype Clusters (HPCs)), where each dimension describes the percentage contribution of an HPC to the total tissue area (**5.4**):

$$w = \{w_0, w_1, ..., w_{C-1}\} \text{ where } \{w_i \in [0,1] / \sum_{i=0}^{C-1} w_i = 1\} \tag{5.4}$$

In order to use these WSI vector representations in linear models we perform the center log ratio transformation (**5.5**). Models such as logistic regression or Cox proportional-hazards require independence between the covariates. The original definition of the WSI vector representations are an example of compositional data which violates this assumption ($\sum_{i=0}^{C-1} w_i = 1$), the center log ratio transformation breaks the co-dependency between variables facilitating the use of these models[159]. In addition, we use multiplicative replacement[160] to avoid zero values before applying

center log ratio transformation.

$$clr(w) = \{log\frac{w_0}{g(w)}, log\frac{w_1}{g(w)}, ..., log\frac{w_{C-1}}{g(w)}\} \text{ where } g(w) = \left(\prod_{i=0}^{C-1} w_i\right)^{1/C} \tag{5.5}$$

5.2.6 Evaluation

To train the self-supervised model, we randomly selected 678 slides out of all 1021 LUAD and LUSC TCGA WSIs, leaving the NYU cohort unseen. In addition, we subsampled $250K$ of $583K$ tiles from those selected WSIs. Once the self-supervised model was trained, it was frozen and used to project tissue tiles into representations.

For our lung adenocarcinoma and squamous cell carcinoma classification analysis we perform the following procedure. We setup a 5-fold cross validation, and to further prevent site-specific bias[150], we performed it in a way to obtain non-overlapping institutions between TCGA training, validation, and test sets. For each fold, we defined Leiden clusters based on representations of the TCGA training set and assign clusters (i.e. HPCs) to TCGA validation and test representations based on those found in the training set. The cohort from NYU was used as an additional external test set to which those HPCs were assigned as well. Subsequently, we fit a logistic regression using WSI vector representations from the training set and evaluated performance on TCGA validation and test sets, and NYU cohort. The motivation behind this procedure was to confirm that the methodology is robust against different imaging process across institutions, maintaining consistent performance. These results can be found in **Appendix C.1**. Once we established that HPCs on different training sets provides similar performance, we locked down a particular cluster configuration and used it to evaluate the performance of the logistic regression across the 5-fold cross validation. We used this procedure so clusters are common across folds and we can evaluate their statistical significance to predict lung adenocarcinoma or squamous cell carcinoma, using Wald test to establish significance on each logisitic regression coefficient. Finally, we combined coefficients for the logistic regression across folds by averaging them and combined p-values with Fisher's method, using a p value threshold of 0.05.

We follow a similar process for our LUAD overall survival analysis, performing a separate independent clustering for sake of consistency. We setup a 5-fold cross validation using the TCGA dataset, dividing it into a training and test set, and kept NYU as an additional independent test set. In this case we did not consider a TCGA validation set or non-overlapping institutions due to the limited number of samples and in an effort to have balanced folds. For each fold, we defined HPCs based on representations of the TCGA training set and assigned HPCs to representations of TCGA test and NYU based on those found in the training set. Afterwards, we used Cox proportional hazards regression to model overall survival, we fit the mode using WSI vector representations from the training set and evaluated the performance on TCGA test and NYU sets. As in the case of the lung type classification, the motivation for this approach was to show

that the methodology is robust even when clustering is done across different training sets. These results can be found in **Appendix C.1**.

Then, we locked down a particular cluster configuration to find statistically significant HPCs across folds, using Wald test to establish significance on each Cox coefficient. We combined coefficients for the Cox proportional hazards regression across folds by averaging them and combined p-values with Fisher's method, using a p value threshold of 0.05.

On our LUAD recurrence free survival analysis, we used a 5-fold cross validation over the NYU cohort. In this case, we use the same HPCs defined by the TCGA training sets of overall survival.

Finally, we used the same cluster configuration selected in overall survival. This allows us to relate statistically significant HPCs for recurrence free and overall survival. We combined coefficients for the Cox proportional hazards regression across folds by averaging them and combine p-values with Fisher's method, using a p value threshold of 0.05.

High and low risk groups were defined by the median of the hazard predictions of the training set. This median value was later used to divide risk groups for the test and additional independent sets. We used logrank test to measure statistical significance between risk groups and use a p-value threshold of 0.05.

5.2.7 Cluster characterizations

Correlation analysis between HPCs and immune landscape features was done through Spearman's rank correlation, significant threshold is set to be 0.01 of the adjusted p-value. P-values were adjusted for multiple comparisons through the Benjamini/Hochberg method for false discovery rate[161].

To account for enrichment and depletion of cell types in tissue tiles of each HPC, we used Hover-Net[19] to estimate counts of neoplastic, connective, inflammatory, and dead cells in each tile of the lung adenocarcinoma NYU_2 cohort. These annotations allows us to measure the distribution of cell type counts per tile for each HPC and the entire population of tiles. We measured over and under-representation of each cell type in an HPC by measuring the distribution shift between the entire population of tiles and tiles assigned to specific clusters. We used the Two-sample Kolmogorov-Smirnov (K-S) test to account for this distribution shift and used the K-S statistic $D_{n,m}$ to quantify for over-representation and under-representations, assigning $+D_{n,m}$ if there is over-representation and $-D_{n,m}$ if there is under-representation. The Two-sample Kolmogorov-Smirnov test uses the statistic $D_{n,m}$ to quantify the distance between empirical cumulative distributions. The statistic is defined as $D_{n,m} = sup_x|F_{1,n}(x) - F_{2,m}|$ where $F_{1,n}(x)$, $F_{2,m}$ are the empirical cumulative distribution functions of the first and second samples, n is the sample size for the F_1, m is the sample size for the F_2, and x the support for both distributions. Significant threshold is set to be 0.01 of the adjusted p-value and p-values were adjusted for multiple comparisons through the Benjamini/Hochberg method for false discovery rate[161]. An example

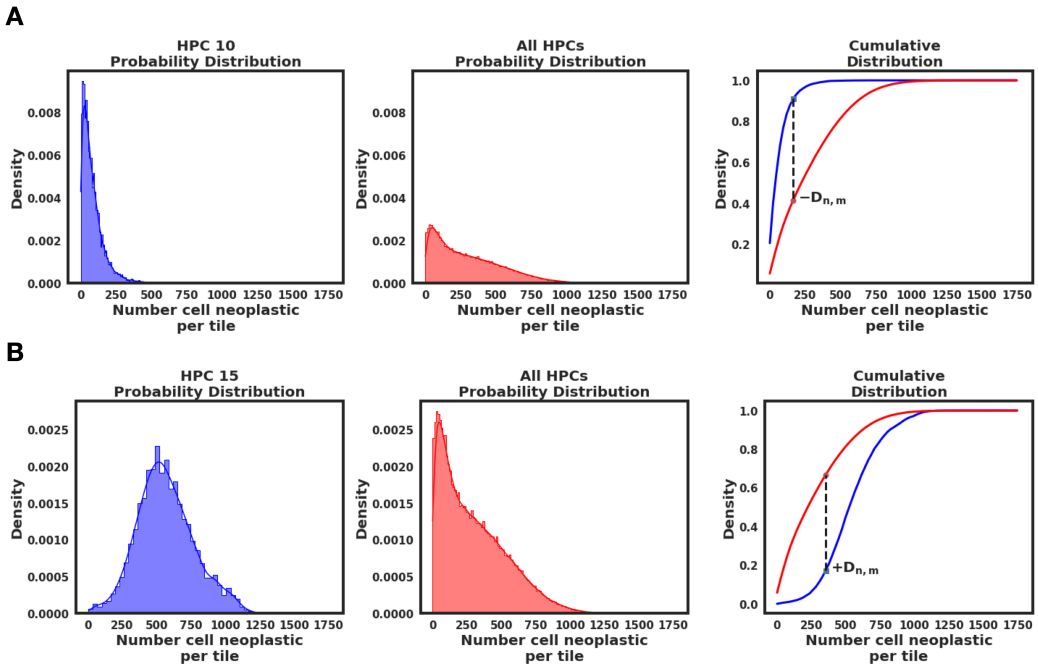


Figure 5.2: Kolmogorov-Smirnov test for lung adenocarcinoma cell type over and under-representation. Representations of usage of Two-sample Kolmogorov-Smirnov (K-S) test to account for over and under-representation of cell types in an HPC. We used the K-S statistic $D_{n,m}$ to quantify for over-representation and under-representations, assigning $+D_{n,m}$ if there is over-representation (**A**) and $-D_{n,m}$ if there is under-representation (**B**). The Two-sample Kolmogorov-Smirnov test uses the statistic $D_{n,m}$ to quantify the distance between empirical distributions. The statistic is defined as $D_{n,m} = sup_x|F_{1,n}(x) - F_{2,m}|$ where $F_{1,n}(x), F_{2,m}$ are the empirical distribution functions of the first and second samples, n is the sample size for the F_1, m is the sample size for the F_2, and x the support for both distributions.

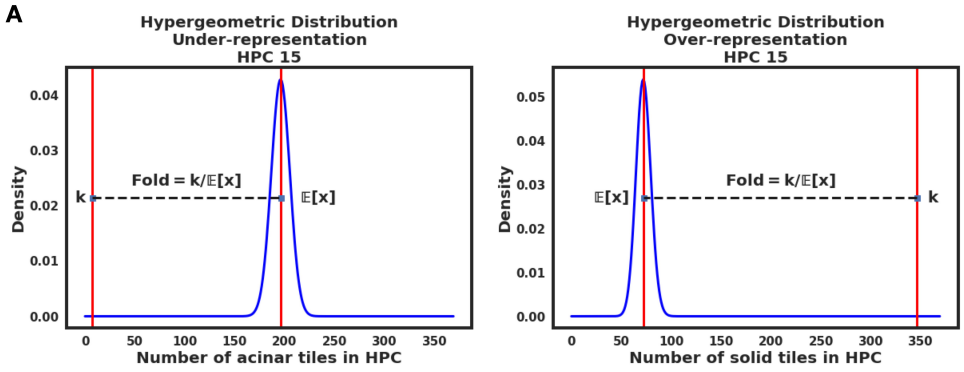


Figure 5.3: Hypergeometric test for lung adenocarcinoma histological subtype enrichment. The hypergeometric test uses the hypergeometric distribution in order to measure the statistical significance of seeing k successes in n draws from a population of size N with K successes, without replacement. In our case, we specify k as the number of tiles with of a particular histological subtype given as the number of total tiles in the HPC n, K is the number of tiles with the same particular histological subtype in the entire population of annotated tiles of size N. We measure enrichment by using a *fold* metric defined as $k/\mathbb{E}[x]$ where $\mathbb{E}[x]$ is the expectation of the hypergeomtric distribution given n, K, N. Enrichment for a particular histological subtype shows as *fold*> 1 and depletion as *fold*< 1. This figure show an example of under-representation of acinar growth pattern and over-representation of solid growth pattern in HPC 15.

for over and under-representation and how it translates to the statistic $D_{n,m}$ can be found in the **Figure 5.2**.

Additionally, we measured enrichment of lung adenocarcinoma subtypes such as solid, acinar, papillary, micropapillary, and lepidic for each HPC. The lung adenocarcinoma NYU_2 cohort contains manual annotations from pathologists on these histological subtypes, and we translated the manual region annotations into individual tile annotations. Subtype enrichment is measured by using the hypergeometric test comparing each HPC to the entire population of annotated tiles.

The hypergeometric test uses the hypergeometric distribution in order to measure the statistical significance of seeing k successes in n draws from a population of size N with K successes, without replacement. In our case, we specify k as the number of tiles with of a particular histological subtype given as the number of total tiles in the HPC n, K is the number of tiles with the same particular histological subtype in the entire population of annotated tiles of size N. We measure enrichment by using a *fold* metric defined as $k/\mathbb{E}[x]$ where $\mathbb{E}[x]$ is the expectation of the hypergeomtric distribution given n, K, N. Enrichment for a particular histological subtype shows as $fold > 1$ and depletion as $fold < 1$, we use a significance threshold of 0.01 for the adjusted p-value. P-values were adjusted for multiple comparisons through the Benjamini/Hochberg method for false discovery rate[161]. Examples of how we used hypergeometric test for enrichment and depletion can be found in the **Figure 5.3**.

5.2.8 Cluster histological assessment

Each HPC defined for tasks (A) lung adenocarcinoma (LUAD) versus cell carcinoma (LUSC) and (B) LUAD overall and recurrence free survival was examined by a subspecialty histopathologist (Prof. John Le Quesne).

HPCs were accounted by examining 100 randomly selected tiles per cluster, each tile had a resolution of 224×224 pixels at 5X magnification, covering approximately $452\mu m^2$ of tissue ($2.016\mu m$ per pixel), since it's known such resolution and field of view are sufficient for lung classification purposes[13]. Each HPC was morphologically described and classified. In addition, HPCs used in LUAD overall and recurrence free survival were described based on lymphocytic infiltration, assigning the value 'hot' to HPCs with most tiles showing moderate to dense stromal lymphocyte infiltration, 'cold' to HPCs with tiles where lymphocyte infiltration is absent or rare, and 'warm' to HPCs with tiles showing a moderate degree of lymphocyte infiltration, a in-between 'hot' and 'cold' conditions. This assessment method was derived from Hendry *et al.*[162] These values were assigned in a manner blinded to HPC outcome associations[152].

5.3 Results

5.3.1 HPL-discovered histomorphological phenotypes can classify lung adenocarcinoma and squamous cell carcinoma

We evaluate HPL's performance at classifying lung adenocarcinoma (LUAD) versus squamous cell carcinoma (LUSC) on The Cancer Genome Atlas (TCGA) and New York University (NYU_1) cohorts. Following HPL methodology, we divide LUAD and LUSC whole slide images (WSIs) into 224×224 tile images (**Figure 5.1A**). Next, we train the self-supervised model on a subset of TCGA's 224×224 tile images (**Figure 5.1B**), followed by a projection of all 224×224 tile images into z tile vector representations. Afterwards, Histomorphological Phenotype Clusters (HPCs) are defined only from a subset of TCGA's z tile vector representations (**Figure 5.1C**). Finally, we define WSI vector representations from each HPC percentage contribution to the total tissue area (**Figure 5.1D**).

For the lung type classification task, we use a logistic regression over WSI vector representations; setting up a 5-fold cross-validation over TCGA's whole slides images (WSIs) and use NYU_1 as an independent test set. The TCGA cohort is composed of 1021 WSIs, 513 of LUAD and 508 of LUSC, while NYU_1 is composed of 138 WSIs, 72 of LUAD and 66 of LUSC. Each TCGA fold is built from a training, validation, and test set with WSIs from different institutions across sets. On average, the training set accounts for 60% of the TCGA slides, validation for 20%, and test set for 20%, while the NYU slides are fully kept as an additional external test cohort. The motivation behind setting a 5-fold cross-validation with non-overlapping institutions is to test HPL reliability against the variability of slide preparation and imaging processing across institutions. HPL achieves an average TCGA test set AUC of 0.930 with 95% confidence intervals (CI) of 0.913-0.949, and an average of 0.990 with 95% CI of 0.988-0.992 on the independent test cohort from NYU. In addition, HPL shows an F1-Score of 0.848 with 95% confidence intervals (CI) of 0.817-0.877 on the TCGA test set, and an average of 0.962 with 95% CI of 0.952-0.970 on the independent test cohort from NYU. These results provide evidence of robustness in prediction of LUAD and LUSC types across different institutions (**Figure 5.4A-B**). On the TCGA dataset, we provide further results on AUC performance per institution and institution contribution (TCGA is a multi-institutional cohort) per HPC on **Appendix C.3**.

One of the main features of HPL's approach is the interpretability behind the outcome prediction. **Figure 5.4B-D** and **Figure 5.5A-E** provide insight into the relationship between histomorphological phenotypes and classification of WSIs into LUAD and LUSC, highlighting also the spatial heterogeneity and spatial relationships between phenotypes.

Figure 5.4C shows a Uniform Manifold Approximation and Projection (UMAP)[145] dimensionality reduction of WSI vector representations for the TCGA and NYU cohorts, showing a division between LUAD and LUSC even in 2 dimensions. The WSI vector representations are defined in complete unsupervised manner, as previously described. **Figure 5.4D** displays a bi-hierarchical clustering based of WSI vector representations. We use a gradient color scheme to label HPCs based on their type purity from 100% LUSC (blue) to 100% LUAD (orange), we

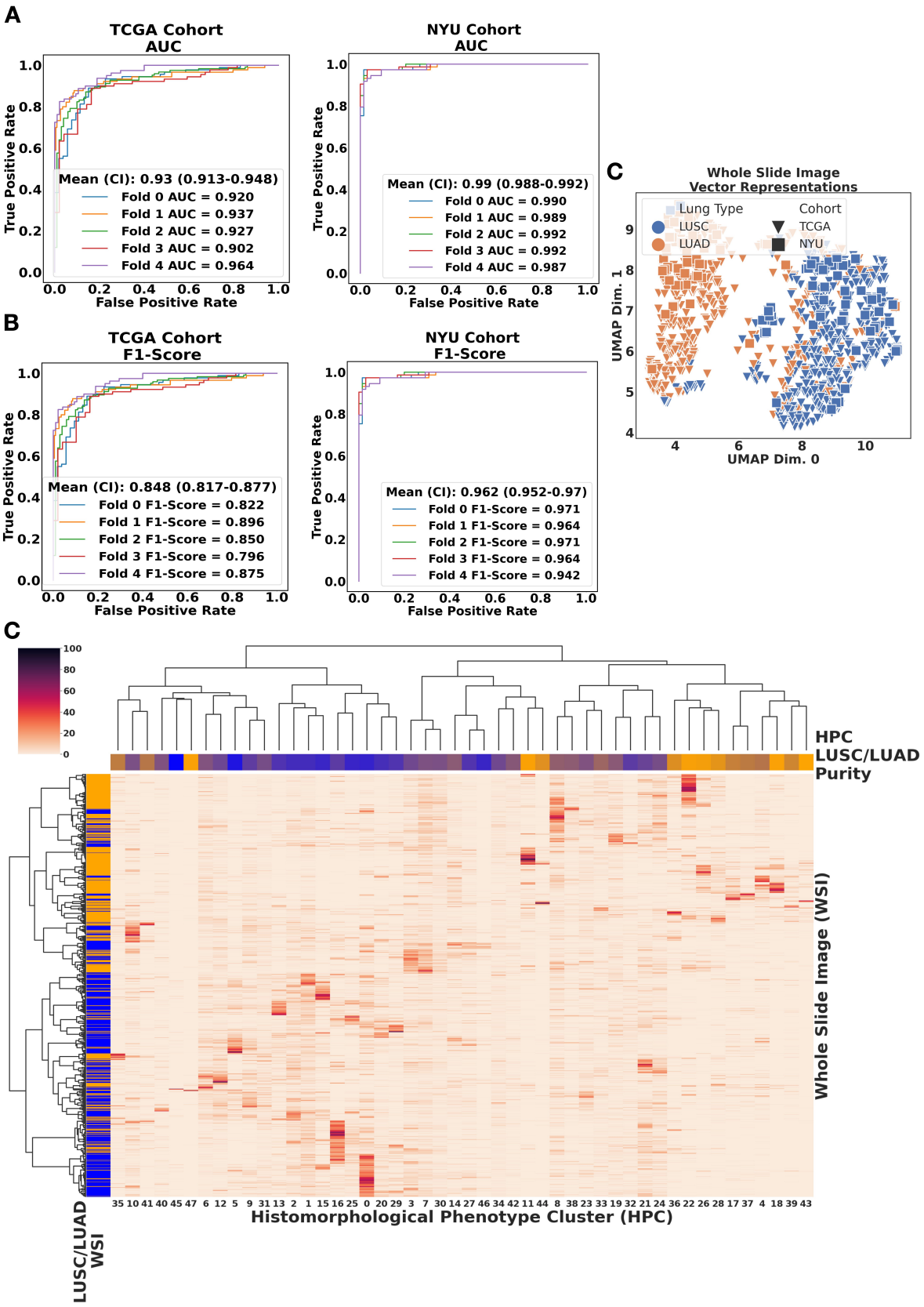

A
TCGA Cohort
AUC
Mean (CI): 0.93 (0.913-0.948)
Fold 0 AUC = 0.920
Fold 1 AUC = 0.937
Fold 2 AUC = 0.927
Fold 3 AUC = 0.902
Fold 4 AUC = 0.964
NYU Cohort
AUC
Mean (CI): 0.99 (0.988-0.992)
Fold 0 AUC = 0.990
Fold 1 AUC = 0.989
Fold 2 AUC = 0.992
Fold 3 AUC = 0.992
Fold 4 AUC = 0.987
True Positive Rate
False Positive Rate
B
TCGA Cohort
F1-Score
Mean (CI): 0.848 (0.817-0.877)
Fold 0 F1-Score = 0.822
Fold 1 F1-Score = 0.896
Fold 2 F1-Score = 0.850
Fold 3 F1-Score = 0.796
Fold 4 F1-Score = 0.875
NYU Cohort
F1-Score
Mean (CI): 0.962 (0.952-0.97)
Fold 0 F1-Score = 0.971
Fold 1 F1-Score = 0.964
Fold 2 F1-Score = 0.971
Fold 3 F1-Score = 0.964
Fold 4 F1-Score = 0.942
C
Whole Slide Image
Vector Representations
Lung Type
LUSC
LUAD
Cohort
TCGA
NYU
UMAP Dim. 1
UMAP Dim. 0
C
HPC
LUSC/LUAD
Purity
Whole Slide Image (WSI)
35 10 41 40 45 47 6 12 5 9 31 13 2 1 15 16 25 0 20 29 3 7 30 14 27 46 34 42 11 44 8 38 23 33 19 32 21 24 36 22 26 28 17 37 4 18 39 43
Histomorphological Phenotype Cluster (HPC)
LUSC/LUAD
WSI

Figure 5.4: Lung adenocarcinoma (LUAD) versus squamous cell carcinoma (LUSC) classification performance and representation analysis. **A** Receiver operating characteristic (ROC) curve over a 5-fold cross-validation on TCGA dataset and NYU dataset as independent cohort, with mean AUCs of 0.93 and 0.99 respectively. **B** F1-Score on TCGA and NYU as independent cohort with mean values of 0.85 and 0.96 respectively. **C** Uniform Manifold Approximation and Projection (UMAP)[145] dimensionality reduction of whole slide image (WSI) compositional vector representations for TCGA and NYU cohorts. Each representation is labeled with the corresponding lung type, LUSC (blue) and LUAD (orange). **C** Bi-hierarchical clustering of WSI vector representations based on HPCs. We use a gradient color scheme to label HPCs, from 100% LUSC (blue) to 100% LUAD (orange). The ground truth label of each WSI is shown on the y-axis using the same color code, LUAD (orange) and LUSC (blue). We calculate purity as the percentage of the dominant lung type tiles with respect to the total cluster tiles.

calculate purity as the percentage of the dominant lung type tiles with respect to the total HPC tiles. The ground truth label of each WSI is shown on the y-axis using the same color code, LUAD (orange) and LUSC (blue). LUAD WSIs show correlation with HPCs of high LUAD purity, concurrently, LUSC slides show correlation with HPCs of high LUSC purity.

Additionally, **Figure 5.5A** shows a UMAP dimensionality reduction[145] of tile vector representations, each tile is labeled LUAD or LUSC based on its associated WSI ground truth label; this figure also shows each tile and its corresponding HPC assignment. Finally, it displays HPCs and their connections through partition-based graph abstraction (PAGA)[163], each node graph is represented with the percentage of lung type tiles in the cluster. We see that distinct sets of HPCs in the graph are attributed to a specific cancer type with high percentage. **Figure 5.5B** displays a Forest plot of the logistic regression coefficients across the 5-fold cross-validation, showing that HPCs $11, 31, 28, 36, 22$ are statistically significant predictors of LUAD while HPCs $5, 45$ predict LUSC. Histomorphological phenotype patterns can be easily visualized by individual tiles in the HPCs (**Figure 5.5C-D**) or in the context of a WSI (**Figure 5.5E-F**), providing a simple way for pathologists to interpret tissue patterns with clinical annotations. Making use of this feature, HPL-discovered HPCs are analyzed by a subspecialty histopathologist (John Le Quesne). The analysis finds that HPCs favoring LUSC are characterized by solid sheets of malignant cells displaying marked intercellular pallor and necrosis (e.g. **Figure 5.5C** HPC 45) or peripheral palisading and eddy formation (e.g. **Figure 5.5C** HPC 5). Clusters favoring LUAD are enriched for a range of known growth patterns: lepidic and papillary pattern tiles with acellular luminal debris (e.g. **Figure 5.5D** HPC 11), acinar and papillary in appearance (e.g. **Figure 5.5D** HPC 28), and cribriform and micropapillary regions with complex small acinar spaces (e.g. **Figure C.5** HPC 36). Impressively, the successful assignment of tumor types is achieved using images at a 5X magnification, without sufficient resolution to identify intercellular bridging, a common identifier of lung squamous cell carcinoma (LUSC). We provide further examples of cluster tiles for TCGA and NYU_1 cohort in **Appendix C.3**.

Collectively, **Figures 5.4** and **5.5** demonstrate that HPCs $31, 11, 22, 36, 28$ (LUAD) and $5, 45$ (LUSC) have a high percentage of each lung type tiles, clear correlations between WSI labels

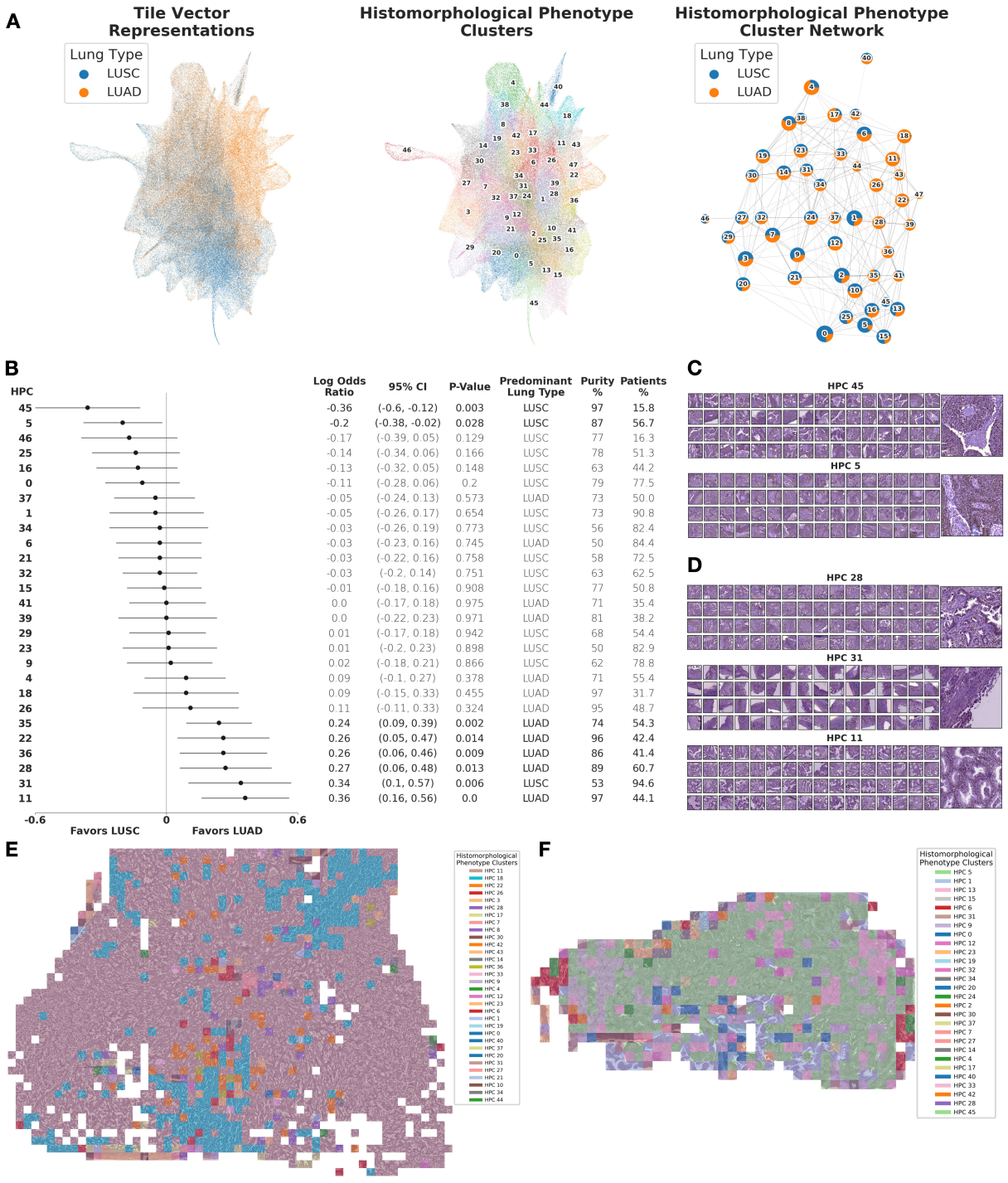


Figure 5.5: Lung adenocarcinoma (LUAD) versus squamous cell carcinoma (LUSC) interpretability behind classification performance. **A** UMAP dimensionality reduction of tile vector representations, each tile label corresponds to the associated WSI lung type (left) and HPC membership (middle), and partition-based graph abstraction (PAGA)[163] of HPC and their connections (right). **B** Forest plot of the logistic regression's coefficients. We highlight statistically significant HPCs (p-value<0.05) and include each HPC's dominant type and purity. We calculate purity as the percentage of the dominant cancer type tiles with respect to the total HPC tiles. **C** Tissue tile samples for HPCs favoring LUAD prediction. **D** Tissue tile samples for HPCs favoring LUSC prediction. **E** LUAD WSI with an overlay of HPCs. The order of clusters in the legend is sorted by percentage of area, from largest to smallest. **F** LUSC WSI with an overlay of HPCs. HPCs 31, 11, 22, 36, 28 (LUAD) and 5, 45 (LUSC) show (1) high percentage of respective lung cancer types, (2) correlation between WSI labels and phenotype purity, and (3) statistical significance when predicting each cancer type.

and type purity, and statistical significance in predicting LUAD and LUSC.

5.3.2 HPL identifies prognostic histomorphological phenotypes in lung adenocarcinoma associated with clinical outcomes

We then use HPL to discover patterns that are related to lung adenocarcinoma (LUAD) clinical outcomes. More specifically, we use HPL only on lung adenocarcinoma samples. We define HPCs by only using z tile vector representations from LUAD WSIs, obtaining a new set of HPCs. Next, we define patient vector representations from each HPC percentage contribution to the total tissue area. Finally we test the HPCs relevance in overall and recurrence free survival by using patient vector representations on Cox proportional hazards models.

For the LUAD overall survival analysis, we use a 5-fold cross-validation over the TCGA data with a new cohort from NYU (NYU_2) as an additional independent set. The TCGA cohort is composed by 443 patients while NYU_2 is composed of 276 patients. We use a Cox proportional hazards model over patient vector representations, describing each patient as a composition of the different HPCs. HPCs are able to achieve a mean concordance index (c-index) of 0.60 with 95% confidence intervals (CI) of 0.56-0.63 on the TCGA test set and a mean c-index of 0.65 with 95% CI of 0.63-0.67 on NYU_2. **Figure 5.6A** shows a UMAP dimensionality reduction[145] of tile vector representations labeled by HPC membership and **Figure 5.6B** displays HPCs and their similarities through partition-based graph abstraction (PAGA)[163]. **Figure 5.6C** shows a Forest plot of the Cox proportional hazards coefficients over the 5-fold cross-validation, providing insight into the relationship between cluster phenotypes and their relevance in predicting overall survival; negative values show association with survival while positive values are associated with a death event. In **Figure 5.6D** we show samples of the top HPCs associated with lower risk and **Figure 5.6E** samples of HPCs associated with higher risk. Finally, we perform histological assessments on HPCs, including a lymphocytic infiltration assessment where clusters are labeled as immune 'hot' if most tiles show moderate to dense stromal lymphocyte infiltration, 'cold' if lymphocyte infiltration is absent or rare in cluster tile images, and 'warm' if a cluster lies between these extremes . We include a detailed description of the HPCs evaluation at **Figure C.8**. We found that HPCs associated with lower risk are typified by low-grade lepidic growth (e.g. **Figure 5.6D** HPC 37) and/or dense lymphocytic infiltrates overrunning and possibly destroying epithelial areas, i.e. they are immune 'hot' (e.g. **Figure 5.6D** HPC 1). High-risk HPCs show poorly differentiated high-grade growth patterns (e.g. **Figure 5.6E** HPCs 15, 39) or necrosis (e.g. **Figure C.6** HPC 14), and are never immune 'hot'. **Figure 5.6F** and **Figure 5.6G** display high and low risk groups for the TCGA and NYU cohorts respectively, both of them statistically significant (p values < 0.05).

We also perform a LUAD recurrence free survival study on systemic and loco-regional recurrence. In this case, we use NYU_2 as it provides detailed information of these types of recurrence

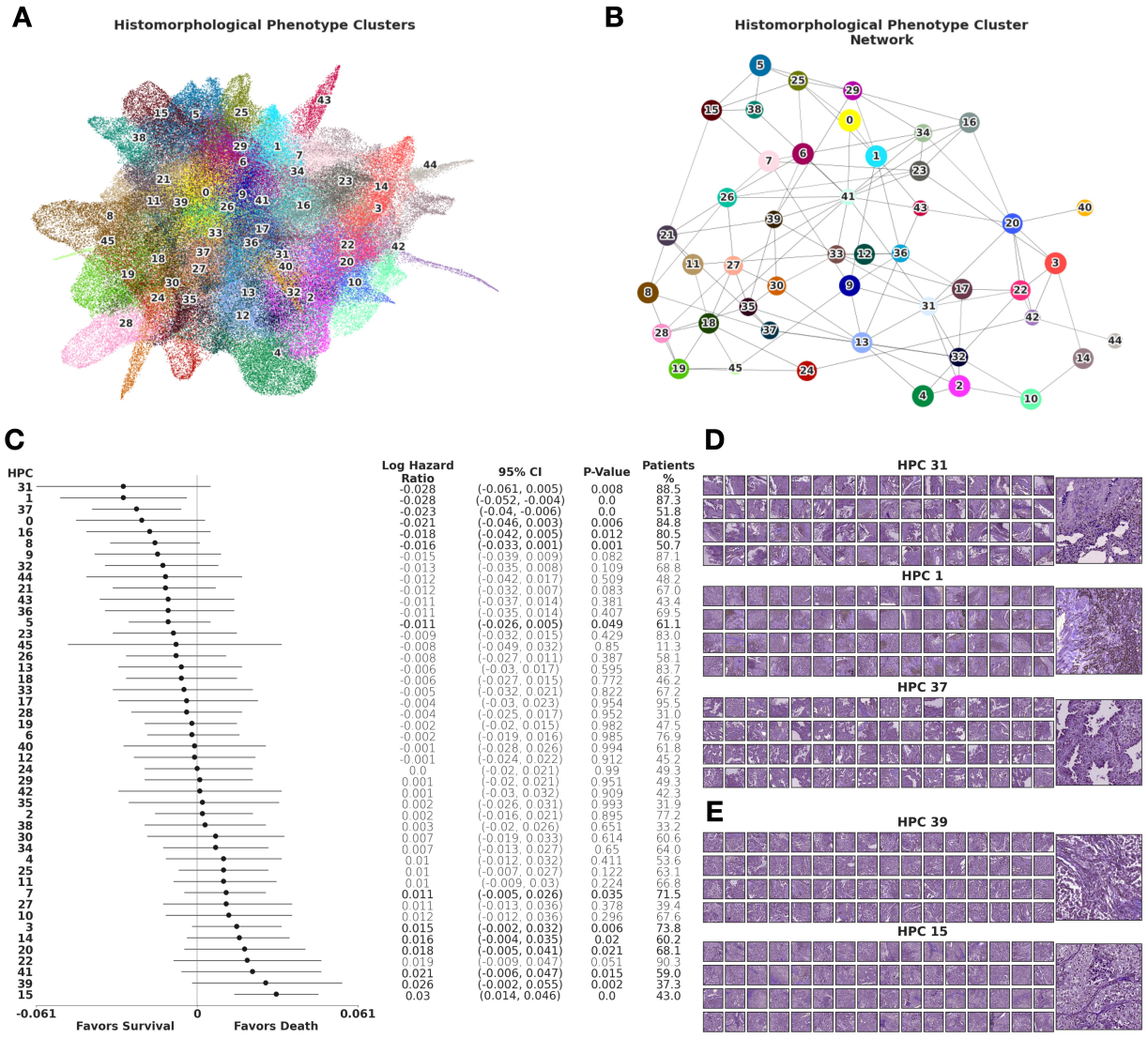

A
Histomorphological Phenotype Clusters
B
Histomorphological Phenotype Cluster Network
C
HPC
Log Hazard Ratio
95% CI
P-Value
Patients %
31 -0.028 (-0.061, 0.005) 0.008 88.5
1 -0.028 (-0.052, -0.004) 0.0 87.3
37 -0.023 (-0.04, -0.006) 0.0 51.8
0 -0.021 (-0.046, 0.003) 0.006 84.8
16 -0.018 (-0.042, 0.005) 0.012 80.5
8 -0.016 (-0.033, 0.001) 0.001 50.7
9 -0.015 (-0.039, 0.009) 0.082 87.1
32 -0.013 (-0.035, 0.008) 0.109 68.8
44 -0.012 (-0.042, 0.017) 0.509 48.2
21 -0.012 (-0.032, 0.007) 0.083 67.0
43 -0.011 (-0.037, 0.014) 0.381 43.4
36 -0.011 (-0.035, 0.014) 0.407 69.5
5 -0.011 (-0.026, 0.005) 0.049 61.1
23 -0.009 (-0.032, 0.015) 0.429 83.0
45 -0.008 (-0.049, 0.032) 0.85 11.3
26 -0.008 (-0.027, 0.011) 0.387 58.1
13 -0.006 (-0.03, 0.017) 0.595 83.7
18 -0.006 (-0.027, 0.015) 0.772 46.2
33 -0.005 (-0.032, 0.021) 0.822 67.2
17 -0.004 (-0.03, 0.023) 0.954 95.5
28 -0.004 (-0.025, 0.017) 0.952 31.0
19 -0.002 (-0.02, 0.015) 0.982 47.5
6 -0.002 (-0.019, 0.016) 0.985 76.9
40 -0.001 (-0.028, 0.026) 0.994 61.8
12 -0.001 (-0.024, 0.022) 0.912 45.2
24 0.0 (-0.02, 0.021) 0.99 49.3
29 0.001 (-0.02, 0.021) 0.951 49.3
42 0.001 (-0.03, 0.032) 0.909 42.3
35 0.002 (-0.026, 0.031) 0.993 31.9
2 0.002 (-0.016, 0.021) 0.895 77.2
38 0.003 (-0.02, 0.026) 0.651 33.2
30 0.007 (-0.019, 0.033) 0.614 60.6
34 0.007 (-0.013, 0.027) 0.65 64.0
4 0.01 (-0.012, 0.032) 0.411 53.6
25 0.01 (-0.007, 0.027) 0.122 63.1
11 0.01 (-0.009, 0.03) 0.224 66.8
7 0.011 (-0.005, 0.026) 0.035 71.5
27 0.011 (-0.013, 0.036) 0.378 39.4
10 0.012 (-0.012, 0.036) 0.296 67.6
3 0.015 (-0.002, 0.032) 0.006 73.8
14 0.016 (-0.004, 0.035) 0.02 60.2
20 0.018 (-0.005, 0.041) 0.021 68.1
22 0.019 (-0.009, 0.047) 0.051 90.3
41 0.021 (-0.006, 0.047) 0.015 59.0
39 0.026 (-0.002, 0.055) 0.002 37.3
15 0.03 (0.014, 0.046) 0.0 43.0
-0.061
0
0.061
Favors Survival
Favors Death
D
HPC 31
HPC 1
HPC 37
E
HPC 39
HPC 15
F
TCGA Cohort
Survival Probability
Time (Years)
Risk Group
Low
High
G
NYU Cohort
Survival Probability
Time (Years)
Risk Group
Low
High

Figure 5.6: Lung adenocarcinoma overall survival. **A** Uniform Manifold Approximation and Projection (UMAP)[145] dimensionality reduction of lung adenocarcinoma tile vector representations labeled by HPC membership. **B** Partition-based graph abstraction (PAGA)[163] of HPCs and their similarities. **C** Log hazard ratio of Cox proportional hazards model over a 5-fold cross-validation over TCGA data and NYU as independent cohort. We average coefficients across fold and combine p-values with Fisher's combined probability test. Statistically significant HPCs (p-value$<$ 0.05) and HPCs that contain at least 10% of total patients are in bold, which is motivated by finding tissue patterns that can generalize across the cohort. This plot shows the relationship between HPCs and their relevance in predicting overall survival, negative values show association with survival while positive values are associated with a death event. **D** Top relevant HPCs associated with survival. **E** Top relevant HPCs associated with a patient's death. These figures correspond to HPL's performance of mean concordance index (c-index) of 0.60 with 95% confidence intervals (CI) of 0.56-0.63 on the TCGA test set and a mean c-index of 0.65 with 95% CI of 0.63-0.67 on the NYU set. HPCs associated with lower risk are typified by low-grade lepidic appearances (e.g. **D** HPC 37) or dense lymphocytic infiltrates overunning and possibly destroying epithelial areas (e.g. **D** HPC 1). High-risk HPCs often show poorly differentiated high-grade growth patterns with minimal inflammation (e.g. **E** HPCs 15, 39). **F** High and low risk groups on TCGA cohort showing statistical significance (p-value $3.7 \times 10^{-3} < 0.05$). **G** High and low risk groups on NYU cohort showing statistical significance (p-value $1.6 \times 10^{-2} < 0.05$).

and has considerably larger follow up times compared to the TCGA dataset, allowing a more refined study of recurrence. Once again we use a 5-fold cross-validation, achieving a mean c-index of 0.74 with 95% CI of 0.70-0.80 on NYU_2. **Figure 5.7A** displays a Forest plot of the Cox proportional hazards coefficients over the 5-fold cross-validation, showing the relationship between HPCs and their relevance in recurrence. In order to find tissue patterns that can generalize across the cohort, we focus on HPCs that contain at least 10% of the total number of patients. In **Figure 5.7B** we show samples of the top HPCs associated with lower risk of recurrence, and in **Figure 5.7C** samples of HPCs associated with higher risk of recurrence. Finally, **Figure 5.7D** displays Kaplan-Meier curves and **Figure 5.7E** UMAP dimensionality reduction on patient representations for high and low risk groups on NYU_2, both figures show a clear split between risk groups. Ultimately, histopathological scrutiny of cluster tile images shows that low risk HPCs are all at least immune 'warm' (e.g. **Figure 5.7B** HPC 29) while high-risk HPCs all display predominantly high-grade growth patterns (e.g. **Figure 5.7C** HPCs 6, 15, 11 and **Figure C.7 HPCs** 24, 5). We provide further tile examples of lung adenocarcinoma HPCs used in the survival analysis in **Appendix C.4**.

5.3.3 A systematic association of HPL-discovered patterns with cell types, histological growth patterns, and molecular phenotypes

Furthermore, we characterize the HPCs in three different ways: (1) By using Spearman's rank correlation between cluster contributions and immune signatures such as tumor infiltrating leukocytes (TIL), proliferation, or wound healing[66], (2) by annotating cells in the tiles from the NYU_2 cohort with Hover-Net[19], and using the two-sample Kolmogorov-Smirnov test to measure over

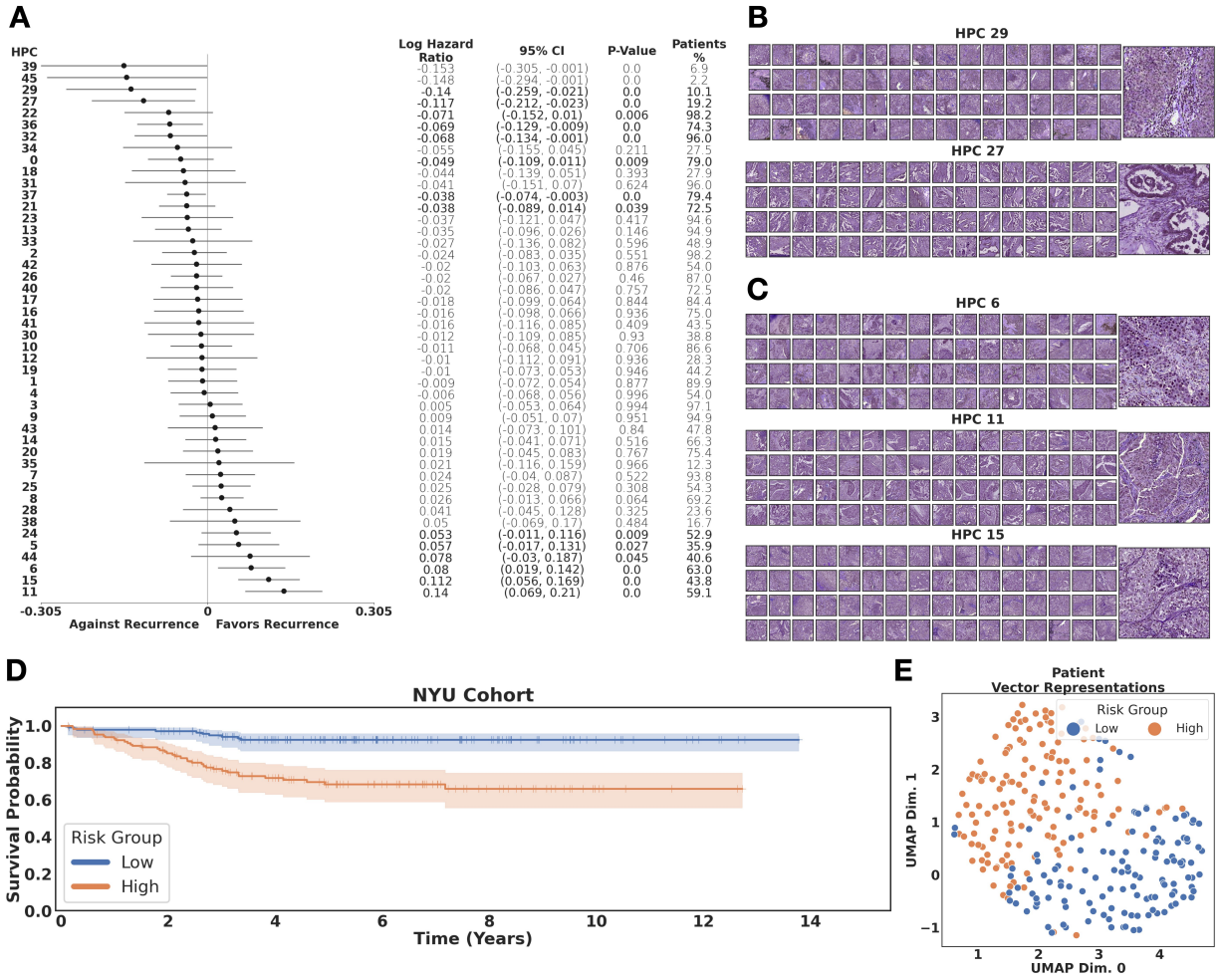


Figure 5.7: Lung adenocarcinoma recurrence free survival. **A** Log hazard ratio of Cox proportional hazards model over a 5-fold cross-validation on the NYU cohort. We average coefficients across fold and combine p-values with Fisher's combined probability test. Each cluster displays the percentage of total patients that show the HPC. This plot shows the relationship between HPCs and their positive or negative relevance in predicting recurrence. In order to find tissue patterns that can generalize across the cohort, we focus on HPCs that contain at least 10% of the total number of patients and show statistical significance (p-value<0.05); these are displayed in bold. **B** Top relevant HPCs associated with lower risk recurrence. **C** Top relevant HPCs associated with high risk of recurrence. **D** High and low risk groups on NYU cohort showing statistical significance (p-value $7.26 \times 10^{-6} < 0.05$). **E** Uniform Manifold Approximation and Projection (UMAP)[145] dimensionality reduction of patient vector representations for the NYU cohort, each representation is labeled according to the risk group for recurrence, low-risk (blue) and high-risk (orange). These figures correspond to HPL's performance of mean concordance index (c-index) of 0.74 with 95% CI of 0.70-0.80 on the NYU test set. We focus on HPCs that contain at least 10% of the total patients motivated by finding tissue patterns that can generalize across the cohort. Histopathological scrutiny of tile images shows low risk HPCs which include dense chronic inflammatory infiltrates overunning epithelial areas (e.g. **B** HPC 29), while high-risk HPCs are typified by solid and cribriform growth patterns with relatively sparse inflammation (e.g. **C** HPCs 6, 11, 15).

and under-representation of cell types in each HPC, and (3) using manual annotations of pathologists in NYU_2; measuring enrichment or depletion of LUAD histological subtypes such as solid, acinar, papillary, micropapillary, and lepidic using the hypergeometric test.

Through these characterizations we are able to link HPCs' ability to predict survival to direct histological observations of tissue, including LUAD histological growth patterns and cell type densities, and a wide ranging of molecular underpinnings. **Figure 5.8A** shows a bi-hierarchical clustering of HPCs and RNASeq-derived immune signature correlations where positive correlations are showed in red and negative as blue. In addition, Cox log hazard ratio coefficients for overall and recurrence free survival are gradient colored from favoring poor survival outcomes as purple to favoring better survival outcomes as green. **Figure 5.8B** displays a bi-hierarchical clustering of HPCs and cell type over and under-representations, where over-representations are represented in red and under-representations in blue. Finally, **Figure 5.8C** shows a bi-hierarchical clustering of HPCs and LUAD histological growth pattern enrichment; depletion is shown in blue and enrichment in red. The column dendrograms in all subfigures correspond to the bi-hierarchical clustering of HPCs and immune signatures, as we use this setting to more easily relate these analyses in the same context. Throughout this analysis we find that HPCs that are associated with better outcomes are also histopathologically immune 'hot' rather than warm or cold, and have positive correlations with multiple signatures reflecting inflammatory cells, tumor infiltrating leukocytes (TIL) and macrophage regulation as well as neoplastic cells and dead cells, highlighting successful immune infiltration and tumor cell killing. On the other hand, HPCs that are associated with worse outcomes, are exclusively not immune 'hot' and show positive correlations with proliferation, mutation rate, homologous recombination defects and wound healing signatures, while showing under-representation of inflammatory and dead cells, and enrichment for solid histological patterns.

Finally, we further analyze the relationship between HPCs with respect to the different cell type over and under-representations. **Figure 5.9A** shows a partition-based graph abstraction (PAGA)[163] of HPCs and their connections. Each node is colored based on neoplastic, inflammatory, and dead cell enrichment, where over-representations are represented in red and under-representations in blue. **Figure 5.9B** shows a Uniform Manifold Approximation and Projection (UMAP)[145] dimensionality reduction of the vector representations of the 224×224 tissue tiles, each tile label corresponds to the cluster cell type enrichment as in **Figure 5.9A**. These two figures show how neighboring HPCs show similar cell type enrichments or depletions, also revealing how different parts of the space defined by the tile representations capture information about the amount of neoplastic, inflammatory, and dead cells. In addition, **Figure 5.9C** and **Figure 5.9D** provide further insight into representative HPCs for good outcome (**Figure 5.8C** HPC 1) and poor outcome (**Figure 5.8D** HPC 15). **Figure 5.9C** displays scatter plots of each patient's HPC 1 contribution and their relationship with TIL fraction and leukocyte fraction, as well as corresponding individual tiles for HPC 1 from TCGA and NYU_2 cohorts. **Figure 5.9D** shows

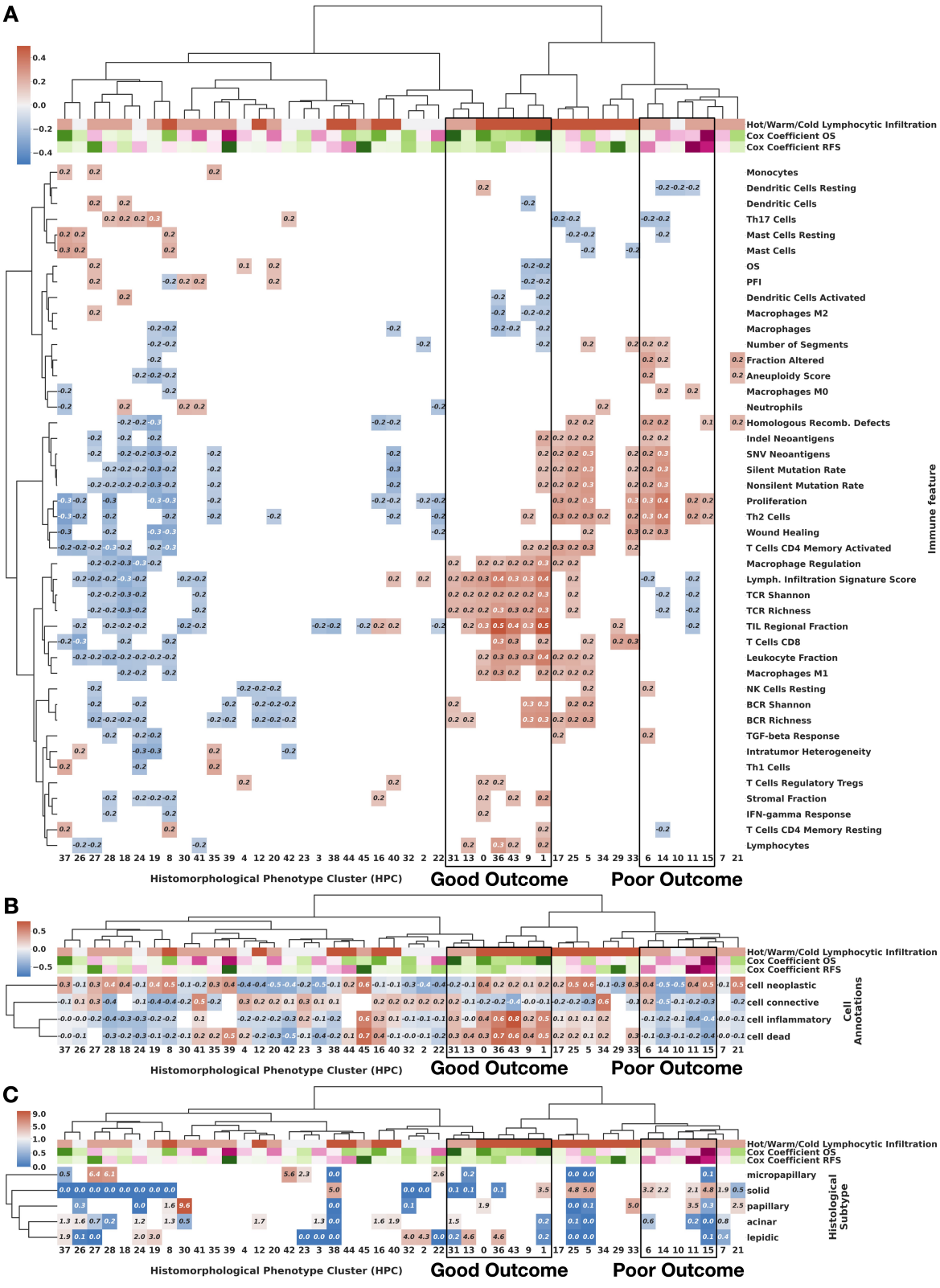

A
Hot/Warm/Cold Lymphocytic Infiltration
Cox Coefficient OS
Cox Coefficient RFS
Monocytes
Dendritic Cells Resting
Dendritic Cells
Th17 Cells
Mast Cells Resting
Mast Cells
OS
PFI
Dendritic Cells Activated
Macrophages M2
Macrophages
Number of Segments
Fraction Altered
Aneuploidy Score
Macrophages M0
Neutrophils
Homologous Recomb. Defects
Indel Neoantigens
SNV Neoantigens
Silent Mutation Rate
Nonsilent Mutation Rate
Proliferation
Th2 Cells
Wound Healing
T Cells CD4 Memory Activated
Macrophage Regulation
Lymph. Infiltration Signature Score
TCR Shannon
TCR Richness
TIL Regional Fraction
T Cells CD8
Leukocyte Fraction
Macrophages M1
NK Cells Resting
BCR Shannon
BCR Richness
TGF-beta Response
Intratumor Heterogeneity
Th1 Cells
T Cells Regulatory Tregs
Stromal Fraction
IFN-gamma Response
T Cells CD4 Memory Resting
Lymphocytes
Immune feature
37 26 27 28 18 24 19 8 30 41 35 39 4 12 20 42 23 3 38 44 45 16 40 32 2 22 31 13 0 36 43 9 1 17 25 5 34 29 33 6 14 10 11 15 7 21
Histomorphological Phenotype Cluster (HPC)
Good Outcome
Poor Outcome
B
Hot/Warm/Cold Lymphocytic Infiltration
Cox Coefficient OS
Cox Coefficient RFS
cell neoplastic
cell connective
cell inflammatory
cell dead
Cell Annotations
37 26 27 28 18 24 19 8 30 41 35 39 4 12 20 42 23 3 38 44 45 16 40 32 2 22 31 13 0 36 43 9 1 17 25 5 34 29 33 6 14 10 11 15 7 21
Histomorphological Phenotype Cluster (HPC)
Good Outcome
Poor Outcome
C
Hot/Warm/Cold Lymphocytic Infiltration
Cox Coefficient OS
Cox Coefficient RFS
micropapillary
solid
papillary
acinar
lepidic
Histological Subtype
37 26 27 28 18 24 19 8 30 41 35 39 4 12 20 42 23 3 38 44 45 16 40 32 2 22 31 13 0 36 43 9 1 17 25 5 34 29 33 6 14 10 11 15 7 21
Histomorphological Phenotype Cluster (HPC)
Good Outcome
Poor Outcome

Figure 5.8: Lung adenocarcinoma (LUAD) survival analysis and Histomorphological Phenotype Cluster (HPC) correlations. A Bi-hierarchical clustering of HPCs and immune signature[66] correlations where positive correlations are shown in read and negative as blue. Cox coefficients for overall and recurrence free survival are colored with a gradient scheme, from favoring death or recurrence as purple to favoring survival or no recurrence as green. HPCs are also colored based on a histological assessment of lymphocytic infiltration[162], 'hot' HPCs are colored as dark red while 'warm' HPCs are colored as red and 'cold' HPCs as white. **B** Bi-hierarchical clustering of HPCs and cell type over and under-representations, over-representations are represented in red and under-representations in blue. **C** Bi-hierarchical clustering of HPCs and LUAD histological subtype enrichment, depletion is shown in blue and enrichment in red. The column dendrogram in all subfigures corresponds to the bi-hierarchical clustering of HPCs and immune signatures, we use this scheme to more easily relate these analyses in the same context. HPCs that are associated with better survival outcomes (low risk of death and recurrence) show positive correlations with being histopathologically immune 'hot' and RNASeq signatures of tumor infiltrating leukocytes (TIL), lymphocyte infiltration signature score, T-cell receptors (TCR), and macrophage regulation; and show over-representations of inflammatory, dead, and neoplastic cells. HPCs associated with worse survival outcomes (high risk of death and recurrence) are exclusively not 'hot' and show positive correlations with proliferation, mutation rate, homologous recombination defects, and wound healing signatures, under-representations for inflammatory and dead cells, and enrichment for solid histological patterns.

scatter plots of each patient's HPC 15 contribution and their relationship with proliferation and Th2 cells, as well as corresponding individual tiles for HPC 15 from TCGA and NYU_2 cohorts. These results show the regional consistency of sets of HPCs associated with specific enrichment of depletion of certain cell types.

5.4 Conclusion

We propose Histomorphological Phenotype Learning (HPL) as an unsupervised methodology to automatically identify histomorphological phenotypes in whole slide images (WSI). HPL consists of four major steps, (1) WSI pre-processing, (2) self-supervised learning of tissue tiles, (3) tissue tile representation clustering into Histomorphological Phenotype Clusters (HPCs), and (4) HPC characterization. This approach therefore performs learning and clustering of tiles from histopathology WSIs in an unbiased manner, removing the need for expensive annotations, and providing clear insights in the decision process for a selected task.

From a technical point of view, the self-supervised representations are learned using Barlow Twins[123]. Apart from a comparison with other semi or fully-supervised approaches to classify breast cancer regions[55], this novel method has not yet been extensively applied to histopathology. Very recently developed by Zbontar *et al.*[123], Barlow Twins was developed to learn representations invariant to distortions while improving several shortcomings present in many existing self-supervised methods (avoidance of collapse by de-correlation, large batches, or architecture asymmetry are not required). In our pipeline, we use the Leiden algorithm[158] to define clusters from a network of tile vector representations, where each tile vector representations encodes morphological patterns found in tissue tiles. Consequently, each WSI or patient (one or more

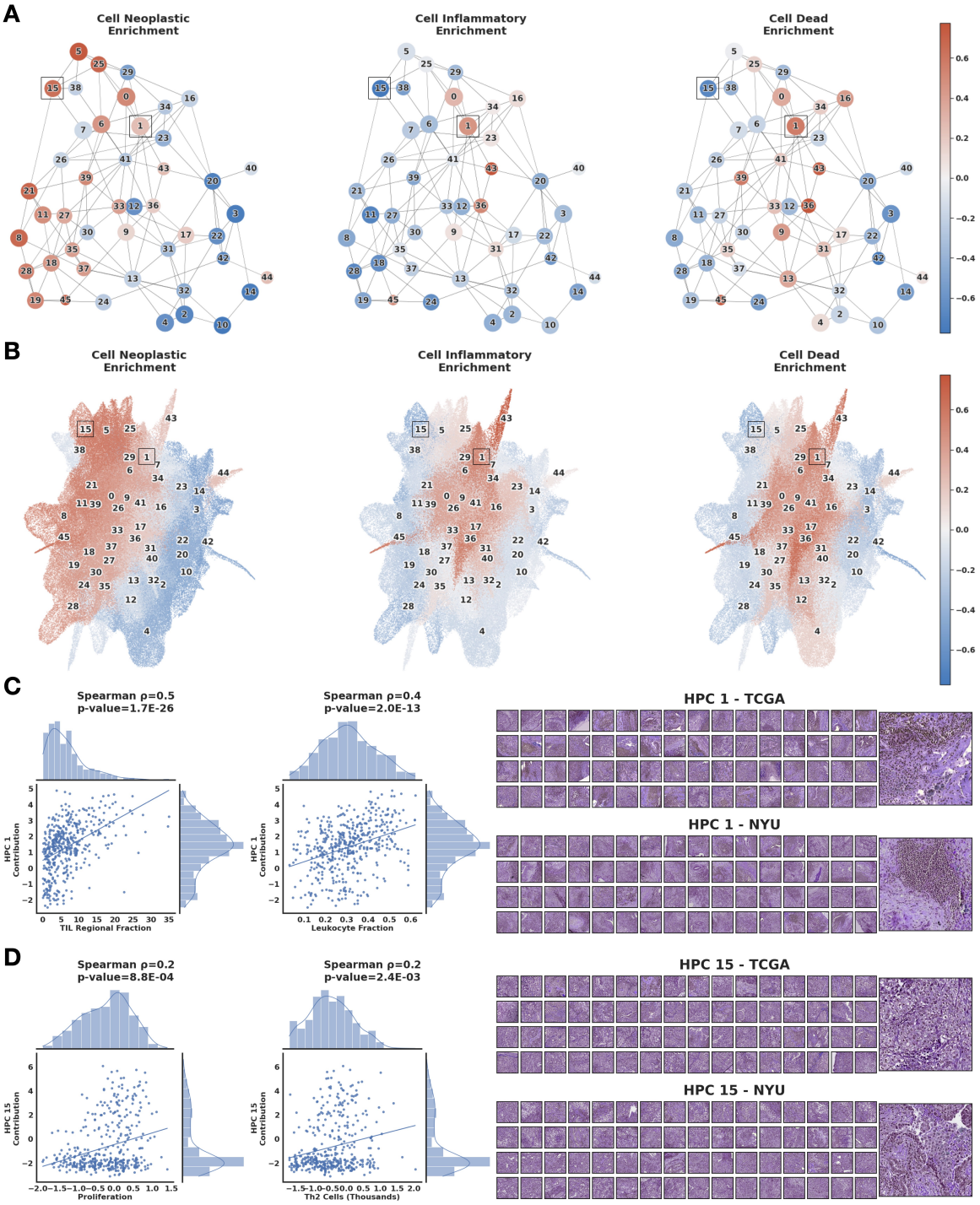


Figure 5.9: Lung adenocarcinoma (LUAD) survival Histomorphological Phenotype Cluster (HPC) representations and reference HPCs analysis. A Partition-based graph abstraction (PAGA)[163] of HPCs and their similarities. Each node is colored based on neoplastic, inflammatory, and dead cell enrichment, where over-representations are represented in red and under-representations in blue. **B** Uniform Manifold Approximation and Projection (UMAP)[145] dimensionality reduction of the vector representations of the 224×224 tissue tiles, each tile label corresponds to the cluster cell type enrichment as in **Figure A**. **C** Scatter plot between HPC 1 contribution and omic-based immune signatures of each patient (tumor infiltration leukocytes (TIL) and leukocyte fraction). On the right we show representative HPC 1 tiles from TCGA and NYU_2 cohorts. **D** Scatter plot between HPC 15 contribution and omic-based immune signatures of each patient (proliferation and Th2 cells). On the right we show representative HPC 15 tiles from TCGA and NYU_2 cohorts.

WSIs) can be described by a simple vector which describes its Leiden cluster contributions. Like a simplified barcode, it is this representation which can then be used to interpret the slide or patient. This approach introduces three important advantages compared to other deep learning approaches: firstly our HPCs are created solely from tissue information, not by correlation with clinical annotations; secondly, it introduces interpretability by allowing visual inspection of the HPC tiles and by proving statistical significance with respect to the clinical prediction task; and thirdly, it facilitates integration with other data that can be represented as vectors, such as gene expression or mutations.

As we show in **Figure 5.4** and **Figure 5.5**, we use this method to analyze WSIs from patients with lung cancer and assess its ability to distinguish adenocarcinoma (LUAD) from squamous cell carcinoma (LUSC). We use the TCGA cohort as a training set and to a posteriori identify HPCs that are most relevant to the task. On the independent cohort from NYU (NYU_1), the average AUC of 0.99 is even better than results previously obtained by training Inception-V3 on analogous cohorts[13]. Such results are in line with performance obtained with other recent strategies: AUCs of 0.94-0.96 were obtained using a unified memory mechanism on WSIs rather than tiles[164], AUC of 0.915 using a pre-trained ResNet50 followed by multiple instance learning, or 0.968 using a weak supervision expectation-maximization framework[165]. In several supervised deep learning approaches, the decision is described as a "black box" and understanding how the classifier makes a decision is not straightforward. In contrast, our methodology provides clear insight on the relevant HPCs and the correlation with the classification task. Our method delivers clusters that show distinct patterns of lung type and supports this conclusion by providing evidence of significance, and correlations between WSI labels and lung type purity. Furthermore, it allows visualization of the HPCs as individual tiles or in the context of WSI, providing a simple way for a pathologist to interpret tissue patterns with clinical annotations.

We also apply HPL methodology to identify and study histomorphological phenotypes specifically in lung adenocarcinoma. **Figure 5.6** and **Figure C.6** show how HPL's HPCs recapitulate established LUAD growth patterns, with each World Health Organization (WHO) described growth pattern, as well as the mucinous subtype, predominating in at least one cluster, and usually more than one with different levels of lymphocyte infiltration ('hotness'[152], **Figure C.8**). No fewer than 8 clusters are predominantly solid growth pattern, being differentiated by the presence of stromal bands, lymphocyte infiltration, cohesiveness, necrosis and folding artefacts. Other HPCs align closely with non-tumor patterns, such as bronchial cartilage, vascular smooth muscle, haemorrhage, and metaplastic changes in architecturally normal lung. Cluster contributions were examined for their ability to predict overall survival and recurrence free survival. Our results show that lymphocyte density (measured both by histological appearances and by association with signatures from bulk RNASeq data) in tumor regions is relevant for overall survival prediction. This is in line with other literature results such as Chen *et al.*[27] where the authors showed that tumor infiltrating lymphocyte (TIL) is an important marker for cancer prognosis

by using histological and gene expression data. For the overall survival prediction, our method achieved a mean concordance-index (c-index) of 0.65 over the external NYU cohort (NYU_2), and of 0.60 over TCGA only using WSIs, as compare to Chen *et al.* where the authors used histological and gene expression data achieving a mean c-index of 0.59, or 0.548 only using histological data. HPL also highlights that clusters associated with worse outcomes contain more classically high-risk growth patterns, while the presence of more predominant low-grade lepidic growth favors low risk. This mirrors the current practice in which histological grade is estimated by the pathologist from growth pattern, an approach that is well accepted and corroborated (e.g. the multi-hospital study from Sun *et al.*[166]).

Figure 5.7 shows HPL's mean concordance-index of 0.74 in recurrence free survival which is in line with the recently latest grading scheme suggested in Moreira *et al.*[5]. Moreira *et al.* proposes a revised grading system based on expert histopathological assessment of LUAD WSIs. Classical pathological studies have established that the proportions of high-risk growth patterns (solid, micropapillary, and complex glandular patterns such as the cribriform pattern) have predictive value for the risk of lung adenocarcinoma recurrence[167]. HPL similarly finds that poor outcome HPCs are dominated by solid and cribriform patterns as well as necrosis, another recognized correlate of poor outcome[168]. However, micropapillary-predominant HPCs are not strongly related to poor outcome. There are two likely reasons: first, our use of 5X images may not convey adequate cellular detail to distinguish all micropapillae, and second, micropapillary growth pattern is by nature a combination of a fractional area of small micropapillae against a majority background of an acinar or papillary pattern, and these background appearances may predominate in tile representation.

Finally, we analyze the HPCs through three different sources of information; immune signatures derived from genomic and transcriptomic profiles, automated cellular annotations from tissue images, and histological subtype annotations from an expert pathologist. Along with the survival analysis, these correlations provide a comprehensive overview of the information contained in the HPCs and their predictive potential, showing that tissue patterns that favor better prognostic outcomes suggest an effective immune response[152] and low-grade disease, while HPCs that favor a patient's death or recurrence show a lack of immune response, high-grade tumor morphology, and cellular proliferation. Interestingly, the wound healing process is shown to be positively correlated with HPCs associated with poor prognosis. Mechanisms and pathways associated with wound healing have indeed been pointed as being implicated in cancer proliferation and metastasis[169,170].

Chapter 6

Conclusion and discussion

This thesis proposes the use of unsupervised learning methods for cancer tissue representations. The proposition is motivated by their ability to alleviate the bottleneck of high-quality annotations and also as a means to further study tissue morphologies through growth patterns, cellular densities, and associations with molecular underpinnings. It makes the following contributions in GAN-based and self-supervised learning-based methods for that purpose.

In **Chapter 3** we introduced PathologyGAN, a generative adversarial network (GAN) for unsupervised representation learning of cancer tissue images. PathologyGAN generates high-fidelity tissue images through a structured and interpretable latent space. We evaluated the model's ability to reproduce morphological and cellular features in tissue through quantitative measures and pathologist analyses. In addition, we showed the representation learning properties of the model where its latent space holds a structure based on key tissue characteristics such as density of cancer cells, lymphocytes, stroma, or tissue type morphologies.

Chapter 4 further developed PathologyGAN by tackling its main limitation, the ability to find representations of real tissue images. It proposed to modify PathologyGAN by introducing an encoder that train concurrently with the GAN, learning to map tissue images back to the GAN's latent space. We showed evidence of how the tissue representations from real images can be used in three different scenarios: We visualized representations with clinical annotations, we quantified how informative they are by training a linear classifier to predict tissue types, and we demonstrated their suitability on a multiple instance learning (MIL) task, predicting the presence of tumor on whole slide images (WSIs).

Finally, **Chapter 5** focused on self-supervised learning for cancer tissue representation. It introduced Histomorphological Phenotype Learning (HPL), a methodology to automatically identify and cluster tissue morphologies through self-supervised learning and community detection. The proposed methodology is easily interpretable, allowing pathologists to scrutinize tissue patterns and their relations to annotations such as cancer type, overall survival, recurrence free survival, or molecular phenotypes. Furthermore, it provided evidence of its applicability by extracting tissue morphologies that can discriminate between cancer types and be prognostic in

overall and recurrence free survival. The presented results aligned with previous histopathology literature.

Overall, this thesis studied the use of Generative Adversarial Networks (GANs) and self-supervised learning for unsupervised learning of cancer tissue representations. It evaluated the significance of the tissue-related information enclosed in the representations, relating them with several clinically relevant annotations such as growth patterns, patient outcomes, or omics-based signatures. It showed the benefit of the interpretability introduced by these models, allowing patholgists to analyze representations with annotations. We envision these unsupervised approaches that extract information solely from tissue as powerful discovery approaches that can be used to study tissue morphologies across cancer types, giving pathologists and scientists a tool to further study the disease.

6.1 Future directions

In **Chapter 3**, we propose PathologyGAN as a generative adversarial network that combines representation learning properties and effectively learns and reproduces cellular and morphological characteristics found in tissue. However, this model also has its limitations.

PathologyGAN is able to generate tissue tiles of 224×224 resolution. In the context of a whole slide image (WSI) or Tissue Micro-Array (TMA), this is a small portion of the total tissue area. Although tiles of such resolution can capture cell types densities or tissue types accurately, larger tissue areas would be necessary in order to capture more complex tissue dynamics. Motivated by this idea, PathologyGAN could be extended to achieve higher image resolutions[171], providing representations and generating samples for larger tissue sections. This line of work would imply several challenges, such as stable GAN training for resolutions larger than 2048×2048 and also learning representations from a limited amount of samples. The latter is especially important if we wanted to accurately capture known tissue dynamics in WSIs or TMAs. Furthermore, since RNA-seq samples are taken along with tissue biopsies in WSIs, further developments would include conditioning tissue generation based on different RNA-seq signatures, tying tissue context dynamics and RNA-seq information.

An appealing feature of PathologyGAN is its latent space and translations that correspond to feature transformations in the generated images. Currently, the performed linear vector operations make the assumption of PathologyGAN's latent space being Euclidean. The vector operations translate into coarse tissue feature transformations. Nonetheless, this is a strong assumption and further study on the latent space could allow more fine-grained or more complex vector operations. An interesting approach would be to combine the existing model with a constraint such that it induces disentanglement in the model's latent space, perhaps not only *InfoGAN*'s solution but alternatively a redundancy reduction among the representation dimensions as in *Barlow Twins*.

PathologyGAN's tissue representations might be susceptible to batch effect across institutions. PathologyGAN uses Adaptative Instance Normalization (AdaIN) as a building block to generate samples based on each instance representation. AdaIN was first proposed in *neural style transfer* to transform an image based on the content style of another. *i.e.* manipulate color, contrast, or texture from an image based on another. The inherent architecture of AdaIN makes PathologyGAN potentially susceptible to the imaging variations across institutions, also referred to as batch effect[150]. This phenomenon introduces slight variations in resolution, color, contrast, and brightness in WSIs and TMAs due to the imaging process of different institutions. As a result, tissue representations could introduce non-relevant tissue information over tissue feature characteristics, causing the representations not to generalize on a given task across institutions. Combining self-supervised learning and generative adversarial networks[172] could mitigate this issue, constraining tissue representations with slight imaging variations to be similar or relatively close in the latent space. Alternatively, constraining the GAN's latent space with confounding or nuance factor removal could reduce the impact of the batch effect. In this setting, we would want to make the latent representations invariant to the nuance factor, in this case the institution where the WSI or TMA comes from.

As an alternative to GANs, diffusion models[173–177] offer a promising approach to high-quality sample generation. These models have a more stable training process than GANs and do not suffer from common GANs shortcomings such as collapse or complicated min-max optimization between generator and discriminator. Originally, diffusion models were more computationally demanding than GANs but recent developments have solved this problem, offering to train the generative model in a single GPU[176] . Combining diffusion models with representation learning properties[178,179] would be a promising approach to learning tissue representations, capturing more subtle details than GANs[177] and possibly creating richer representations.

Generative models could be used as an educational tool, providing tissue samples with rare cellular characteristics. They could mimic possible transitions of tissue (e.g. from tumor to high lymphocyte infiltration) or enable the study of borderline cases between generated images and pathologists' interpretations (e.g. atypia). In addition, these models could be used to generate synthetic samples to improve classifier robustness and generalization, helping to avoid overfitting and providing samples with rare tissue pathologies.

As introduced in **Chapter 5**, self-supervised learning offers an alternative and more efficient approach to learning representations when the generative process is not a requirement. So far, *HPL* only uses tile images to create representations but it could be expanded to handle multiple modalities of data such as RNA-seq data. In addition, *HPL* could use multiple magnifications to create tissue representations, the current architecture uses a single magnification and this could be a limiting factor in creating more complex representations. For instance, as mentioned in the conclusion section of **Chapter 5**, differentiating the micropapillary histological subtype in lung adenocarcinoma requires a view of 20x or 40x magnification to distinguish all micropapillae

where some other histological patterns are better distinguished at higher magnifications such as 5x. By utilizing multiple magnifications, we could combine tissue details only appreciated at the different levels of zoom into a single representation.

Tissue representations created from the self-supervised model appear to be driven by the density of different cell types such as inflammatory, connective, or epithelial while morphological patterns seem to have a second-order impact. This idea is based on the communities created by the nearest neighbors graph, where tiles in the same community have similar enrichment of cell types while it can combine different histological subtypes like acinar or papillary. We hypothesize that cell type densities are driven by image texture while histological subtypes are based on morphological patterns. Perhaps alternative backbone architectures to a CNN such vision transformers (ViTs) could pick up morphological patterns in the images[180,181].

Whether tissue representations are created with generative models or self-supervised methods, understanding the impact of batch effect would be relevant to creating more robust and generalizable representations. Minimizing the impact of the variation introduced by different imaging processes would benefit finding tissue morphologies that can generalize across different institutional cohorts. Previous suggestions on confounding or nuance factor removal would also benefit representations created by the self-supervised learning framework.

In **Figure 5.8A** we show clear correlations between clusters (HPCs) and RNASeq-derived immune signatures. These results suggest that histomorphological phenotype clusters can be used to provide insight into gene expression directly from tissue without the need for expensive sequencing. Spatial transcriptomics would be especially relevant on this topic[182–184] since these methods precisely relate histological samples with RNASeq-derived information, providing a tool to verify the validity of histomorphological phenotype clusters as a proxy for RNASeq-derived signatures.

Finally, in **Figure C.2** we show an initial approach to a Pan-cancer study where *HPL* can differentiate between 7 different cancer types. We propose to further study how to scale the self-supervised models so they can be used across different types of cancer. Given previous work[123], some of these methods benefit from higher dimensional representations, and finding the optimal size of a deep learning encoder and the computational resources would allow doing larger cohort studies, ultimately seeking to find tissue morphologies that can be diagnostic and prognostic not only for each cancer type but across them.

Bibliography

[1] Andrion, A. *et al.* Malignant mesothelioma of the pleura: interobserver variability. *J. clinical pathology* **48**, 856–860 (1995).

[2] Kujan, O. *et al.* Why oral histopathology suffers inter-observer variability on grading oral epithelial dysplasia: an attempt to understand the sources of variation. *Oral oncology* **43**, 224–231 (2007).

[3] Warth, A. *et al.* Interobserver variability in the application of the novel iaslc/ats/ers classification for pulmonary adenocarcinomas. *Eur. respiratory journal* **40**, 1221–1227 (2012).

[4] Ozkan, T. A. *et al.* Interobserver variability in gleason histological grading of prostate cancer. *Scand. journal urology* **50**, 420–424 (2016).

[5] Moreira, A. L. *et al.* A grading system for invasive pulmonary adenocarcinoma: A proposal from the international association for the study of lung cancer pathology committee. *J. Thorac. Oncol.* **15**, 1599–1610 (2020).

[6] Almendro, V., Marusyk, A. & Polyak, K. Cellular heterogeneity and molecular evolution in cancer. *Annu. Rev. Pathol. Mech. Dis.* **8**, 277–302 (2013).

[7] de Sousa, V. M. L. & Carvalho, L. Heterogeneity in lung cancer. *Pathobiology* **85**, 96–107 (2018).

[8] Beck, A. H. *et al.* Systematic analysis of breast cancer morphology uncovers stromal features associated with survival. *Sci Transl Med* **3**, 108ra113, DOI: 10.1126/scitranslmed.3002564 (2011).

[9] Yuan, Y. *et al.* Quantitative image analysis of cellular heterogeneity in breast tumors complements genomic profiling. *Sci. Transl. Med.* **4**, 157ra143 (2012).

[10] Cireşan, D. C., Giusti, A., Gambardella, L. M. & Schmidhuber, J. Mitosis detection in breast cancer histology images with deep neural networks. In Mori, K., Sakuma, I., Sato, Y., Barillot, C. & Navab, N. (eds.) *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013* (Springer Berlin Heidelberg, 2013).

[11] Cruz-Roa, A. *et al.* Automatic detection of invasive ductal carcinoma in whole slide images with convolutional neural networks. In *Medical Imaging 2014: Digital Pathology*, vol. 9041, 904103 (SPIE, 2014).

[12] Wang, H. *et al.* Mitosis detection in breast cancer pathology images by combining handcrafted and convolutional neural network features. *J. Med. Imaging* **1**, 034003 (2014).

[13] Coudray, N. *et al.* Classification and mutation prediction from non–small cell lung cancer histopathology images using deep learning. *Nat. medicine* **24**, 1559–1567 (2018).

[14] Hekler, A. *et al.* Deep learning outperformed 11 pathologists in the classification of histopathological melanoma images. *Eur. J. Cancer* **118**, 91–96 (2019).

[15] Bulten, W. *et al.* Automated deep-learning system for gleason grading of prostate cancer using biopsies: a diagnostic study. *The Lancet Oncol.* **21**, 233–241 (2020).

[16] Li, Y. & Ping, W. Cancer metastasis detection with neural conditional random field. *arXiv preprint arXiv:1806.07064* (2018).

[17] Nagpal, K. *et al.* Development and validation of a deep learning algorithm for improving gleason scoring of prostate cancer. *NPJ digital medicine* **2**, 1–10 (2019).

[18] Sirinukunwattana, K. *et al.* Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images. *IEEE transactions on medical imaging* **35**, 1196–1206 (2016).

[19] Graham, S. *et al.* Hover-net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images. *Med. Image Analysis* **58**, 101563 (2019).

[20] Xing, F., Xie, Y. & Yang, L. An automatic learning-based framework for robust nucleus segmentation. *IEEE transactions on medical imaging* **35**, 550–566 (2015).

[21] Graham, S. *et al.* Mild-net: Minimal information loss dilated network for gland instance segmentation in colon histology images. *Med. image analysis* **52**, 199–211 (2019).

[22] Lin, H. *et al.* Fast scannet: Fast and dense analysis of multi-gigapixel whole-slide images for cancer metastasis detection. *IEEE transactions on medical imaging* **38**, 1948–1958 (2019).

[23] Swiderska-Chadaj, Z. *et al.* Learning to detect lymphocytes in immunohistochemistry with deep learning. *Med. image analysis* **58**, 101547 (2019).

[24] Kather, J. N. *et al.* Pan-cancer image-based detection of clinically actionable genetic alterations. *Nat. cancer* **1**, 789–799 (2020).

[25] Fu, Y. *et al.* Pan-cancer computational histopathology reveals mutations, tumor composition and prognosis. *Nat. Cancer* **1**, 800–810 (2020).

[26] Johannet, P. *et al.* Using machine learning algorithms to predict immunotherapy response in patients with advanced melanoma. *Clin. Cancer Res.* **27**, 131–140 (2021).

[27] Chen, R. J. *et al.* Pan-cancer integrative histology-genomic analysis via multimodal deep learning. *Cancer Cell* (2022).

[28] Li, B., Li, Y. & Eliceiri, K. W. Dual-stream multiple instance learning network for whole slide image classification with self-supervised contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14318–14328 (2021).

[29] Yao, J., Zhu, X., Jonnagaddala, J., Hawkins, N. & Huang, J. Whole slide images based cancer survival prediction using attention guided deep multiple instance learning networks. *Med. Image Analysis* **65**, 101789 (2020).

[30] Kather, J. N. *et al.* Predicting survival from colorectal cancer histology slides using deep learning: A retrospective multicenter study. *PLoS medicine* **16**, e1002730 (2019).

[31] Wulczyn, E. *et al.* Deep learning-based survival prediction for multiple cancer types using histopathology images. *PLoS One* **15**, e0233678 (2020).

[32] Courtiol, P. *et al.* Deep learning-based classification of mesothelioma improves prediction of patient outcome. *Nat. Medicine* **25**, 1519–1525 (2019).

[33] Campanella, G., Silva, V. W. K. & Fuchs, T. J. Terabyte-scale deep multiple instance learning for classification and localization in pathology. *arXiv preprint arXiv:1805.06983* (2018).

[34] Akbar, S. & Martel, A. L. Cluster-based learning from weakly labeled bags in digital pathology. *arXiv preprint arXiv:1812.00884* (2018).

[35] Li, C. *et al.* Weakly supervised mitosis detection in breast histopathology images using concentric loss. *Med. image analysis* **53**, 165–178 (2019).

[36] Bokhorst, J.-M. *et al.* Learning from sparsely annotated data for semantic segmentation in histopathology images. In *International Conference on Medical Imaging with Deep Learning–Full Paper Track* (2018).

[37] Liu, J. *et al.* An end-to-end deep learning histochemical scoring system for breast cancer tma. *IEEE transactions on medical imaging* **38**, 617–628 (2018).

[38] Wang, D., Khosla, A., Gargeya, R., Irshad, H. & Beck, A. H. Deep learning for identifying metastatic breast cancer. *arXiv preprint arXiv:1606.05718* (2016).

[39] Noorbakhsh, J. *et al.* Pan-cancer classifications of tumor histological images using deep learning. *Prepr. at https://www. biorxiv. org/content/10.1101/715656v1* (2019).

[40] Ciga, O., Xu, T. & Martel, A. L. Self supervised contrastive learning for digital histopathology. *Mach. Learn. with Appl.* **7**, 100198 (2022).

[41] Xu, J. *et al.* Stacked sparse autoencoder (ssae) for nuclei detection on breast cancer histopathology images. *IEEE transactions on medical imaging* **35**, 119–130 (2016).

[42] Mahmood, F. *et al.* Deep adversarial training for multi-organ nuclei segmentation in histopathology images. *IEEE Trans. Med. Imaging* **39**, 3257–3267 (2020).

[43] de Bel, T. *et al.* Automatic segmentation of histopathological slides of renal tissue using deep learning. In *Medical Imaging 2018: Digital Pathology*, vol. 10581, 1058112 (International Society for Optics and Photonics, 2018).

[44] Gadermayr, M., Gupta, L., Klinkhammer, B. M., Boor, P. & Merhof, D. Unsupervisedly training gans for segmenting digital pathology with automatically generated annotations. In Cardoso, M. J. *et al.* (eds.) *Proceedings of The 2nd International Conference on Medical Imaging with Deep Learning*, vol. 102 of *Proceedings of Machine Learning Research*, 175–184 (PMLR, 2019).

[45] Gadermayr, M. *et al.* Generative adversarial networks for facilitating stain-independent supervised and unsupervised segmentation: a study on kidney histology. *IEEE transactions on medical imaging* **38**, 2293–2302 (2019).

[46] Rana, A., Yauney, G., Lowe, A. & Shah, P. Computational histological staining and destaining of prostate core biopsy RGB images with Generative Adversarial Neural Networks. *2018 17th IEEE Int. Conf. on Mach. Learn. Appl. (ICMLA)* (2018).

[47] Xu, Z., Moro, C. F., Bozóky, B. & Zhang, Q. GAN-based Virtual Re-Staining: A promising solution for whole slide image analysis (2019).

[48] Zanjani, F. G., Zinger, S., Bejnordi, B. E., van der Laak, J. A. W. M. & de With, P. H. N. Stain normalization of histopathology images using generative adversarial networks. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, 573–577 (2018).

[49] Krause, J. *et al.* Deep learning detects genetic alterations in cancer histology generated by adversarial networks. *The J. Pathol.* **n/a** (2021).

[50] Levine, A. B., Peng, J., Farnell, D., Nursey, M. & et al. Synthesis of diagnostic quality cancer pathology images by generative adversarial networks. *The J. Pathol.* **252**, 178–188 (2020).

[51] Deshpande, S., Minhas, F., Graham, S. & Rajpoot, N. Safron: Stitching across the frontier network for generating colorectal cancer histology images. *Med. Image Analysis* **77**, 102337 (2022).

[52] Hu, B. *et al.* Unsupervised learning for cell-level visual representation in histopathology images with generative adversarial networks. *IEEE J. Biomed. Heal. Informatics* **23**, 1316–1328 (2019).

[53] Hou, L. *et al.* Sparse autoencoder for unsupervised nucleus detection and representation in histopathology images. *Pattern recognition* **86**, 188–200 (2019).

[54] Lafarge, M. W., Pluim, J. P. & Veta, M. Orientation-disentangled unsupervised representation learning for computational pathology. *arXiv preprint arXiv:2008.11673* (2020).

[55] Zhang, L., Amgad, M. & Cooper, L. A. D. A histopathology study comparing contrastive semi-supervised and fully supervised learning (2021).

[56] Dehaene, O., Camara, A., Moindrot, O., de Lavergne, A. & Courtiol, P. Self-supervision closes the gap between weak and strong supervision in histology (2020).

[57] Chen, R. J. *et al.* Scaling vision transformers to gigapixel images via hierarchical self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 16144–16155 (2022).

[58] Koohbanani, N. A., Unnikrishnan, B., Khurram, S. A., Krishnaswamy, P. & Rajpoot, N. Self-path: Self-supervision for classification of pathology images with limited annotations. *IEEE Transactions on Med. Imaging* **40**, 2845–2856 (2021).

[59] Sahasrabudhe, M. *et al.* Self-supervised nuclei segmentation in histopathological images using attention. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 393–402 (Springer, 2020).

[60] Abbet, C., Zlobec, I., Bozorgtabar, B. & Thiran, J.-P. Divide-and-rule: Self-supervised learning for survival analysis in colorectal cancer. In Martel, A. L. *et al.* (eds.) *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*, 480–489 (Springer International Publishing, 2020).

[61] Macenko, M. *et al.* A method for normalizing histology slides for quantitative analysis. In *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 1107–1110 (2009).

[62] Kather, J. N., Halama, N. & Marx, A. 100,000 histological images of human colorectal cancer and healthy tissue, DOI: 10.5281/zenodo.1214456 (2018).

[63] Raczkowski, L., Mozejko, M., Zambonelli, J. & Szczurek, E. Ara: accurate, reliable and active histopathological image classification framework with bayesian deep learning. *Sci. Reports* **9**, 14347 (2019).

[64] Kather, J. N. *et al.* Multi-class texture analysis in colorectal cancer histology. *Sci. Reports* **6**, 27988 (2016).

[65] Reinhard, E., Adhikhmin, M., Gooch, B. & Shirley, P. Color transfer between images. *IEEE Comput. Graph. Appl.* **21**, 34–41 (2001).

[66] Thorsson, V. *et al.* The immune landscape of cancer. *Immunity* **48**, 812–830.e14 (2018).

[67] Kingma, D. P. & Welling, M. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings* (2014).

[68] Blei, D. M., Kucukelbir, A. & McAuliffe, J. D. Variational inference: A review for statisticians. *J. Am. Stat. Assoc.* **112**, 859–877 (2017).

[69] Hoffman, M. D., Blei, D. M., Wang, C. & Paisley, J. W. Stochastic variational inference. *J. Mach. Learn. Res.* **14**, 1303–1347 (2013).

[70] Bengio, Y., Courville, A. & Vincent, P. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis Mach. Intell.* **35**, 1798–1828 (2013).

[71] Eastwood, C. & Williams, C. K. I. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations* (2018).

[72] Higgins, I. *et al.* Towards a definition of disentangled representations (2018).

[73] Do, K. & Tran, T. Theory and evaluation metrics for learning disentangled representations (2019).

[74] Kim, H. & Mnih, A. Disentangling by factorising. In Dy, J. & Krause, A. (eds.) *Proceedings of the 35th International Conference on Machine Learning*, vol. 80 of *Proceedings of Machine Learning Research*, 2649–2658 (PMLR, 2018).

[75] Higgins, I. *et al.* beta-vae: Learning basic visual concepts with a constrained variational framework. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings* (OpenReview.net, 2017).

[76] Burgess, C. P. *et al.* Understanding disentangling in β-vae (2018).

[77] Watanabe, S. Information theoretical analysis of multivariate correlation. *IBM J. Res. Dev.* **4**, 66–82 (1960).

[78] Chen, R. T. Q., Li, X., Grosse, R. B. & Duvenaud, D. K. Isolating sources of disentanglement in variational autoencoders. In Bengio, S. *et al.* (eds.) *Advances in Neural Information Processing Systems*, vol. 31 (Curran Associates, Inc., 2018).

[79] Goodfellow, I. *et al.* Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. & Weinberger, K. (eds.) *Advances in Neural Information Processing Systems*, vol. 27 (Curran Associates, Inc., 2014).

[80] Goodfellow, I. Nips 2016 tutorial: Generative adversarial networks (2016).

[81] Arjovsky, M. & Bottou, L. Towards principled methods for training generative adversarial networks (2017).

[82] Metz, L., Poole, B., Pfau, D. & Sohl-Dickstein, J. Unrolled generative adversarial networks (2017).

[83] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B. & Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In Guyon, I. *et al.* (eds.) *Advances in Neural Information Processing Systems 30*, 6626–6637 (Curran Associates, Inc., 2017).

[84] Brock, A., Donahue, J. & Simonyan, K. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations* (2019).

[85] Karras, T., Laine, S. & Aila, T. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).

[86] Miyato, T., Kataoka, T., Koyama, M. & Yoshida, Y. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations* (2018).

[87] Zhang, H., Goodfellow, I., Metaxas, D. & Odena, A. Self-attention generative adversarial networks. In *Proceedings of the 36th International Conference on Machine Learning*, vol. 97 of *Proceedings of Machine Learning Research*, 7354–7363 (PMLR, 2019).

[88] Salimans, T. *et al.* Improved techniques for training gans (2016).

[89] Bińkowski, M., Sutherland, D. J., Arbel, M. & Gretton, A. Demystifying mmd gans (2018).

[90] Lopez-Paz, D. & Oquab, M. Revisiting classifier two-sample tests. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings* (2017).

[91] Barratt, S. & Sharma, R. A note on the inception score (2018).

[92] Deng, J. *et al.* Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255 (Ieee, 2009).

[93] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2818–2826 (2016).

[94] Fréchet, M. Sur la distance de deux lois de probabilité. *COMPTES RENDUS HEBDOMADAIRES DES SEANCES DE L ACADEMIE DES SCIENCES* **244**, 689–692 (1957).

[95] Radford, A., Metz, L. & Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings* (2016).

[96] Jolicoeur-Martineau, A. The relativistic discriminator: a key element missing from standard GAN. In *International Conference on Learning Representations* (2019).

[97] Arjovsky, M., Chintala, S. & Bottou, L. Wasserstein generative adversarial networks. In Precup, D. & Teh, Y. W. (eds.) *Proceedings of the 34th International Conference on Machine Learning*, vol. 70 of *Proceedings of Machine Learning Research*, 214–223 (PMLR, 2017).

[98] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V. & Courville, A. C. Improved training of wasserstein gans. In Guyon, I. *et al.* (eds.) *Advances in Neural Information Processing Systems*, vol. 30 (Curran Associates, Inc., 2017).

[99] Karras, T., Aila, T., Laine, S. & Lehtinen, J. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations* (2018).

[100] Karnewar, A. & Wang, O. Msg-gan: Multi-scale gradients for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020).

[101] Chen, X. *et al.* Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I. & Garnett,

R. (eds.) *Advances in Neural Information Processing Systems*, vol. 29 (Curran Associates, Inc., 2016).

[102] Makhzani, A., Shlens, J., Jaitly, N. & Goodfellow, I. Adversarial autoencoders. In *International Conference on Learning Representations* (2016).

[103] Donahue, J. & Simonyan, K. Large scale adversarial representation learning. In Wallach, H. *et al.* (eds.) *Advances in Neural Information Processing Systems*, vol. 32 (Curran Associates, Inc., 2019).

[104] Donahue, J., Krähenbühl, P. & Darrell, T. Adversarial feature learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings* (OpenReview.net, 2017).

[105] Karras, T. *et al.* Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020).

[106] Karras, T. *et al.* Alias-free generative adversarial networks. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P. & Vaughan, J. W. (eds.) *Advances in Neural Information Processing Systems*, vol. 34, 852–863 (Curran Associates, Inc., 2021).

[107] Pidhorskyi, S., Adjeroh, D. A. & Doretto, G. Adversarial latent autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020).

[108] Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (Association for Computational Linguistics, Minneapolis, Minnesota, 2019).

[109] Wu, J., Wang, X. & Wang, W. Y. Self-supervised dialogue learning. *Proc. 57th Annu. Meet. Assoc. for Comput. Linguist.* (2019).

[110] Jabri, A., Owens, A. & Efros, A. A. Space-time correspondence as a contrastive random walk. *Adv. Neural Inf. Process. Syst.* (2020).

[111] Kong, Q., Wei, W., Deng, Z., Yoshinaga, T. & Murakami, T. Cycle-contrast for self-supervised video representation learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. & Lin, H. (eds.) *Advances in Neural Information Processing Systems*, vol. 33, 8089–8100 (Curran Associates, Inc., 2020).

[112] Han, T., Xie, W. & Zisserman, A. Self-supervised co-training for video representation learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. & Lin, H. (eds.) *Advances in Neural Information Processing Systems*, vol. 33, 5679–5690 (Curran Associates, Inc., 2020).

[113] Valverde, F. R., Valeria Hurtado, J. & Valada, A. There is more than meets the eye: Self-supervised multi-object detection and tracking with sound by distilling multimodal knowledge. *2021 IEEE/CVF Conf. on Comput. Vis. Pattern Recognit. (CVPR)* (2021).

[114] Morgado, P., Nvasconcelos, N., Langlois, T. & Wang, O. Self-supervised generation of spatial audio for 360°video. In Bengio, S. *et al.* (eds.) *Advances in Neural Information Processing Systems*, vol. 31 (Curran Associates, Inc., 2018).

[115] Tonekaboni, S., Eytan, D. & Goldenberg, A. Unsupervised representation learning for time series with temporal neighborhood coding. In *International Conference on Learning Representations* (2021).

[116] Yue, Z. *et al.* Ts2vec: Towards universal representation of time series (2021).

[117] Franceschi, J.-Y., Dieuleveut, A. & Jaggi, M. Unsupervised scalable representation learning for multivariate time series. In Wallach, H. *et al.* (eds.) *Advances in Neural Information Processing Systems*, vol. 32 (Curran Associates, Inc., 2019).

[118] Chen, T., Kornblith, S., Norouzi, M. & Hinton, G. A simple framework for contrastive learning of visual representations. In III, H. D. & Singh, A. (eds.) *Proceedings of the 37th International Conference on Machine Learning*, vol. 119 of *Proceedings of Machine Learning Research*, 1597–1607 (PMLR, 2020).

[119] He, K., Fan, H., Wu, Y., Xie, S. & Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020).

[120] Chen, X. & He, K. Exploring simple siamese representation learning. *2021 IEEE/CVF Conf. on Comput. Vis. Pattern Recognit. (CVPR)* (2021).

[121] Grill, J.-B. *et al.* Bootstrap your own latent - a new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, vol. 33, 21271–21284 (Curran Associates, Inc., 2020).

[122] Caron, M. *et al.* Unsupervised learning of visual features by contrasting cluster assignments. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. & Lin, H. (eds.) *Advances in Neural Information Processing Systems*, vol. 33, 9912–9924 (Curran Associates, Inc., 2020).

[123] Zbontar, J., Jing, L., Misra, I., LeCun, Y. & Deny, S. Barlow twins: Self-supervised learning via redundancy reduction. In Meila, M. & Zhang, T. (eds.) *Proceedings of the 38th International Conference on Machine Learning*, vol. 139 of *Proceedings of Machine Learning Research*, 12310–12320 (PMLR, 2021).

[124] Doersch, C., Gupta, A. & Efros, A. A. Unsupervised visual representation learning by context prediction. *2015 IEEE Int. Conf. on Comput. Vis. (ICCV)* (2015).

[125] Gidaris, S., Singh, P. & Komodakis, N. Unsupervised representation learning by predicting image rotations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings* (OpenReview.net, 2018).

[126] Noroozi, M. & Favaro, P. Unsupervised learning of visual representations by solving jigsaw puzzles. *Lect. Notes Comput. Sci.* 69–84 (2016).

[127] Dosovitskiy, A., Springenberg, J. T., Riedmiller, M. & Brox, T. Discriminative unsupervised feature learning with convolutional neural networks. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. & Weinberger, K. (eds.) *Advances in Neural Information Processing Systems*, vol. 27 (Curran Associates, Inc., 2014).

[128] Everingham, M. *et al.* The pascal visual object classes challenge: A retrospective. *Int. J. Comput. Vis.* **111**, 98–136 (2015).

[129] Lin, T.-Y. *et al.* Microsoft coco: Common objects in context (2014).

[130] Robinson, J., Chuang, C.-Y., Sra, S. & Jegelka, S. Contrastive learning with hard negative samples. *Int. Conf. on Learn. Represent.* (2021).

[131] Chuang, C.-Y., Robinson, J., Lin, Y.-C., Torralba, A. & Jegelka, S. Debiased contrastive learning (2020).

[132] Wang, X., Liu, Z. & Yu, S. X. Unsupervised feature learning by cross-level instance-group discrimination. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12586–12595 (2021).

[133] He, K., Fan, H., Wu, Y., Xie, S. & Girshick, R. Momentum contrast for unsupervised visual representation learning. *2020 IEEE/CVF Conf. on Comput. Vis. Pattern Recognit. (CVPR)* (2020).

[134] van den Oord, A., Li, Y. & Vinyals, O. Representation learning with contrastive predictive coding (2018).

[135] Patacchiola, M. & Storkey, A. J. Self-supervised relational reasoning for representation learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F. & Lin, H. (eds.) *Advances in Neural Information Processing Systems*, vol. 33, 4003–4014 (Curran Associates, Inc., 2020).

[136] Fetterman, A. & Albrecht, J. Understanding self-supervised and contrastive learning with bootstrap your own latent. *https://generallyintelligent.ai* (2020).

[137] YM., A., C., R. & A., V. Self-labelling via simultaneous clustering and representation learning. In *International Conference on Learning Representations* (2020).

[138] Caron, M., Bojanowski, P., Joulin, A. & Douze, M. Deep clustering for unsupervised learning of visual features. *Lect. Notes Comput. Sci.* 139–156 (2018).

[139] Hartigan, J. A. & Wong, M. A. A k-means clustering algorithm. *JSTOR: Appl. Stat.* **28**, 100–108 (1979).

[140] He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. *2016 IEEE Conf. on Comput. Vis. Pattern Recognit. (CVPR)* (2016).

[141] Tian, Y., Chen, X. & Ganguli, S. Understanding self-supervised learning dynamics without contrastive pairs (2021).

[142] Lim, J. H. & Ye, J. C. Geometric GAN. *ArXiv* **abs/1705.02894** (2017).

[143] He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. *2016 IEEE Conf. on Comput. Vis. Pattern Recognit. (CVPR)* DOI: 10.1109/cvpr.2016.90 (2016).

[144] Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (2015).

[145] McInnes, L., Healy, J., Saul, N. & Großberger, L. UMAP: Uniform Manifold Approximation and Projection. *J. Open Source Softw.* **3** (2018).

[146] Zhu, J., Shen, Y., Zhao, D. & Zhou, B. In-domain gan inversion for real image editing. In *Proceedings of European Conference on Computer Vision (ECCV)* (2020).

[147] Abdal, R., Qin, Y. & Wonka, P. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2019).

[148] Szegedy, C. *et al.* Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–9 (2015).

[149] Ilse, M., Tomczak, J. & Welling, M. Attention-based deep multiple instance learning. In Dy, J. & Krause, A. (eds.) *Proceedings of the 35th International Conference on Machine Learning*, vol. 80 of *Proceedings of Machine Learning Research*, 2127–2136 (2018).

[150] Howard, F. M. *et al.* The impact of site-specific digital histology signatures on deep learning model accuracy and bias. *Nat. Commun.* **12**, 4423 (2021).

[151] Lou, P. *et al.* Histoml, a markup language for representation and exchange of histopathological features in pathology images. *Sci. Data* **9**, 387 (2022).

[152] AbdulJabbar, K. *et al.* Geospatial immune variability illuminates differential evolution of lung adenocarcinoma. *Nat. Medicine* **26**, 1054–1062 (2020).

[153] Gertych, A. *et al.* Convolutional neural networks can accurately distinguish four histologic growth patterns of lung adenocarcinoma in digital slides. *Sci. Reports* **9**, 1483 (2019).

[154] Lu, C. *et al.* A prognostic model for overall survival of patients with early-stage non-small cell lung cancer: a multicentre, retrospective study. *The Lancet Digit. Heal.* **2**, e594–e606 (2020).

[155] Traag, V., Waltman, L. & van Eck, N. J. From louvain to leiden: guaranteeing well-connected communities. *Sci. Reports* (2018).

[156] Wang, X., Girshick, R., Gupta, A. & He, K. Non-local neural networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7794–7803, DOI: 10.1109/CVPR.2018.00813 (2018).

[157] Quiros, A. C. *et al.* Adversarial learning of cancer tissue representations. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Springer, 2021).

[158] Blondel, V. D., Guillaume, J.-L., Lambiotte, R. & Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**, P10008 (2008).

[159] McGregor, D., Palarea-Albaladejo, J., Dall, P., Hron, K. & Chastin, S. Cox regression survival analysis with compositional covariates: Application to modelling mortality risk from 24-h physical activity patterns. *Stat. Methods Med. Res.* **29**, 1447–1465, DOI: 10.1177/0962280219864125 (2020).

[160] Martín-Fernández, J. A., Barceló-Vidal, C. & Pawlowsky-Glahn, V. Dealing with zeros and missing values in compositional data sets using nonparametric imputation. *Math. Geol.* **35**, 253–278 (2003).

[161] Benjamini, Y. & Hochberg, Y. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *J. Royal Stat. Soc. Ser. B (Methodological)* **57**, 289–300, DOI: http://dx.doi.org/10.2307/2346101 (1995).

[162] Hendry, S. *et al.* Assessing tumor-infiltrating lymphocytes in solid tumors: A practical review for pathologists and proposal for a standardized method from the international Immuno-Oncology biomarkers working group: Part 2: TILs in melanoma, gastrointestinal tract carcinomas, non-small cell lung carcinoma and mesothelioma, endometrial and ovarian carcinomas, squamous cell carcinoma of the head and neck, genitourinary carcinomas, and primary brain tumors. *Adv. Anat. Pathol.* **24**, 311–335 (2017).

[163] Wolf, F. A. *et al.* Paga: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells. *Genome Biol.* **20**, 59 (2019).

[164] Chen, C.-L. *et al.* An annotation-free whole-slide training approach to pathological classification of lung cancer types using deep learning. *Nat. Commun.* **12**, 1193 (2021).

[165] Xu, X. *et al.* A weak supervision-based framework for automatic lung cancer classification on whole slide image. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*, 1372–1375, DOI: 10.1109/EMBC44109.2020.9176620 (2020).

[166] Sun, Z. *et al.* Histologic grade is an independent prognostic factor for survival in non-small cell lung cancer: an analysis of 5018 hospital- and 712 population-based cases. *J. Thorac. Cardiovasc. Surg.* **131**, 1014–1020 (2006).

[167] Sica, G. *et al.* A grading system of lung adenocarcinomas based on histologic pattern is predictive of disease recurrence in stage i tumors. *The Am. J. Surg. Pathol.* **34** (2010).

[168] Oiwa, H. *et al.* Clinicopathological, gene expression and genetic features of stage I lung adenocarcinoma with necrosis. *Lung Cancer* **159**, 74–83 (2021).

[169] Schäfer, M. & Werner, S. Cancer as an overhealing wound: an old hypothesis revisited. *Nat. Rev. Mol. Cell Biol.* **9**, 628–638 (2008).

[170] Rybinski, B., Franco-Barraza, J. & Cukierman, E. The wound healing, chronic fibrosis, and cancer progression triad. *Physiol. Genomics* **46**, 223–244 (2014).

[171] Chai, L., Gharbi, M., Shechtman, E., Isola, P. & Zhang, R. Any-resolution training for high-resolution image synthesis. *arXiv preprint arXiv:2204.07156* (2022).

[172] Karras, T. *et al.* Training generative adversarial networks with limited data. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. & Lin, H. (eds.) *Advances in Neural Information Processing Systems*, vol. 33 (2020).

[173] Song, Y. & Ermon, S. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, vol. 32 (2019).

[174] Song, Y., Durkan, C., Murray, I. & Ermon, S. Maximum likelihood training of score-based diffusion models. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P. & Vaughan, J. W. (eds.) *Advances in Neural Information Processing Systems*, vol. 34, 1415–1428 (Curran Associates, Inc., 2021).

[175] Ho, J., Jain, A. & Abbeel, P. Denoising diffusion probabilistic models. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. & Lin, H. (eds.) *Advances in Neural Information Processing Systems*, vol. 33, 6840–6851 (Curran Associates, Inc., 2020).

[176] Rombach, R., Blattmann, A., Lorenz, D., Esser, P. & Ommer, B. High-resolution image synthesis with latent diffusion models (2021). 2112.10752.

[177] Dhariwal, P. & Nichol, A. Diffusion models beat gans on image synthesis. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P. & Vaughan, J. W. (eds.) *Advances in Neural Information Processing Systems*, vol. 34, 8780–8794 (Curran Associates, Inc., 2021).

[178] Zhang, Z., Zhao, Z. & Lin, Z. Unsupervised representation learning from pre-trained diffusion probabilistic models. In Oh, A. H., Agarwal, A., Belgrave, D. & Cho, K. (eds.) *Advances in Neural Information Processing Systems* (2022).

[179] Traub, J. Representation learning with diffusion models (2022).

[180] Geirhos, R. *et al.* Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations* (2019).

[181] Raghu, M., Unterthiner, T., Kornblith, S., Zhang, C. & Dosovitskiy, A. Do vision transformers see like convolutional neural networks? (2021).

[182] Williams, C. G., Lee, H. J., Asatsuma, T., Vento-Tormo, R. & Haque, A. An introduction to spatial transcriptomics for biomedical research. *Genome Medicine* **14**, 68 (2022).

[183] Rao, A., Barkley, D., França, G. S. & Yanai, I. Exploring tissue architecture using spatial transcriptomics. *Nature* **596**, 211–220 (2021).

[184] Hunter, M. V., Moncada, R., Weiss, J. M., Yanai, I. & White, R. M. Spatially resolved transcriptomics reveals the architecture of the tumor-microenvironment interface. *Nat. Commun.* **12**, 6278 (2021).

Appendices

A PathologyGAN: Learning deep representations of cancer tissue

A.1 Hinge vs Relativistic Average Discriminator

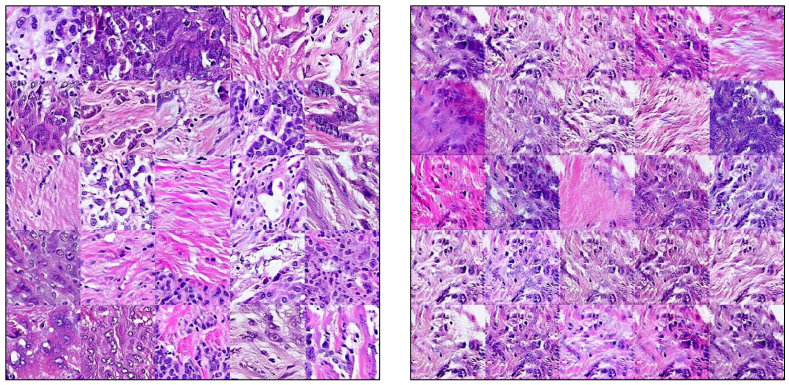


Figure A.1: Images from a Relativistic Average Discriminator loss model (A) vs images from a Hinge loss model (B). We can see how the Relativistic Average model is able to reproduce the diverse features of the cancer tissue while the Hinge loss model is prone to collapse where images in the batch share similar features without diversity between samples.

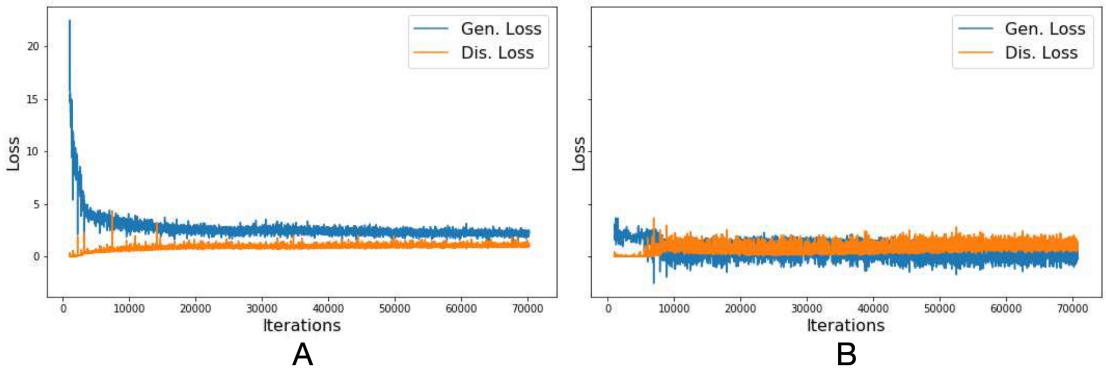


Figure A.2: Generator and discriminator loss functions of the Relativistic Average Discriminator model (A) and Hinge loss model (B). These loss functions correspond to the generated tissue samples in Figure A.1, both of them converge but only the Relativistic Average Discriminator model produces meaningful images.

A.2 Nearest neighbors

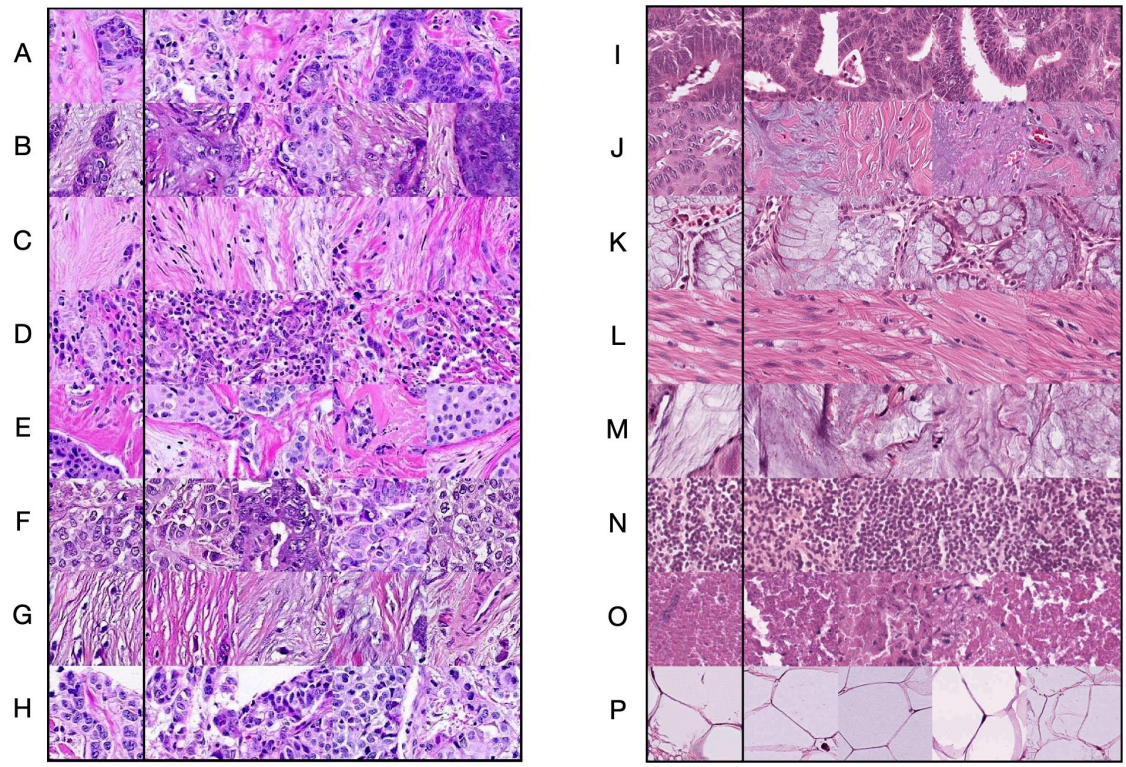


Figure A.3: Additional examples: Nearest neighbors in Inception-V1 feature space for breast cancer (A-H) and colorectal cancer (I-P). For each row, the first column image corresponds to a generated tissue samples from PathologyGAN, the remaining columns are the closest real images in Inception-V1 feature space.

A.3 Vector operations

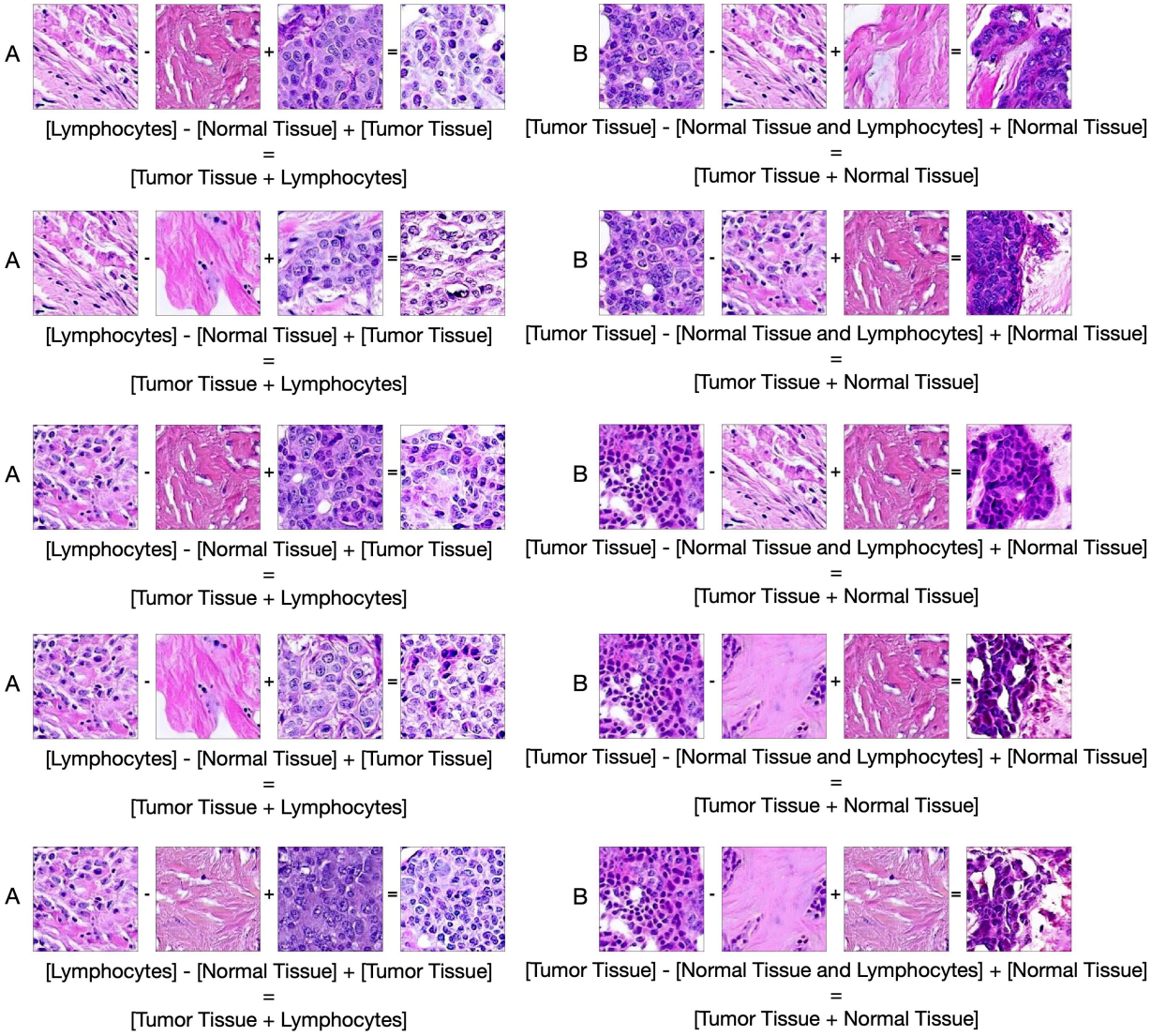


Figure A.4: Additional examples: Breast cancer vector operations with different images, all operations correspond to: (A) Lymphocytes - Normal Tissue + Tumor Tissue = Tumor Tissue and Lymphocytes. (B) Tumor Tissue and Lymphocytes - Lymphocytes + Normal Tissue = Tumor and Normal Tissue.

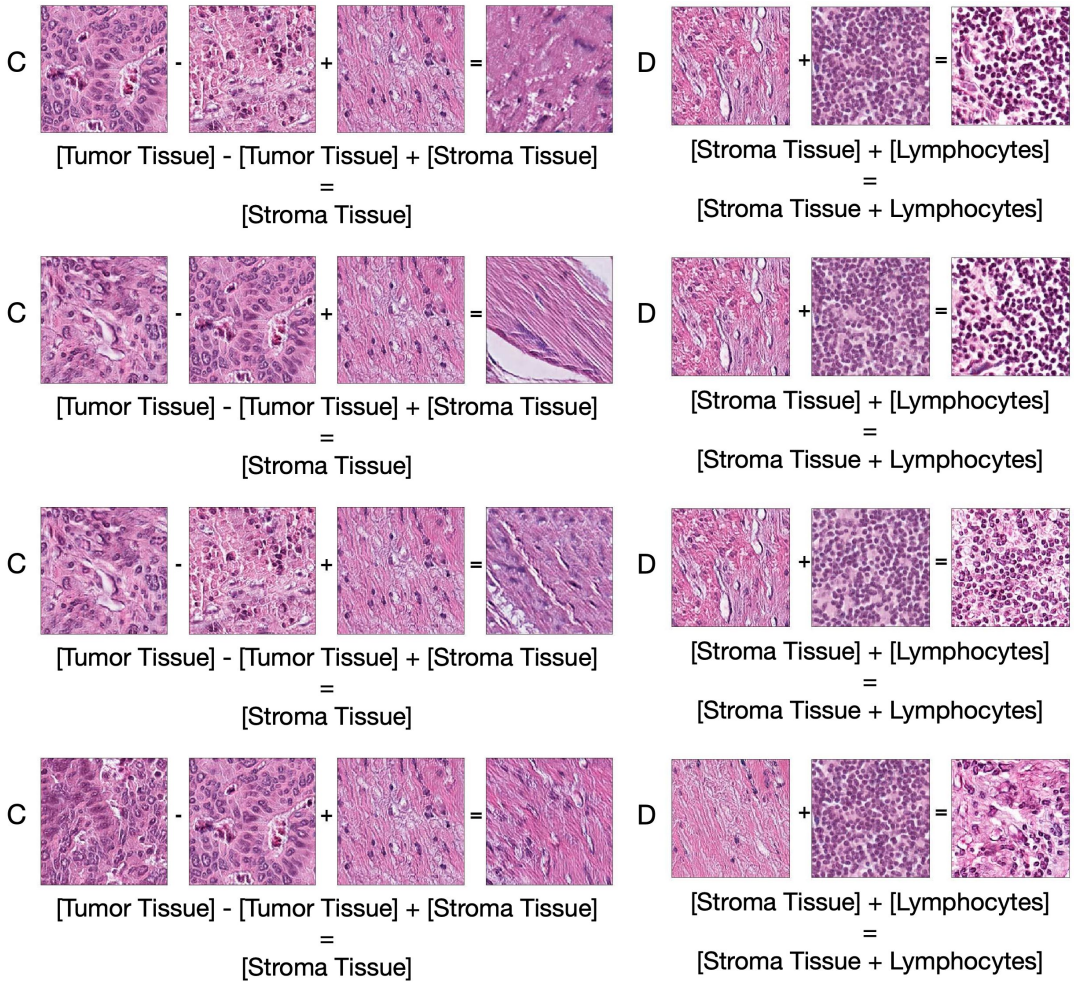


Figure A.5: Additional examples: Colorectal cancer vector operations with different images, all operations correspond to: (C) Tumor Tissue - Tumor Tissue + Stroma Tissue = Stroma Tissue. (D) Stroma Tissue + Lymphocytes = Stroma Tissue and Lymphocytes.

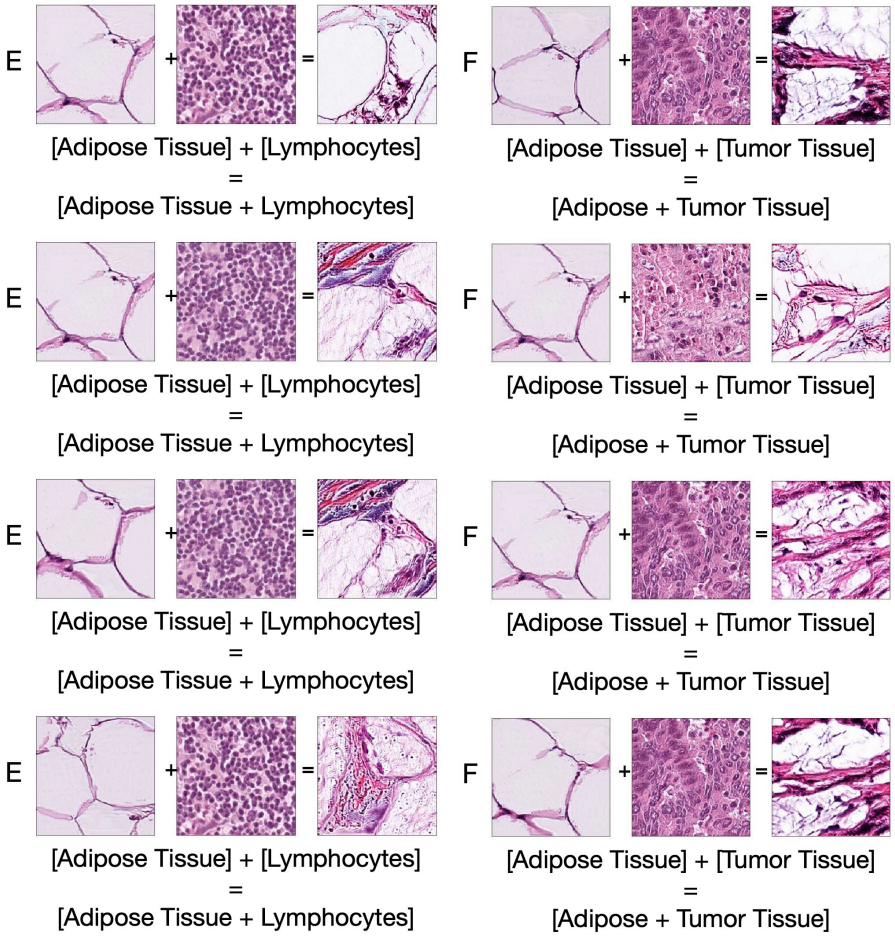


Figure A.6: Additional examples: Colorectal cancer vector operations with different images, all operations correspond to: (E) Adipose Tissue + Lymphocytes = Adipose Tissue and Lymphocytes. (E) Adipose Tissue + Tumor Tissue = Adipose and Tumor Tissue.

A.4 Visualization of linear interpolations and vector operations in the latent space.

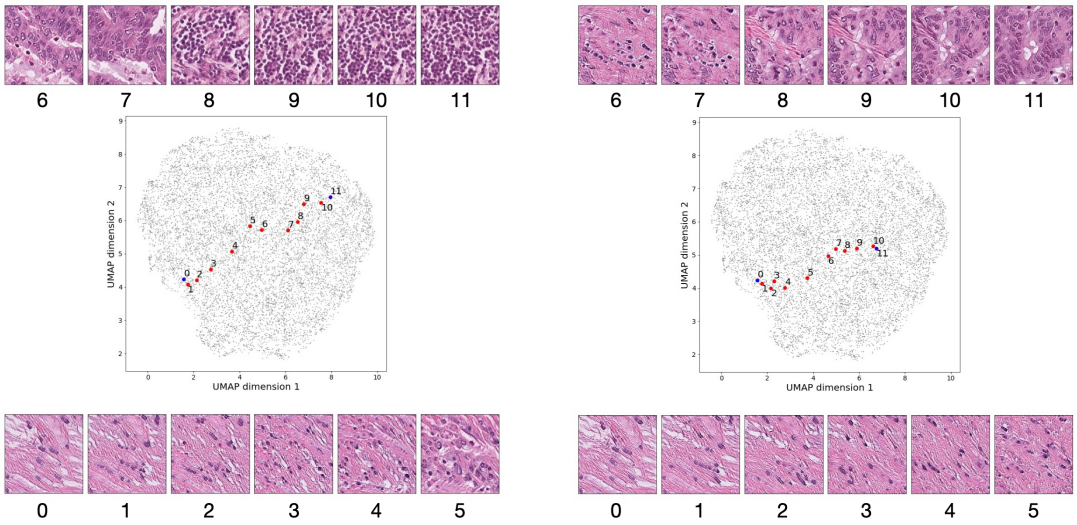


Figure A.7: Additional examples: Uniform Manifold Approximation and Projection (UMAP) of linear interpolations in PathologyGAN's latent space w and associated images in colorectal cancer tissue. We present transitions from stroma to tumor, stroma to lymphocytes, and tumor to lymphocytes. Starting vectors are colored in blue while intermediate points of the interpolations are colored in red. Through the intermediate vectors, we show that gradual transitions in the latent space translate into smooth feature transformations, increase/decrease of tumor cells, or increase of lymphocyte counts.

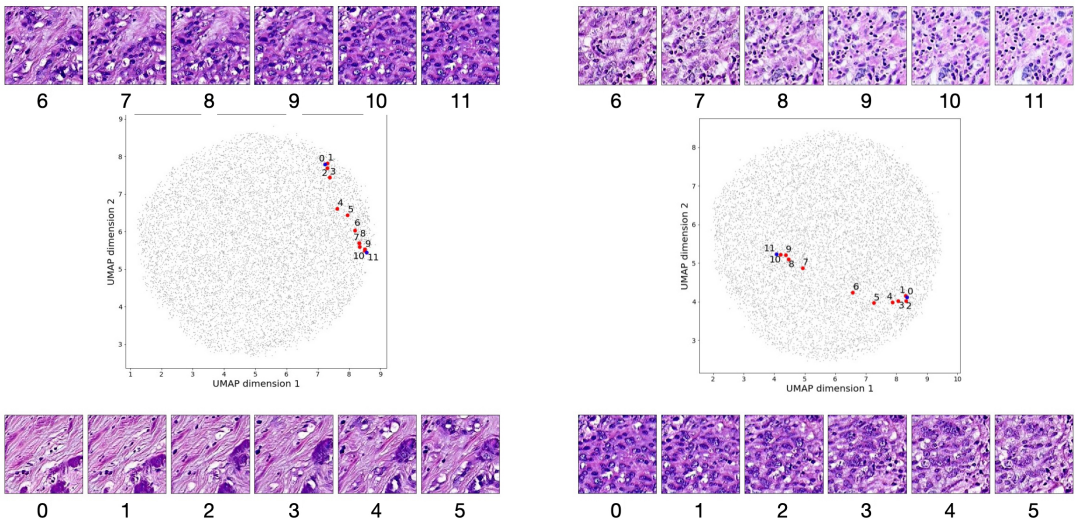


Figure A.8: Additional examples: Uniform Manifold Approximation and Projection (UMAP) of linear interpolations in PathologyGAN's latent space w and associated images in breast cancer tissue. We present transitions from stroma to tumor, stroma to lymphocytes, and tumor to lymphocytes. Starting vectors are colored as in blue while intermediate points of the interpolations are colored as in red. Through the intermediate vectors we show that gradual transitions in the latent space translate into smooth feature transformations, increase/decrease of tumor cells, or increase of lymphocyte counts.

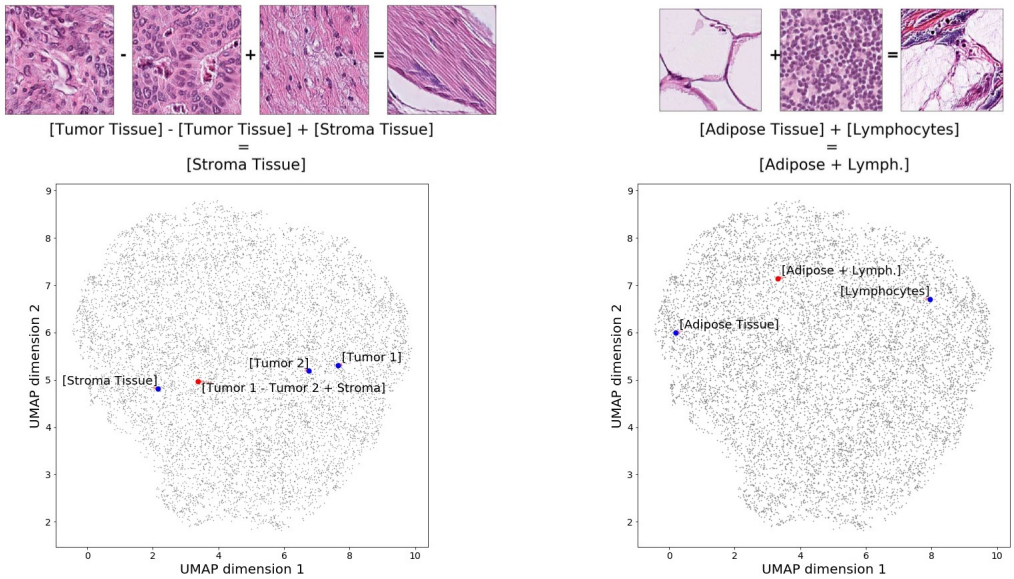


Figure A.9: Additional examples: Uniform Manifold Approximation and Projection (UMAP) of linear vector operations in PathologyGAN's latent space w and associated images in colorectal cancer tissue. Original vectors are colored in blue while the results are colored in red. After vector operations, the results fall into regions of the latent space that correspond to the tissue type.

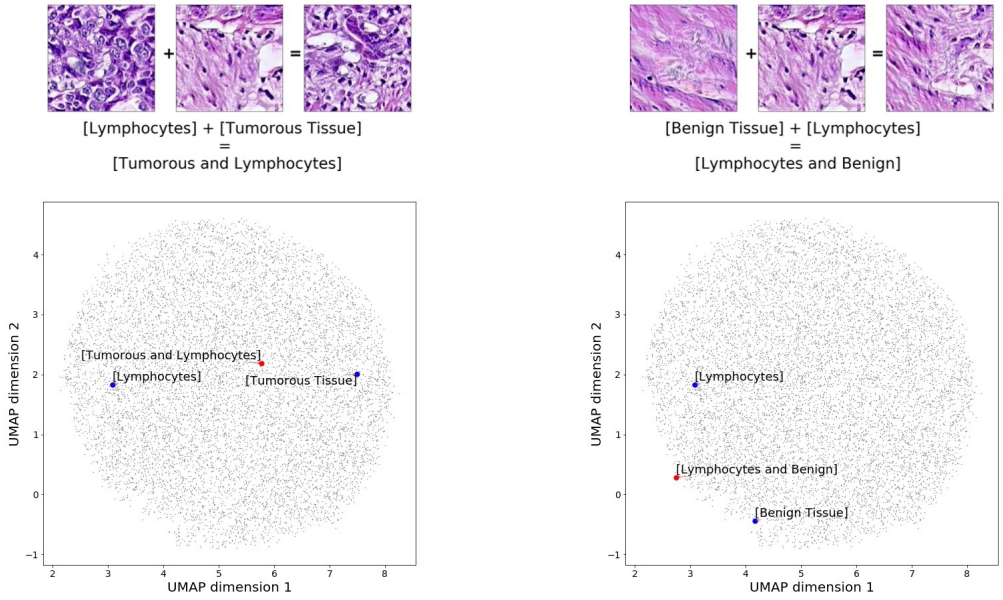


Figure A.10: Additional examples: Uniform Manifold Approximation and Projection (UMAP) of linear vector operations in PathologyGAN's latent space w and associated images in breast cancer tissue. Original vectors are colored in blue while the results are colored in red. After vector operations, the results fall into regions of the latent space that correspond to the resulting features. E.g. tumor, lymphocytes, or stroma cells.

B Adversarial learning of cancer tissue representations

B.1 Tissue Reconstructions

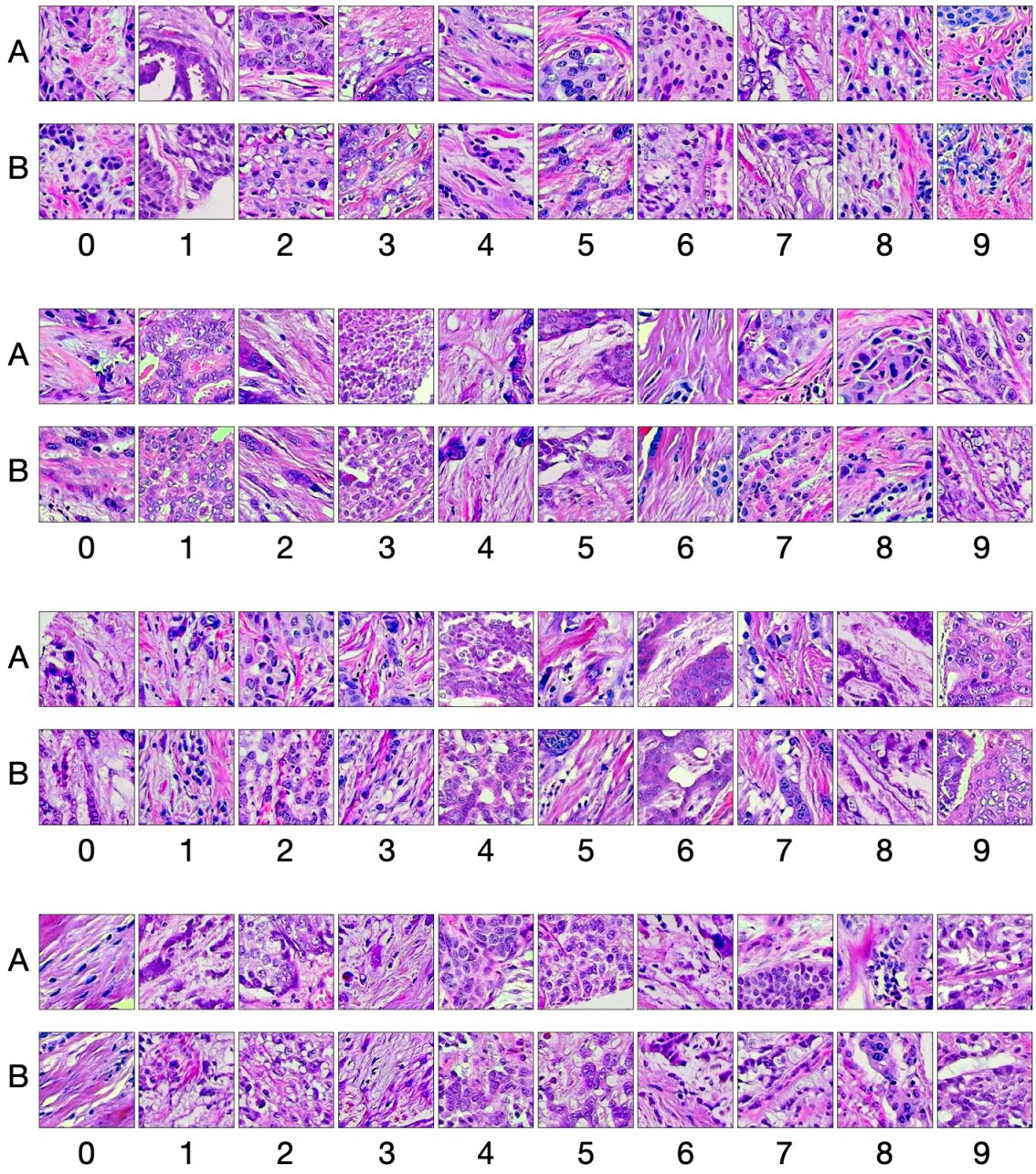


Figure B.1: Additional examples: Real tissue images and their reconstructions on breast cancer tissue. We take real tissue images and map them to the latent space with our encoder, then we use the generator with the latent vector representations to generate the image reconstructions. (A) correspond to the real tissue images x_{real} and (B) to the reconstructions $x_{recon} = G(E(x_{real}))$, the images are paired in columns. We present samples of breast cancer tissue from the VGH and NKI cohorts.

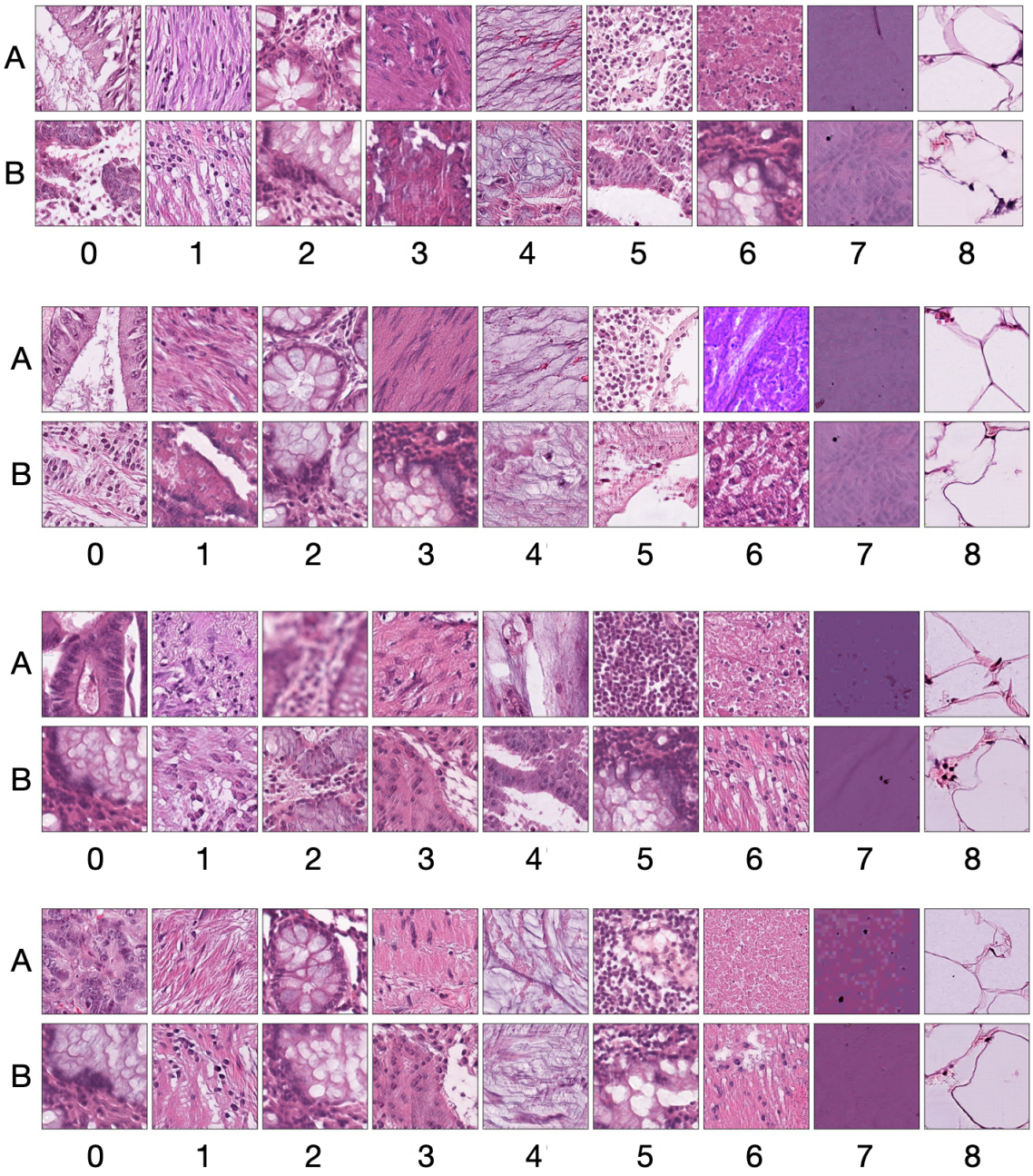


Figure B.2: Additional examples: Real tissue images and their reconstructions on breast colorectal tissue. We take real tissue images and map them to the latent space with our encoder, then we use the generator with the latent vector representations to generate the image reconstructions. (A) correspond to the real tissue images x_{real} and (B) to the reconstructions $x_{recon} = G(E(x_{real}))$, the images are paired in columns. We present colorectal cancer tissue samples from the NCT cohort where each tissue image is labeled by the tissue type: (0) colorectal adenocarcinoma epithelium (tumor), (1) cancer-associated stroma, (2) normal colon mucosa, (3) smooth muscle, (4) mucus, (5) lymphocytes, (6) debris, (7) background, and (8) adipose.

B.2 Receiver operating characteristic curve and confusion matrix of a logistic regression for tissue type classification

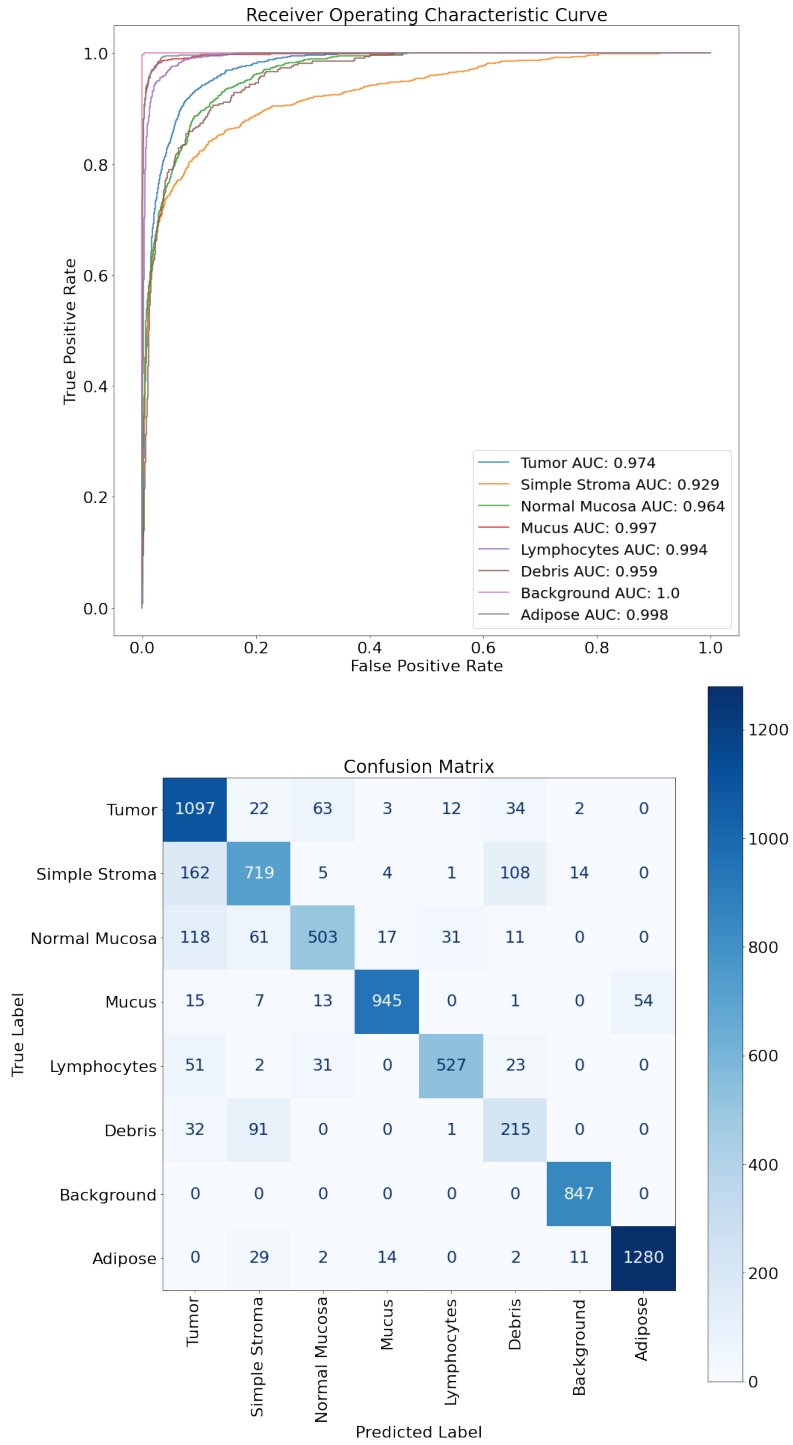


Figure B.3: Receiver Operating Characteristic (ROC) curve and confusion matrix of a logistic regression for tissue type classification on PathologyGAN tissue representations. We use the colorectal cancer tissue images from NCT and train a logistic regression (one- vs-rest) over tissue representations of the training set, evaluating performance on test set representations. We show how a logistic regression can reliably find information in representations to predict tissue types.

C Histomorphological phenotype learning: a self-supervised methodology for discovery of morphological tissue patterns

C.1 Methods: Evaluation

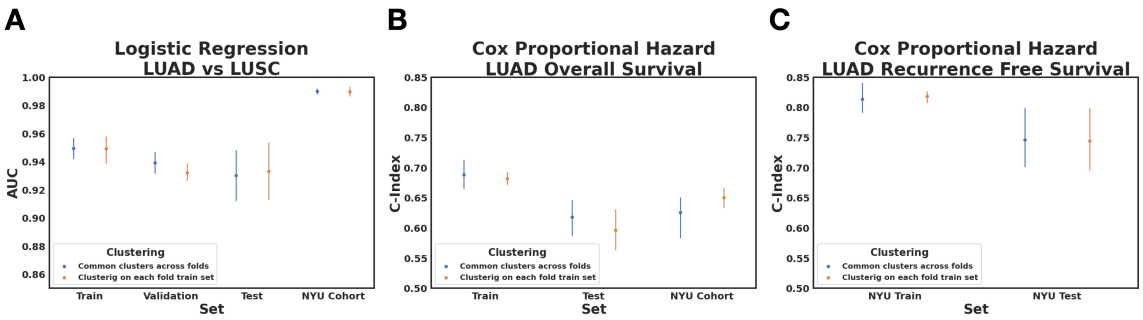


Figure C.1: Clustering impact on performance. Performance variation when clustering is done on each fold's train set compared to when a particular cluster configuration is hold the same across folds for the classification or survival task. **A** Performance variation on logistic regression for LUSC versus LUAD classification. **B** Performance variation on Cox proportional hazard for LUAD overall survival. **C** Performance variation on Cox proportional hazard for LUAD recurrence free survival.

C.2 Results: Pan-cancer classification

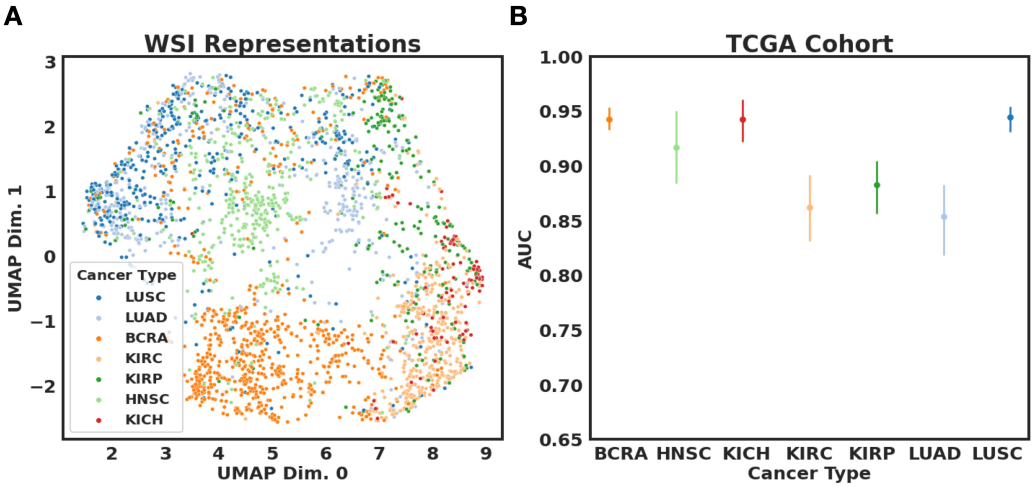


Figure C.2: Pan-Cancer classification performance and representation analysis. A Uniform Manifold Approximation and Projection (UMAP)[145] dimensionality reduction of whole slide image (WSI) vector representations for BCRA, HNSC, KICH, KIRC, KIRP, LUAD, and LUSC cancer type from TCGA. **B** Area under curve (mean and 95% confidence interval) over a 5-fold cross-validation on cancer types from TCGA using a logistic regression (one-vs-rest) over WSI vector representations. These figures show the general applicability of HPL on different cancer types.

C.3 Results: LUAD versus LUSC classification

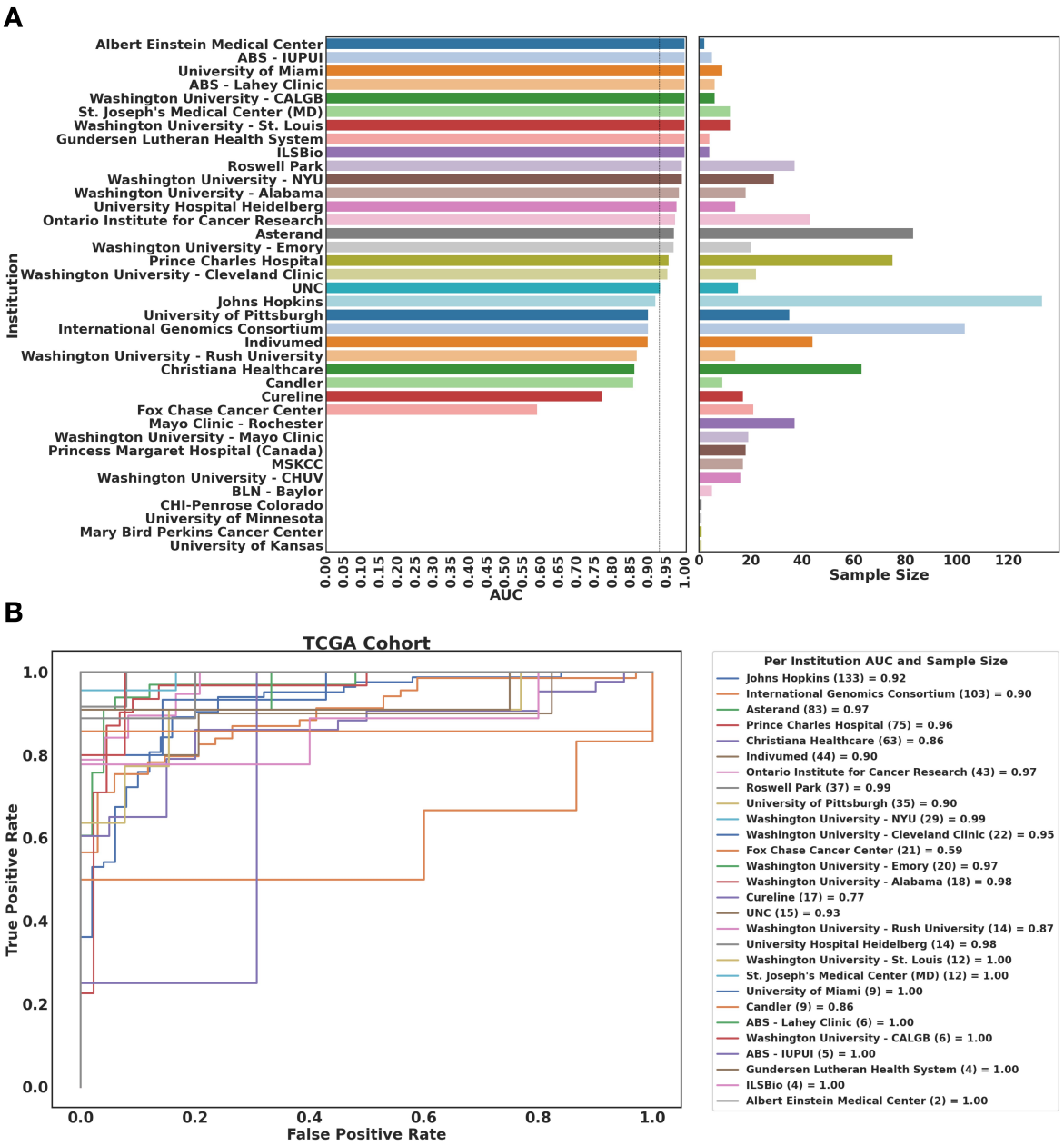


Figure C.3: Lung adenocarcinoma (LUAD) versus squamous cell carcinoma (LUSC) classification performance per institution. **A** Receiver operating characteristic (ROC) area under curve (AUC) per TCGA institution and WSI sample size. Institutions without an AUC value only have one kind of lung type, not allowing to compute the ROC AUC. **B** ROC curve per TCGA institution. These results correspond to the 5-fold cross-validation on TCGA dataset and NYU dataset as independent cohort as included in **Figure 5.4 A**.

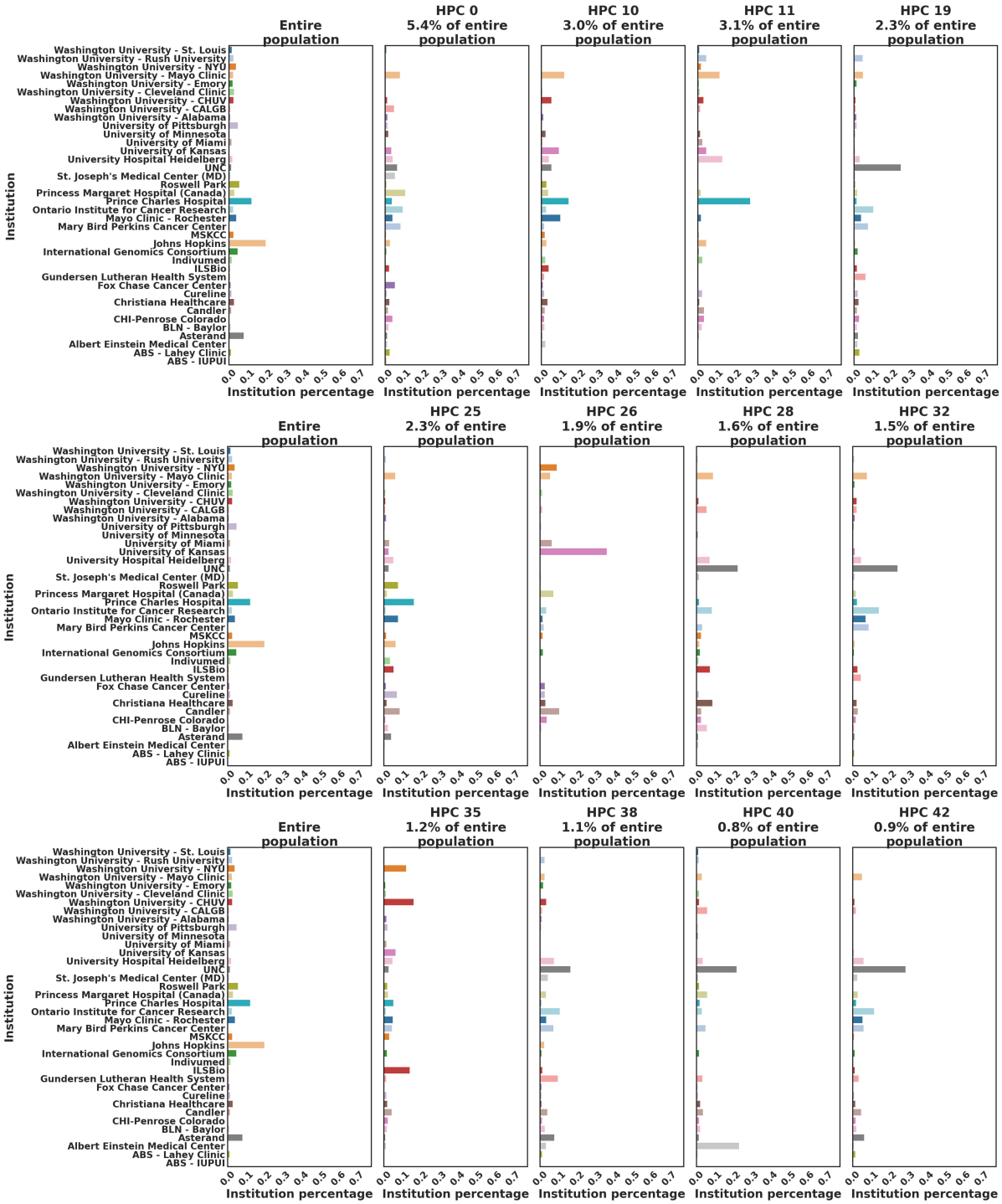


Figure C.4: Lung adenocarcinoma (LUAD) versus squamous cell carcinoma (LUSC) HPCs. Institution distribution per HPC and for the entire population of tissue tiles, we show 12 randomly sampled HPCs.

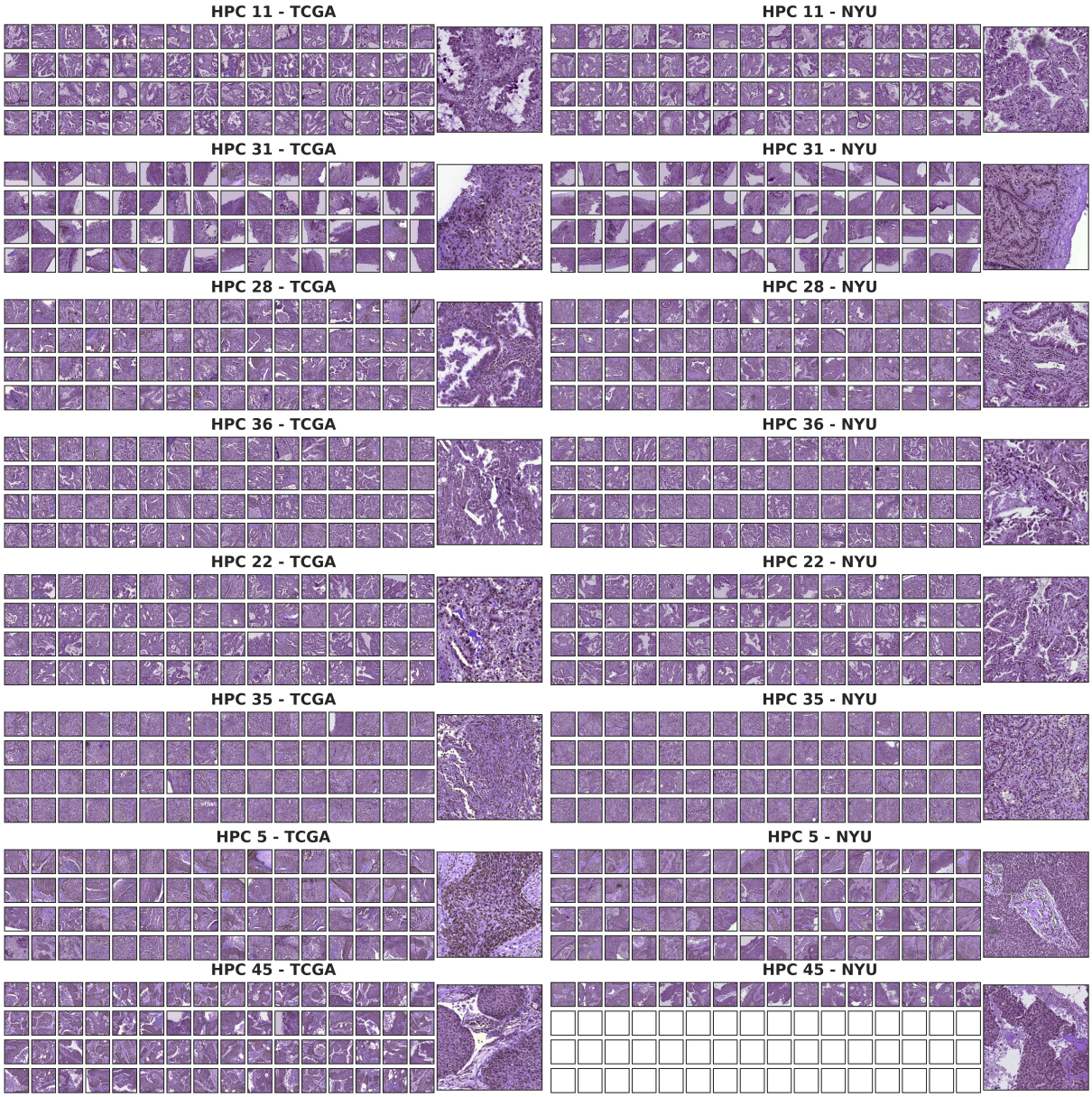


Figure C.5: The Cancer Genome Atlas (TCGA) and New York University (NYU) NYU_1 tile samples for statistically significant HPCs on lung adenocarcinoma (LUAD) versus cell squamous cell carcinoma (LUSC) classification.

C.4 Results: LUAD survival

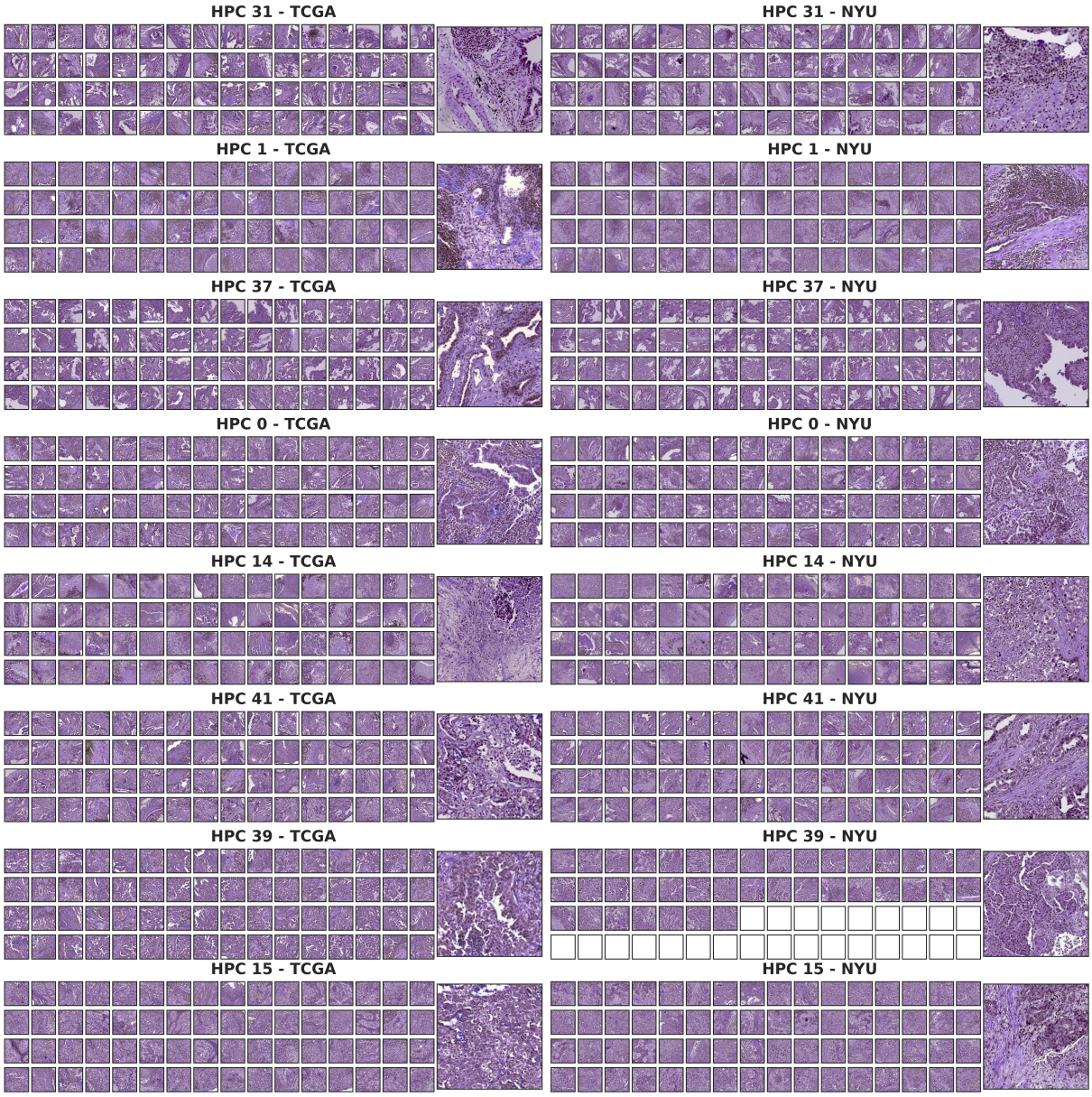


Figure C.6: The Cancer Genome Atlas (TCGA) and New York University (NYU) NYU_2 tile samples for statistically significant HPCs on lung adenocarcinoma (LUAD) overall survival.

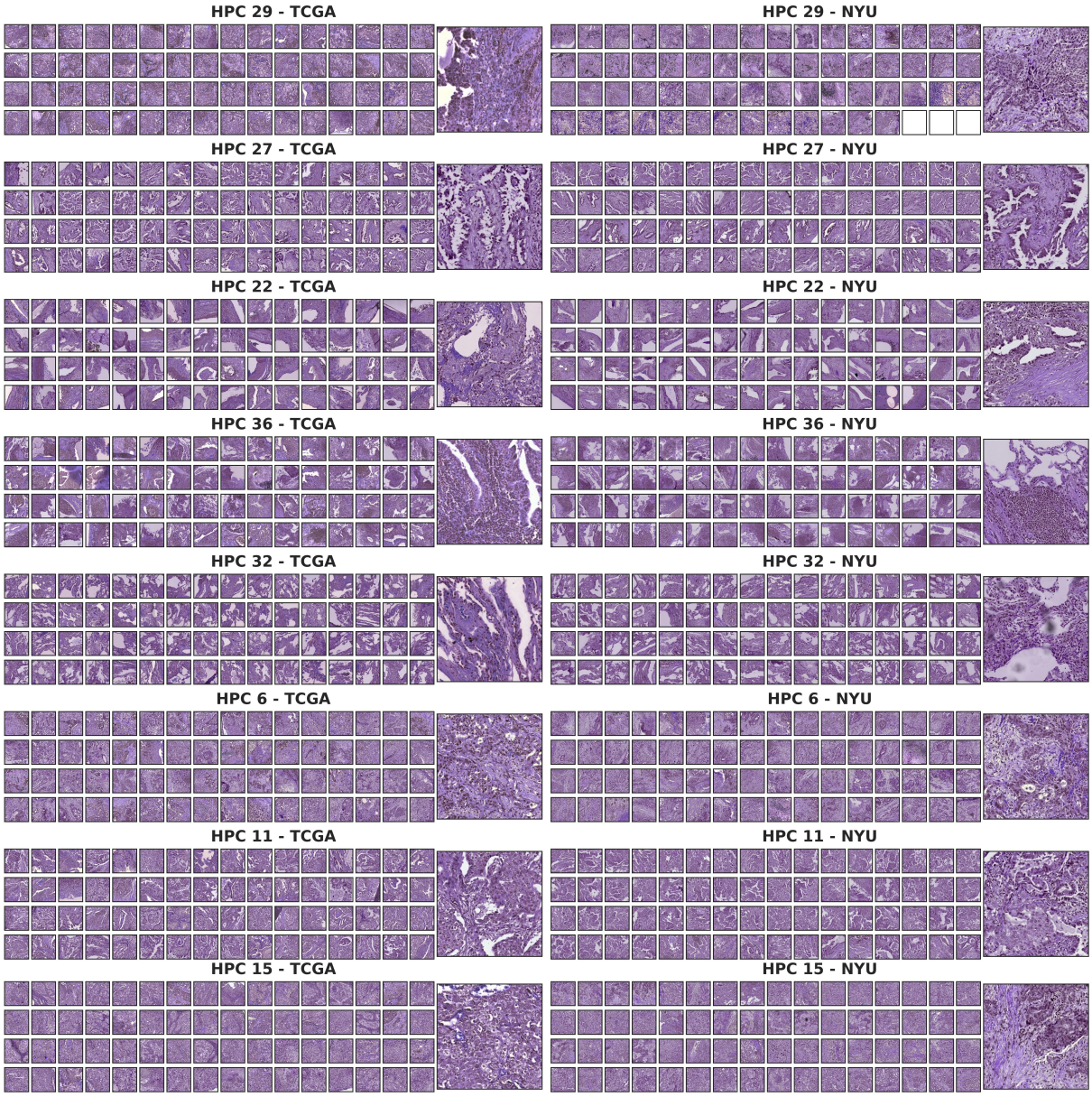


Figure C.7: The Cancer Genome Atlas (TCGA) and New York University (NYU) NYU_2 tile samples for statistically significant HPCs on lung adenocarcinoma (LUAD) recurrence free survival.

HPC	Description	Histological Predominant Pattern	Hot/Warm/Cold Lymphocytic Infiltration	Morphological Supergroup	Immune Signatures
37	Ragged complex appearances	In situ with thickened interstitium, papillary	Warm	Luminal profiles	Low proliferation
26	Stroma-rich, many lumina	Acinar/in situ	Cold	Acinar and stroma-rich	Low proliferation
27	Micropapillary	Micropapillary	Warm	Non-solid cold	Nil
28	Patent spaces	Papillary/in situ	Warm	Non-solid cold	Low proliferation, high Th17
18	Open spaces, mucinous cytology	Acinar/complex	Warm	Non-solid cold	High Th17
24	Gaping lumina with mucin	Micropapillary/in situ	Cold	Non-solid cold	High Th17
19	Mucinous and bronchiolar cytology	In situ	Warm	Non-solid cold	Low proliferation, high Th17
8	Multiple small branched lumina	Acinar/papillary/in situ	Hot	Luminal profiles	Low proliferation
30	Micropapillary, often with mucin. Also retraction artefact around small solid nests.	Micropapillary and solid	Warm	Non-solid cold	Nil
41	Small malignant nests, abundant stroma, patchy moderate inflammation	Crib/solid/acinar	Warm	Acinar and stroma-rich	Nil
35	Tufted and micropapillary	Micropapillary/in situ	Warm	Non-solid cold	Low proliferation
39	Solid/crib/micropapillary. Patchily inflamed stroma	Solid/crib/Micropapillary	Warm	Solid and/or hot	Nil
4	Normal lung with intraluminal debris, small vessels	Non-malignant lung	Cold	Non-malignant/in situ	Nil
12	Lambertosis/bronchiolisation and inflammation	Non-malignant lung	Hot	Non-malignant/in situ	Nil
20	Honeycomb' appearance- cartilage/fat/oedema	-	Warm	Special tissues	Nil
42	Cartilage	-	Cold	Special tissues	Nil
23	pure stroma with elastin	-	Cold	Acinar and stroma-rich	Nil
3	Elastotic/collagenous stroma	-	Cold	Acinar and stroma-rich	Nil
38	Solid pattern, clear cell, inflamed stroma.	Solid	Hot	Solid and/or hot	Nil
44	Linear dark folds. Patchy inflammation. Few spaces	Solid	Hot	Solid and/or hot	Nil
45	Complex glandular. Little stroma.	Cribriform	Warm	Non-solid cold	Nil
16	Inflamed collapsed stroma with minimal epithelial elements	Acinar	Hot	Acinar and stroma-rich	Low proliferation
40	Bronchial epithelium, often inflamed	-	Hot	Special tissues	Low proliferation
32	Rounded lumina	Non-malignant lung	Cold	Non-malignant/in situ	NIL
2	Haemorrhage, vessels and airways	Non-malignant lung	Cold	Non-malignant/in situ	Low proliferation
22	Large vessels	-	Cold	Special tissues	Low proliferation
31	Normal/in situ, enriched for small vessels/airways, patchy inflammation	Non-malignant/in situ	Warm	Non-malignant/in situ	Lym infiltration, TCR diversity, stroma high
13	Normal/in situ, some inflammation	Non-malignant/in situ. Usually cold.	Warm	Non-malignant/in situ	Lym infiltration, TCR diversity, stroma high
0	Inflamm with tumour destruction	Non-solid	Hot	Solid and/or hot	Lym infiltration, TCR diversity, stroma high
36	Normal/reactive, dense inflammation	Normal/reactive	Hot	Solid and/or hot	Lym infiltration, TCR diversity, stroma high
43	Sheets of inflammation	-	Hot	Solid and/or hot	Lym infiltration, TCR diversity, stroma high
9	Inflamed stroma with sparse malignant epithelium	Multiple	Hot	Solid and/or hot	Lym infiltration, TCR diversity, stroma high
1	Sheets of inflammation with destruction	Multiple	Hot	Solid and/or hot	Lym infiltration, TCR diversity, stroma high
17	Half empty tiles	Solid	Hot	Special tissues	High prolif, high mts, CD4>CD8
25	High-grade solid and cribriform	Solid	Hot	Solid and/or hot	High prolif, high mts, CD4>CD8
5	Solid. Big pleomorphic nuclei. Inflamed stroma.	Solid	Hot	Solid and/or hot	High prolif, high mts, CD4>CD8
34	Inflamed collagenous stroma and acini	Acinar	Hot	Acinar and stroma-rich	Low signal
29	Solid. Some discohesion with small spaces. inflammation.	Solid	Hot	Solid and/or hot	Low signal
33	Narrow clefts- solid with retraction	Solid	Hot	Solid and/or hot	High proliferation
6	Solid and complex glandular	Crib/Solid	Warm	Acinar and stroma-rich	High proliferation
14	Necrosis	-	Warm	Non-malignant/in situ	High proliferation
10	Haemorrhagic lung	Non-malignant lung	Cold	Non-malignant/in situ	Low signal
11	Linear clefts, often retraction artefact	Solid/crib/papillary/micropapillary	Warm	Luminal profiles	High proliferation
15	Solid, high-grade, necrosis. Patchily inflamed stroma	Solid	Warm	Solid and/or hot	High proliferation
7	Stroma-rich. Stippled textures with high contrast.	Mixed	Warm	Solid and/or hot	Nil
21	Multiple narrow lumina	Non-solid with open lumina	Warm	Luminal profiles	Nil

Figure C.8: Histological assessment of lung adenocarcinoma (LUAD) overall and recurrence free survival HPCs. Each HPC includes an annotation of features found across its tiles: (1) brief description, (2) LUAD growth patterns, (3) hot, warm, or cold lymphocytic infiltration[162], (4) morphological supergroup of cluster features, and (5) summary of immune signature correlations[66]. HPCs are ordered by correlations between HPCs and immune signature as in **Figure 5.8**. Descriptions, LUAD growth patterns, hot, warm, or cold lymphocytic infiltration, and morphological supergroup of cluster features annotations were done by a subspecialty histopathologist (Prof. John Le Quesne).