Integration of Physical Prior Knowledge in Machine Learning Imaging Workflows

---

Marco Aversa

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

Contents

List of Figures

List of Tables

Abstract

In this thesis, we introduce several scientific machine learning methods to circumvent models' data dependencies by bridging the gap between the physical insight about the acquisition data process and the machine learning paradigms. The aim is to open neural networks' black box by combining it with a well-known forward process, the white-box model. Having access to a well-defined white-box model, we can embed its information inside the network in order to obtain a hybrid model where we partially know how it should respond.

Focusing on medical and aerospace imaging applications, we leverage sensor calibration profile and image signal processing prior knowledge to develop three novel validation protocols via a physically faithful differentiable model. Starting from the object, through the optics, to the sensor, the entire imaging process is integrated into the machine learning workflow to detect model failures and enhance model robustness. These novel methods extend model generalization beyond classical techniques like catalogue testing or augmentation, bringing additional freedom to the data and model explainability.

Guided by the principle of metrologically precise data handling, we designed a data-centric machine learning workflow to emulate expensive satellite imaging payloads using more affordable drone image data. The emulation mimics pixel distribution and the optical properties of the target acquisition system, allowing an *in silico* model validation before launching the physical prototype. The experiments demonstrate the lowest resolution and signal-to-noise ratio necessary for conducting a segmentation task on satellite data, offering the optimal range of optical parameters where the model operates effectively.

While in medical imaging the acquisition process has a key role in making the model more resilient to real-world application, it does not cover the out-of-distribution negative impact on the downstream model due to missing data in sparsely annotated datasets. In this thesis, we developed a generative framework based on diffusion models for the synthesis of lung cancer tissue histological data. The model leverages the biological insight on the macroscopic cell arrangement to guide the synthesis using new features from unlabelled data. We evaluated the efficacy and fidelity of the generated content via a comprehensive data assessment and we explored the potential of synthetic data for training on in-out house data.

Chapter 1.

Introduction

Neural networks already have a huge impact on our world. The exponential growth of data and computational resources allowed big companies to train and distribute large trained models to many users. In the palm of a hand, a common user can acquire an image and detect every object in it [101], generate high-resolution images or videos from text [182, 176, 192, 1], organise and speed-up the work with large-language models. These models rely on large sets of annotated data to cover every possible scenario, reducing the chance of deploying a robust model on unseen inputs. Due to this data dependence, they are mainly applied to limited low-risk scenarios. However, in most critical and sensitive applications, collecting data is expensive and time-consuming. Indeed, producing new labels requires highly technical specialists interacting in a closed loop with machine learning experts to generate an ideal dataset for training. Since the data do not completely cover the input domain, deploying a data-driven model is unreliable in a real-world scenario. In the absence of data, it is possible to incorporate domain-specific knowledge of the mathematical and physical properties of the system into the model architecture or learning process. Integrating the system forward model into the machine learning model can lead to significant advancements. This approach results in more accurate and interpretable models that better reflect the underlying physical processes. It also reduces the need for large amounts of training data, consequently making it easier to train models on small or limited datasets. Additionally, this integration leads to robust models, less likely to overfit or fail in situations where the system's dynamics change.

In Chapter 2, I provided an introduction to the foundational concepts for the following chapters. This included a focus on key aspects of scientific machine learning algorithms, focusing on differentiable and probabilistic models.

In Chapter 3, I explored the end-to-end imaging acquisition process, from the source to the post-processed image. While data acquisition is rarely considered in the machine learning workflow, in my work, I investigated how each element in the acquisition pipeline combined with

the machine learning model can enhance the model's resilience to different types of acquisition systems. Additionally, developing models that can accurately predict and interpret the behaviour of sensor and optics systems reduces the need for expensive prototype testing. The application of these models has significant potential in fields such as remote sensing, autonomous driving, aerospace imaging and medical imaging, where accurate and reliable sensing systems are essential for real-world applications.

In Chapter 4, I designed a physical emulation of a satellite imaging payload starting from raw drone images. The synthetic data is used to establish tolerance boundaries in the parameter space, validating model performance before prototyping the actual physical setup. The emulation has been validated consistently with respect to the target sensor and optical calibration profile.

In Chapter 5, I designed and developed a generative framework which allows the generation of arbitrarily large images with biological plausibility given a model trained on spatially localised features. The synthetic data has been assessed through a survey with a team of domain experts and quantitatively with fidelity and diversity metrics.

1.1 Thesis contributions

The material presented in chapters Chapter 3, 4 and 5 is shared with the machine learning community through these works:

- Chapter 3
  - Oala L.*, **Aversa M.***, Nobis G., Willis K., Neuenschwander Y., Buck M., Matek C., Extermann J., Pomarico E., Samek W., Murray-Smith R., Clausen C., Sanguinetti B. *Data Models for Dataset Drift Controls in Machine Learning With Optical Images*. TMLR, 2023.
  - Oala L.*, **Aversa M.***, Nobis G., Willis K., Neuenschwander Y., Buck M., Matek C., Extermann J., Pomarico E., Samek W., Murray-Smith R., Clausen C., Sanguinetti B. Data Models for Dataset Drift Controls in Machine Learning With Optical Images. ICML, Spurious Correlations, Invariance and Stability workshop, 2023.
  - Oala L.*, **Aversa M.***, Nobis G., Willis K., Neuenschwander Y., Buck M., Matek C., Extermann J., Pomarico E., Samek W., Murray-Smith R., Clausen C., Sanguinetti B. *Data Models for Dataset Drift Controls in Machine Learning With Optical Images*. ICML, Differentiable Almost Everything: Differentiable Relaxations, Algorithms, Operators, and Simulators Workshop, 2023.

* equally contributed

Contributions: Co-led the paper with Luis Oala who brought the team together, and managed the structure of the work. I led and conducted the machine learning experiments in the project, and played a key role in the conception, design, development and writing of the paper.

- Chapter 4

  – **Aversa M.**, Malik Z., Geier P., Droz F., Upegui A., Murray-Smith R., Clausen C., Sanguinetti B. *Data-centric AI workflow based on compressed raw images.* 8th International Workshop on OnBoard Payload Data Compression, 2022.

  – **Aversa M.**, Oala L., Clausen C., Murray-Smith R., Sanguinetti B. *Physical Data Model in Machine Learning Imaging Pipelines*. NeurIPS, Machine Learning and the Physical Science Workshop, 2022.

  Selected as *contributed talk*.

Contributions: Led the project; involved in the design of the emulation process; ran all the machine learning experiments.

- Chapter 5

  – **Aversa M.**, Nobis G., Hägele M., Standvoss K., Chirica M., Murray-Smith R., Alaa A., Ruff L., Ivanova D., Samek W., Klauschen F., Sanguinetti B., Oala L. *DiffInfinite: Large Mask-Image Synthesis via Parallel Random Patch Diffusion in Histopathology*. NeurIPS, Dataset and Benchmark track, 2023

  Selected as *spotlight*.

Contributions: Led the project; designed and developed the generative framework; generated the synthetic data; involved in the data quality and diversity validation.

- Other scientific contributions during the Ph.D. period which do not appear in this thesis.

  – Mitton J., Mekhail S., Padgett M., Faccio D., **Aversa M.**, Murray-Smith R. *Bessel Equivariant Networks for Inversion of Transmission Effects in Multi-Mode Optical Fibres*. NeurIPS, 2022.

Contributions: Assisting with the formulation of the mathematical framework along with its associated physical interpretation.

  – Nobis G., **Aversa M.**, Springenberg M., Detzel M., Ermon S., Nakajima S., Murray-Smith R., Lapuschkin S., Knochenhauer C., Oala L., Samek W., *Generative Fractional Diffusion Models*. NeurIPS Workshop on Diffusion Models, 2023.

Contributions: Supervised the project; contributing to the foundational ideas of the project.

Chapter 2.

Background and Related Works

2.1 Scientific machine learning

Scientific machine learning is a broad research area which covers a wide spectrum of methods to reach the same goal. It is defined across research communities with different names, like 'Simulation intelligence' or 'Physics-informed/infused machine learning' but it always converges to the same common denominator, guiding black-box machine learning models with a priori information on the investigated system. These kinds of methods involve improving machine learning robustness with respect to physical perturbations or they can be used for scientific discoveries, by filling missing or unknown data. There are several strategies to embed physical information in the machine learning workflow. This integration leads to a hybrid model, a combination of the physical process (white-box) and neural networks (black-box). Working with a grey-box can lead to several benefits (see Fig. 2.1). With the classical information of the system, we have partial control over the model and we can understand better the way it learns. Integrating prior information with data-driven learning can reduce the computational complexity and the amount of data needed to train. Indeed, this a priori insight can steer or constrain the learning process to a real-world, more robust, consistent solution. With the recent wave of interest in interpretability, researchers have developed methods for building these knowledge-driven neural networks through two kinds of biases (see Fig. 2.2): inductive bias and learning bias [116]. The inductive bias aims to embed the prior insight in the model architecture. The neural network is built consistently with the physical law that describes the data. The white-box physical model can be integrated into the neural network or can precede it. The learning bias instead constrains the model to converge to a subset of solutions, either with a specific loss function or through a multi-fidelity approach where the data is fed with simulation-based ones.

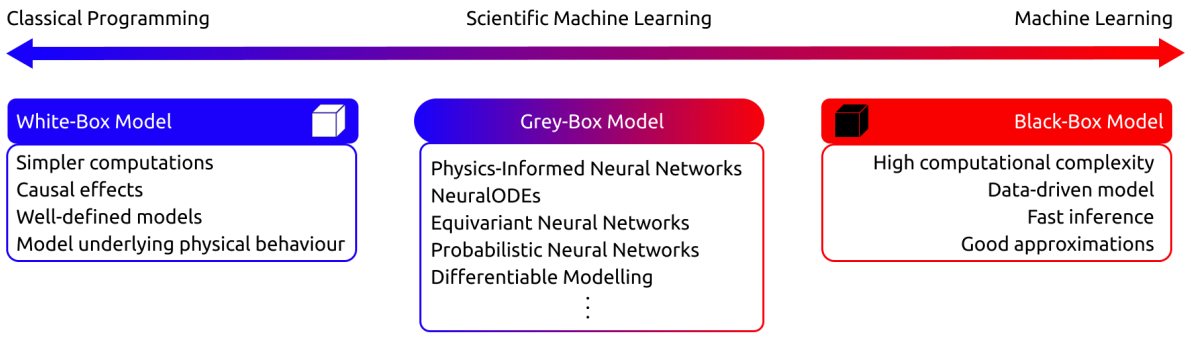

Figure 2.1: A sketch of the grey-box model. The grey-box models combine the advantages of both the black and white box models.

2.2 Differentiable modelling in imaging

Differentiable programming plays a central role in the domain of deep learning. Coupled with with gradient-based optimization techniques, it is extensively employed across various scientific machine learning applications to tackle complex problems involving systems with adjustable parameters. This approach allows for the efficient computation of gradients necessary for training models, thereby facilitating the tuning of parameters to optimize performance on specific tasks. Given a parameterized linear operator $\phi_\theta : x \in X \rightarrow y \in Y$ where X and Y are two vector spaces, the gradient optimization process for the task L is achievable through

$$\frac{\partial L(y)}{\partial x} = \frac{\partial L(y)}{\partial y}\frac{\partial \phi_\theta(x)}{\partial x} = 0, \tag{2.1}$$

where we used the chain rule and imposed the first derivative to zero to minimize the loss. We can generalise the previous equation for a sequence of linear operators, giving a batch of m vector as input

$$\frac{1}{m}\sum_{i=0}^{m-1} \nabla_\theta L(x^{(i)}, y^{(i)}, \theta). \tag{2.2}$$

During the learning phase, neural networks use these gradients for learning through optimization algorithms such as ADAM or stochastic gradient descent (SGD) [66]. The process of information propagation from the output back to the input, along with the updates of the weights θ, is referred to as *backpropagation*. Despite the powerful capabilities of neural networks in learning complex patterns, they are often criticized for being 'black boxes'. This implies a lack of transparency in how these models update their parameters during the learning process and how these updates correlate to the specific tasks they are trained on. To address this issue, integrating known parameterized forward models with neural networks offers a pathway to achieve partial control over the model's behavior. These forward models can range from sequential linear transformations,

as in traditional signal processing, to the integration of differential equations representing physical processes. By embedding these well-understood structures into the learning process, we can gain a more interpretable and controlled approach to how neural networks operate and evolve, providing a bridge between the robustness of machine learning techniques and the interpretability of traditional modeling methods.

2.2.1 Data models for images

A key contribution of this thesis is the conceptualization and implementation of the image acquisition process as a parameterized forward model. This endeavor involves not just a simple application of deep learning techniques to image processing, but a fundamental rethinking of how these processes interact and complement each other. A conventional camera captures raw data through a Color Filter Array (CFA). The CFA consists of a matrix of color filters placed over the image sensor. Each pixel collects light at a specific wavelength and the periodic mosaic of filters that makes up the entire sensor differs by one manufacturer to another. Among the various CFA patterns, the Bayer filter is the most prevalent. It arranges the color filters in a checkerboard pattern that is 50% green, 25% red, and 25% blue. This configuration mimics the human eye's greater sensitivity to green light, aiding in more accurate luminance perception. The typical Bayer pattern includes a row of alternating green and red filters and a row of alternating blue and green filters. After capturing these raw Bayer images, a conventional camera then undergoes a series of signal processing steps to transform these raw captures into fully reconstructed images. This transformation is composed of multiple sequential stages, each with a specific role in the enhancement and refinement of the image. A more detailed description of these transformations can be found in Chapter 3. However, this multi-stage approach is not without its drawbacks. A significant issue arises in the form of residual errors. As the image data passes through each stage of processing, these errors accumulate, leading to a progressive degradation of image quality. This phenomenon underscores a fundamental challenge in conventional image signal processing: maintaining the fidelity of the final image despite the inherent imperfections introduced at each processing stage.

While physically 'data models' of images have to the best of our knowledge not yet found their way into the machine learning, they have been studied in other disciplines, in particular physical optics and metrology. Perhaps, several works investigate deep learning architectures to solve individual image signal processing's transformations, like image denoising [239, 119, 108] or demosaicing [60, 49]. Other works employ end-to-end deep convolutional neural networks to map raw to rgb images in one single step [225, 177]. [238] propose a differentiable image processing pipeline for the purpose of camera lens manufacturing. Their goal, however, is to optimize a physical component (lens) in the image acquisition process and no code or data is

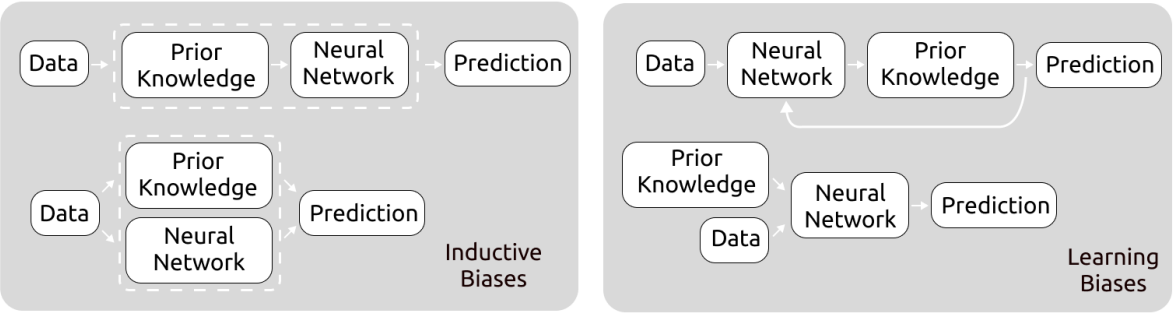

Figure 2.2: (Left) Inductive biases. The prior knowledge is implicitly encoded in the model. (Right) Learning biases. The prior knowledge drives the model learning method.

publicly available. Existing software packages that provide low-level image processing operations include Halide [174], Kornia [179] and the rawpy package [202] which can be integrated with Python and PyTorch code. We should also take note that outside optical imaging there are areas in machine learning and applied mathematics, in particular inverse problems such as magnetic resonance imaging (MRI) or computed tomography, that make use of known operator learning [138, 137] to incorporate the forward model in the optimization [23] or, as in the case of MRI, learn directly in the k-space [266].

Raw image data Camera raw files contain the data captured by the camera sensors [12]. In contrast to processed formats such as `.jpeg` or `.png`, raw files contain the sensor data with minimal processing [243, 151, 127]. The processing of the raw data usually differs by camera manufacturer thus contributing to dataset drift. Existing raw data sets from the machine learning, computer vision and optics literature can be organized into two categories. First, datasets that are sometimes treated - usually not by the creators but by users of the data – as raw data but which are in fact not raw. Examples for this category can be found for both modalities considered here [35, 7, 19, 133, 267]. All of the preceding examples are processed and stored in formats including `.jpeg`, `.tiff`, `.svs`, `.png`, `.mp4` and `.mov`. Second, datasets that are labelled raw data which are raw. In contrast to the labelled and precisely calibrated raw data presented here, existing raw datasets [38, 27, 2, 73] are collected from various sources for image enhancement tasks without full specification of the measurement conditions or labels for classification or segmentation tasks.

2.2.2 Bridging data models with deep learning

Physically-fainthful image emulation The raw data captured by imaging devices is subject to a range of physical perturbations, like shot noise given different illumination, thermal noise depending on the sensor or different optical components. These perturbations, once introduced,

tend to amplify as they pass through the image processing forward model. This exponential propagation results in a diverse array of processed images, each bearing the cumulative impact of these initial variances. The synthesis of these realistic variations in computer vision is often done by applying augmentations directly to the processed data, e.g. a `.jpeg` or `.png` image. Hendrycks et al. [76] have done foundational work in this direction developing a practical, standardized benchmark. However, there is no guarantee that noise added to a processed image will be physically faithful. This is problematic, as nuances matter [43] for assessing the cascading effects data models have on the task model downstream [9, 197]. For the same reason, the use of generative models [62] like GANs has been limited for test data generation as they are known to hallucinate visible and less visible artifacts [34, 201]. Other approaches, like the WILDS data catalogue [14, 103], build on manual curation of so called natural distribution shifts, or, like [221], on artificial worst case constructions. These are important tools for the study on how these perturbations affects deep learning models, especially those that are created outside the camera image signal processing.

In the absence of explicit differentiable data models and raw sensor data, the shared limitation of catalogue approaches is that metrologically faithful data synthesis is not possible and the data generating process cannot be granularly studied and manipulated.

Adversarial failure detection Adversarial attacks have emerged as a useful tool for probing the understanding and robustness deep learning models. These attacks involve manipulating images with small perturbations to fool downstream machine learning models. However, as described above, processed images are already affected by several systematic errors. This means that an adversarial perturbation for a specific camera is not resilient to different sensors and image signal processings. Phan et al.[165] investigated this problem with a differentiable raw processing pipeline, propagating the gradient information back to the raw image. The signal is used for adversarial search. In their work, they optimize adversarial noise on a per-image basis in the raw space. In Chapter 3, we explore adversarial perturbations from a different perspective. Differently from Phan et al, we modify the parameters of the data model itself in pursuit of harmful parameter configurations. The goal is not simply to fool a classifier, but to discover failure modes and susceptible parameters in the data model that will have the most influence on the task model's performance.

Data processing optimization An explicit and differentiable image processing data model allows joint optimization together with the task model. This approach has already shown promising results in the field of radiology imaging. Pioneering works like [183, 224, 136] have utilized this methodology, although the primary focus in these studies is on optimizing sampling patterns. For optical data, a parallel line of research has been exploring the role of inductive biases in the

image acquisition process. Jaroensri et al. [90] involved a parameterized pipeline to achieve better denoising on the image. Perhaps, they start from .jpeg processed images, they undo only the gamma compression and they apply the forward model on that linear image. In that case, most of the processing transformations still affect the output result. Tseng et al. [238] optimised the optical compound in tandem with software and hardware image signal processing. This approach, they demonstrated, enhances visual detail across various fields and surpasses traditional pipelines. The same result is classically achieved in over a month of work by expert designers using Zemax. They train a neural network to map from the optical parameters to the point spread functions PSFs. In contrast to this thesis, they decomposed the optical model the system in small end-to-end blocks, while our gradient flows backward through an optical forward model.

2.2.3 Hidden technical debt in imaging

The machine learning life cycle refers to the process and stages involved in developing, deploying, and maintaining machine learning models for real-world applications. When an ML model goes into production, we need to investigate the whole infrastructure around it.

For more than a decade, researchers have primarily focused on designing ML models to achieve the highest accuracy on benchmark datasets like MNIST, ImageNet, or CIFAR. In a pioneering move toward a data-centric approach, [206] established a significant milestone in this field. They asserted that designing the ML model constitutes only a small part of the ML deployment and maintenance pipeline. They claimed that designing an ML model is a small part of the whole process, introducing an *hidden technical debt* in the ML deployment and maintainance workflow. Data cleaning and normalisation become essential, highlighting the need to investigate more on the data collection than the model architecture.

Even after six years, [198] still claimed that 'everyone wants to engage in model work rather than data work'. This underscores the prevailing inclination towards model-centric activities over data-centric ones. Nevertheless, this observation highlights the increasing importance of focusing more on data, especially in complex ML applications.

An example of hidden technical debt can be highlighted in medical imaging. Training model on data from a single hospital can lead to significant challenges when attempting to evaluate the same model to data from another hospital. This discrepancy primarily arises due to variations in data collection protocols, staining procedures, and imaging equipment between institutions. Such differences can lead to discrepancies in image characteristics, which, if not accounted for, may result in a model that performs well in its training environment but poorly when deployed elsewhere. This scenario underscores the critical importance of incorporating a diverse dataset during the training phase that reflects the potential variability in clinical environments. By doing

so, the model can be more robust and generalizable, thereby enhancing its utility across different hospitals without the need for extensive retraining or adjustments.

In this data-centric perspective, scientific machine learning offers a novel solution for ML deployment pipelines. It allows for the integration of physical concepts within both model development and data preparation stages. This integration enhances the model's resilience to different cameras, making long-term maintenance more manageable.

2.2.4 Differential equations as prior

Even if in this thesis we consider forward processes composed by reversible/irreversible operators, it is worth mentioning that the differential equation which describes the motion of a system is widely used to guide the learning process in physics applications. For a given complex system, solving the differential equation is challenging from a mathematical point of view and computationally expensive with Monte Carlo simulations. Neural networks can be used to facilitate this process and infer the dynamics. Physics-informed neural networks constrain the output to satisfy the differential equation via the loss function [175]. This hard constraint on the solution, allows the network to infer the dynamics of the system given the initial conditions, with a small error depending on how much accurate the differential equation describes the physical system. In a similar way, Lagrangian and Hamiltonian neural networks preserve the energy conservation of the system, inferring the next states in the phase space of a system [237, 37]. On the other hand, NeuralODEs take the dynamics and estimate the solution of the differential equation without having knowledge of the system [31]. This kind of architecture can enhance the scientific discovery given input/output data. While these techniques employ neural networks as dynamics solvers, in this thesis we investigate how to jointly combine the network and the forward model in a unique differentiable model.

2.3 Generative models

Generative models are a class of neural networks which aim to learn and reproduce the true data distribution $\mathcal{X}$ starting from simpler distributions $\mathcal{Z}$

$$g_\theta : \mathcal{Z} \to \mathcal{X}. \tag{2.3}$$

In the last decade, different generative models have been introduced to overcome the necessity to resemble complex dataset distributions. Generative Adversarial Networks (GANs) [65] were

recognised as unchallenged state-of-the-art generative models. However, the delicate balance between the generator and the discriminator leads to several problems, like mode collapsing or vanishing gradient, making the model extremely hard to train. On the other hand, architectures like variational autoencoders (VAEs) [100], flow-based models [178] or autoregressive models [162] are more mathematically interpretable than GANs but they sample new data with lower fidelity and diversity.

In recent years, a new generative model family has outperformed previous models in various image generation tasks. Diffusion Models (DMs) [211, 215, 80] represent a class of parameterized Markov chains that effectively optimize the lower variational bound associated with the likelihood function of the unknown data distribution. DMs can approximate complex distributions much more faithfully than GANs and, by extension, generate more diverse samples without compromising on fidelity [152].

2.3.1 Diffusion models

The model Given an image sampled from a dataset distribution $x_0 \sim q(x_0)$, a diffusion model is composed by a *forward process* $q(x_{1:T}|x_0)$ which gradually degrades the quality of the image up to a simple gaussian distribution $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ and by a parametric *backward process* $p_\theta(x_{0:T})$ which reverses the degradation (see Fig. 2.3). Both processes can be formalised as Markov chains

$$q(x_{1:T} \mid x_0) = \prod_{t=1}^{T} q(x_t \mid x_{t-1}), \qquad \text{where } q(x_t \mid x_{t-1}) \equiv \mathcal{N}\left(x_t; \sqrt{\alpha_t} x_{t-1}, (1-\alpha_t)\mathbf{I}\right) \quad (2.4)$$

$$p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t), \quad \text{where } p(x_{t-1}|x_t) \equiv \mathcal{N}\left(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)\right) \quad (2.5)$$

where α_t is a degradation scheduler function. Step-wise, The model leverages the reparameterization trick introduced with VAEs to estimate the amount of noise added in the forward process. Starting from the assumption that both the transition probabilities $q(x_t|x_{t-1})$ and $p_\theta(x_{t-1}|x_t)$ are normally distributed, [80] defined a probability distribution $q(x_{t-1} \mid x_t, x_0)$ to infer the next state.

Improve sampling speed The increased diversity of samples while preserving sample fidelity comes at the cost of training and sampling speed, with diffusion models being much slower than GANs [42]. The universally adopted solution to this problem is to encode the images from pixel space into a lower dimensional latent space via a Vector Quantised-Variational AutoEncoder (VQ-VAE), and perform the diffusion process over the latents, before decoding back to pixel space [181]. Pairing this with the Denoising diffusion implicit models (DDIMs) sampling method

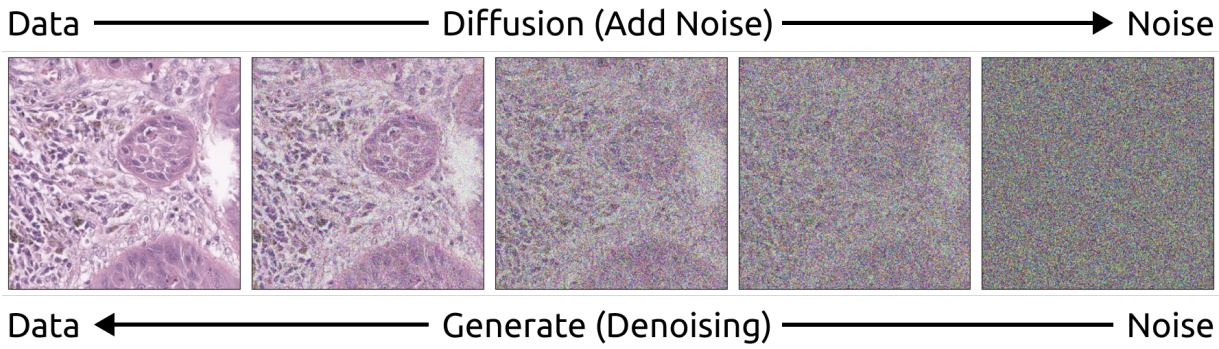

Figure 2.3: Diffusion model idea. The training dataset is slowly degraded by adding noise up to a noisy distribution. In the backward process, the neural network removes this noise step by step, generating a new sample.

[214] leads to faster sampling while preserving the DM objective

$$z_{t-1} = \sqrt{\alpha_{t-1}} \left(\frac{z_t - \sqrt{1-\alpha_t}\epsilon_\theta(z_t, t)}{\sqrt{\alpha_t}} \right) + \sqrt{1 - \alpha_{t-1} - \sigma_t^2}\epsilon_\theta(z_t, t) + \sigma_t \epsilon_t, \tag{2.6}$$

where z_t is the latent variable at timestep t in the VQ-VAE latent space, α_t is the noise scheduler, ϵ_θ is the noise learned by the model and ϵ_t is random noise.

Conditioning Conditioning can be achieved either by specifically feeding the condition with the noised data [81, 191], by guiding an unconditional model using an external classifier [153, 216] or by classifier-free [79] guidance used in this work, where the convex combination

$$\tilde{\epsilon}_\theta(z_t, c) = (1 + \omega)\epsilon_\theta(z_t, c) - \omega\epsilon_\theta(z_t, \varnothing), \tag{2.7}$$

of a conditional diffusion model $\epsilon_\theta(z_t, c)$ and an unconditional model $\epsilon_\theta(z_t, \varnothing)$ is used for noise estimation. The parameter ω controls the tradeoff between conditioning and diversity since $\omega > 0$ introduces more diversity in the generated data by considering the unconditional model while $\omega = 0$ uses only the conditional model.

Embedding prior knowledge From an implementation point of view, it is possible to guide diffusion models in several ways. A common ingredient is the attention mechanism. The most commonly used network is the U-Net, where the building blocks can be adapted to the specific task. The most commonly used U-net block is composed of an embedding block, combined with the output of the previous block, fed into a ResNet block and sequentially to a self-attention layer (see Fig. 2.4a). In the embedding block, we feed the time step and the prior knowledge of the system, which can be a label, mask, signal or image. In Fig. 2.4b, we provide some embedding examples. The easiest example is the unconditional diffusion model. In that case, a multilayer perceptron

(MLP) takes in input the time steps and returns an embedding for the ResNet block. If we have a set of labels, map them in a dictionary (in our case we used the pytorch function nn.Embedding), and feed them with the timestep into the MLP. If we want to generalise to a whole sentence, transformer architectures combined with image encoders provide a robust representation of the input space. CLIP's embeddings [173] have shown how a correct pairing of a batch of (image, text) leads to a robust generation process in models like DALL-E [176, 208]. In image-to-image translation processes, encoding the input image without losing spatial information is a challenging task. The first approach would be to feed directly the conditioning image in input with the noise latent variable [193, 81]. Working in the pixel space is computationally expensive, for this reason, a compromise would be losing a bit of information by downsampling the image for every U-net block and feeding it as a query in a cross-attention block with the noisy input. This guiding method provides a robust and consistent output with the conditional image [182].

Fine-tuning Training diffusion models is a resource-intensive process, often requiring significant time and computational power. Most of the works involves using powerful hardware like the A100 with 80GB VRAM for extended periods, sometimes up to a week, in efforts to surpass existing benchmarks. However, the emergence of large foundational models has introduced the way for more resource-efficient fine-tuning of diffusion models on specific datasets. A popular method for is ControlNet [263], which involves creating duplicates of the embedding layers. This efficiency is achieved by freezing the entire model and introducing perturbations in the embedding blocks. With this approach, it is possible to fine-tune a text to image diffusion model to a model conditioned on an arbitrary input. In terms of flexibility, Dreambooth [188] employs a few-shot fine-tuning process to produce novel images. This model assigns unique identifiers to a small number of training images (typically 3-5) and instructs the model to use these identifiers for generating new content. While these methods speed-up the fine-tuning by updating a subset of parameters, a novel technique, originally developed for large language models, can update all the parameters with fewer resources. Low Ranking Adaptation (LoRA) [85] decomposes the parameters matrices into a product of two lower-rank tensors. Although choosing the rank involves a trade-off between quality and fine-tuning speed, this method can achieve high fidelity outputs by training approximately $\sim 1\%$ of the total trainable parameters.

Score-based models It is worth mentioning that the diffusion model community is moving to a more generalised mathematical framework, the score-based models which are more mathematically interpretable and easier to optimize [217, 97]. Score-based models are described by the following stochastic equation:

$$dx = f(x,t)dt + g(t)dw \qquad \textit{Forward Model} \tag{2.8a}$$

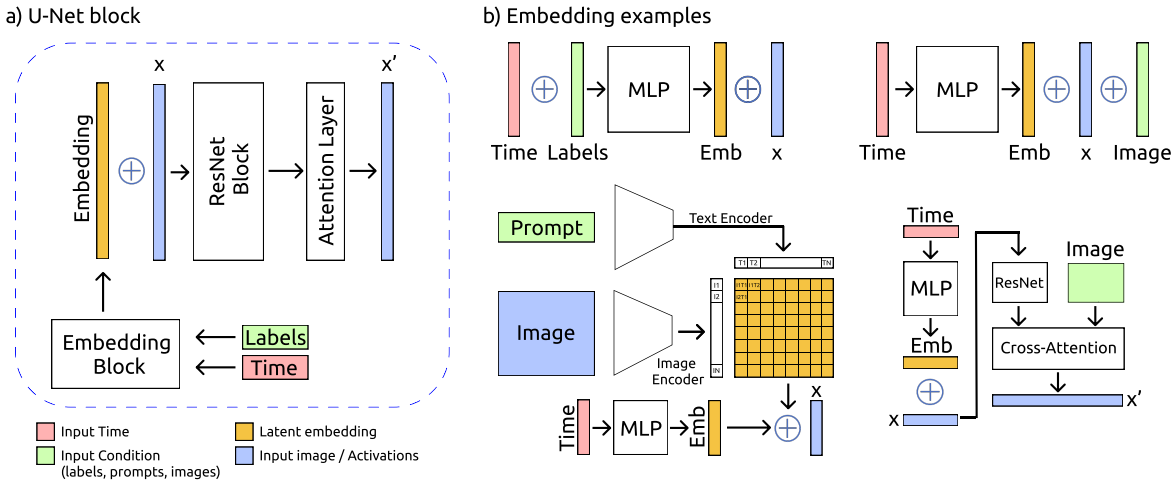

Figure 2.4: Diffusion model embedding examples. a) Core U-Net block. b) (Top-Left) Embedding jointly time and labels via a Multi-Perceptron Layer (MPL). (Top-Right) Embedding time via a MPL and concatenate the condition (image) directly with the output. (Bottom-left) CLIP prompt-image embedding. (Bottom-right) Cross-attention embedding between the condition (image) and ResNet block output.

$$dx = \left[f(x,t) - g(t)^2 \nabla_x \log p_t(x)\right] dt + g(t) dw, \qquad \textit{Reverse Model} \tag{2.8b}$$

where $f(x,t)$ is the drift coefficient, $g(t)$ is the diffusion coefficient and $\nabla_x \log p_t(x)$ is the score function. The model introduced in [80], and used in this thesis, can be obtained by choosing $f(x,t) = -\frac{1}{2}\beta(t)x$ and $g(t) = \sqrt{\beta(t)}$.

2.3.2 Generate beyond the edges

Large-content image generation can be reduced to inpainting/outpainting tasks. Image inpainting is the problem of reconstructing unknown or unwanted areas within an image. It plays a significant role in many downstream computer vision tasks, such as image restoration, object removal, image editing and manipulation. A closely related task is image outpainting, which aims to predict visual content beyond the boundaries of an image. In both cases, the newly in- or outpainted image regions have to be visually indistinguishable with respect to the rest of the image. Such image completion approaches can help utilise models trained on smaller patches for the purpose of generating large images, by initially generating the first patch, followed by its extension outward in the desired direction.

Traditional approaches Traditional methods for completing missing image regions generally rely on reusing image features from the known areas of the image. Expensive nearest neighbour

searching is required to select the most appropriate pixels [46, 249, 16] or patches [255, 124] with which to fill in the missing regions. Furthermore, increasing the size or the structural complexity of the regions to be filled in often produces visually unsatisfactory results [16]. Rather than repurposing existing image regions, deep learning has made it possible to synthesise novel yet realistic image information required for the inpainting and outpainting tasks. While some approaches, such as Deep Image Prior [239], condition the newly generated image areas only on the rest of the image that is being in-painted, most deep learning methods will aim to learn a prior over natural images to achieve high realism of the generated regions [113, 126], while still conditioning on the known image regions at the same time.

Generative modelling for conditional image synthesis Generative Adversarial Networks (GANs) [65] have dominated image-to-image translation tasks like inpainting and outpainting for years [113, 259, 265, 128]. Recently, diffusion models have surpassed GANs in various image generation tasks [42]. Palette [191] was the first to apply diffusion models to tasks like inpainting and outpainting. RePaint [132] and ControlNet [263] demonstrate resampling and masking techniques for conditioning using a pre-trained diffusion model. SinDiffusion [154] and DiffCollage [264] offer state-of-the-art outpainting solutions using diffusion models trained with overlapping patches. In parallel to the work presented in Chapter 5, Bond-Taylor and Willcocks [20] developed a related approach called ∞-Diff which trains on random coordinates, allowing the generation of infinite-resolution images during sampling. However, in contrast to our approach, the method does not involve image compression in a latent space.

Synthetic data assessment The authenticity of synthetic data produced by diffusion models, trained on vast paired labelled datasets [204], remains contentious. Ethical implications necessitate distinguishing if generated images are replicas of training data [213, 24]. The task is complicated due to subjective visual similarities and diverse dataset ambiguities. Various metrics have been proposed for quantifying data replication, including information theory distances from real data [245], consistency measurements using downstream models [4], comparison with inpainted areas [24], and detection of “forgotten" examples [89].

Chapter 3.

Data Models for Dataset Drift Control with Raw Images

In this chapter, we demonstrate how modelling the image data processing and the sensor calibration profile enables more precise control of the validation of machine learning model robustness to dataset drift. We connect raw image data, differentiable data models and the standard machine learning pipeline. This combination enables three novel, physically faithful validation protocols that can be used towards the intended use specifications of machine learning systems, a necessary pre-requisite for the use of any technology in many application domains such as medicine or autonomous vehicles.

3.1 Introduction

Camera image data are a staple of machine learning research, from the early proliferation of neural networks on MNIST [250, 54, 55, 117] to leaps in deep learning on CIFAR and ImageNet [106, 189, 107] or high-dimensional generative models [130, 96]. Camera images also play an important role in the delivery of various high-impact public and commercial services. Deep learning has enabled the automation or enhancement of such services. During the 2010s, "deep learning for ..." became an increasing trend for a wide range of applications domains like cosmology [240], spanning medicine and biology (microscopy for cell detection [51, 230, 143, 247], histopathology [104, 71], opthalmology [251, 233, 236], malaria detection [140, 169, 52, 148]) and more. However, the excitement has been reined in by calls for caution. Machine learning systems exhibit particular failure modes that are contingent on the makeup of their inputs [226, 187, 220]. Many findings from the machine learning robustness literature confirm supervised learning's tremendous capacity for identifying features in the training inputs that are correlated with the true labels [115, 88, 58, 167, 61]. But these findings also point to a flipside of this capacity:

the sensitivity of the resulting machine learning model's performance to changes – both large and small – in the input data. Since this dependency relates to generalization, a top objective in machine learning, the implications have been studied across most of its many sub-disciplines including robustness validation [77, 212, 203, 84], formal model verification [254, 63, 28], out-of-distribution detection [218, 118, 149, 45], semi- [161, 253, 17] and self-supervised learning [232, 69], federated learning [199, 200, 246], or compression [235, 244, 50], among others.

We refer to the mechanism underlying changes in the input data as *dataset drift*. Formally, we characterize it as follows. Let $(\boldsymbol{X}_{RAW}, Y) : \Omega \to \mathbb{R}^{H,W} \times \mathcal{Y}$ be the raw sensor data generating random variable on some probability space $(\Omega, \mathcal{F}, \mathbb{P})$,[1][2] for example with $\mathcal{Y} = \{0,1\}^K$ for a classification task.[3] Raw inputs $\boldsymbol{x}_{\text{RAW}}$ are in a data state before further processing is applied, in our case photons captured by the pixels of a camera sensor as displayed in the outputs of the "Measurement" block in Figure 3.1. The raw inputs $\boldsymbol{x}_{\text{RAW}}$ are then further processed by a *data model* $\boldsymbol{\Phi}_{\text{Proc}} : \mathbb{R}^{H,W} \to \mathbb{R}^{C,H,W}$, in our case the measurement hardware like a camera itself or other downstream data processing pipelines, to produce a processed view $\boldsymbol{v} = \boldsymbol{\Phi}_{\text{Proc}}(\boldsymbol{x}_{\text{RAW}})$ of the data as illustrated in the output of the "Data model" block in Figure 3.1. This processed view $\boldsymbol{v}$ could for example be the finished RGB image, the image data state that most machine learning researchers typically work with to train a *task model* $\boldsymbol{\Phi}_{\text{Task}} : \mathbb{R}^{C,H,W} \to \mathcal{Y}$. Thus, in the conventional machine learning setting we obtain $\boldsymbol{V} = \boldsymbol{\Phi}_{\text{Proc}}(\boldsymbol{X}_{RAW})$ as the image data generating random variable with the target distribution $\mathcal{D}_t = \mathbb{P} \circ (\boldsymbol{V}, Y)^{-1}$.[4] A different data model $\tilde{\boldsymbol{\Phi}}_{\text{Proc}}$ generates a different view $\tilde{\boldsymbol{V}} = \tilde{\boldsymbol{\Phi}}_{\text{Proc}}(\boldsymbol{X}_{RAW})$ of the same underlying raw sensor data generating random variable $\boldsymbol{X}_{RAW}$, resulting in the *dataset drift*

$$\mathcal{D}_s = \mathbb{P} \circ (\tilde{\boldsymbol{V}}, Y)^{-1} \neq \mathcal{D}_t. \tag{3.1}$$

This inequality indicates the distribution shift from $\mathcal{D}_s$ to $\mathcal{D}_t$ due to a shift from the random variable $\boldsymbol{V}$ to $\tilde{\boldsymbol{V}}$, or with the joint notation with the label, from $(\boldsymbol{V}, Y)$ to $(\tilde{\boldsymbol{V}}, Y)$. This characterization of dataset drift is closely related to the concept of distributional robustness in the sense of Huber where "the shape of the true underlying distribution deviates slightly from the assumed model" [86]. Note that the nomenclature around dataset drift is as heterogeneous as the disciplines in which it is studied. See [105] for a good discussion of cross-disciplinary terminological ambiguity. Here we are concerned with dataset drift as defined in Equation (3.1), that is changes in V that are

[1] where $\mathbb{P}$ is probability measure; Ω is the sample space of possible outcomes $\omega \in \Omega$; $\mathcal{F}$ is the σ-Algebra of $\mathbb{P}$-measurable sets such that $\mathbb{P}(A)$, $A \in \mathcal{F}$, $A \subset \mathbb{R}$ gives the probability of A under $\mathbb{P}$.

[2] We write an uppercase letter A for a real valued random variable and a lowercase letter a for its realization. A bold uppercase letter $\boldsymbol{A}$ denotes a random vector and a bold lowercase letter $\boldsymbol{a}$ its realization. For $N \in \mathbb{N}$ realizations of the random vector $\boldsymbol{A}$ we write $\boldsymbol{a}_1, ..., \boldsymbol{a}_N$. The state space of the random vector $\boldsymbol{A}$ is denoted by $\boldsymbol{\mathcal{A}} = \{\boldsymbol{A}(\omega) \mid \omega \in \Omega\}$.

[3] where K is the number of classes.

[4] Let $X : \Omega \to \mathbb{R}$ be a random variable taking values in $\mathbb{R}$. The distribution of X is characterized by the measure of $X^{-1}(A) \coloneqq \{\omega \in \Omega \mid X(\omega) \in A\}$, $A \subset \mathbb{R}$, $A \in \mathcal{F}$ under $\mathbb{P}$. This defines a new measure over $\mathbb{R}$, which characterises the distribution of X: $\mathbb{P} \circ X^{-1} = \mathbb{P}(X^{-1}(\cdot))$ where $\circ$ is composition of $\mathbb{P}$ with the pre-image of X.

induced by changes in $\mathbf{\Phi}_{\text{Proc}}$ which some works also refer to as covariate shift or more generally as distribution shift. In practice, a possible reason for such a dataset drift to occur in images is a change in the camera types or settings, for example different acquisition microscopes across different lab sites s and t that lead to drifted distributions $\mathcal{D}_s \neq \mathcal{D}_t$. Anticipating and validating the robustness of a machine learning model to these variations in a realistic way is not just an engineering concern but also mandated by quality standards in many industries [234, 70, 158]. Omissions to perform physically accurate robustness validations has, among other reasons, slowed or prevented the rollout of machine learning technology in impactful applications such as large-scale automated retinopathy screening [13], machine learning melanoma detection [223, 47] or yield prediction [139] from drone cameras.

Following the problem introduced in 2.2.3, to further illustrate the concept of dataset drift, consider the scenario where a machine learning model developed for histopathology image analysis is initially trained using high-resolution scans from one hospital's advanced scanner. If this model is then transferred to a different hospital that uses a scanner with lower resolution and different optical characteristics, significant issues can arise. The lower resolution images may not capture as much detail as the high-resolution scanner, leading to a loss in the model's accuracy and effectiveness. Optical differences, such as variations in lens quality or imaging technology, can alter the appearance of tissue samples in scans, further complicating the model's ability to accurately analyze the new data. This example underscores the necessity for models to be adaptable and sensitive to such changes in dataset characteristics, which can significantly impact performance when moving from one clinical setting to another.

Hence, the calls for realistic robustness validation of image machine learning systems are not merely an exercise in intellectual novelty but a matter of integrating machine learning research with real world infrastructures and performance expectations around its core ingredient: the data.

3.1.1 The status quo of dataset drift controls for images

How can one go about validating a machine learning model's performance under image dataset drift? The dominant empirical techniques can broadly be categorized into augmentation and catalogue testing approaches, each with their own benefits and limitations (see 3.1 for a conceptual comparison). Augmentation testing involves the application of perturbations, for example Gaussian noise, to already processed images [76, 32, 142] in order to approximate the effect of dataset drift. Given a processed dataset this allows for fast and easy generation of test samples. However, [228] point out that perturbations applied to an already processed image can produce drift artifacts that are unfaithful to the physics of camera processing. Results in optics further support the concern that the noise obtained from an image processing pipeline is distinct from noise added

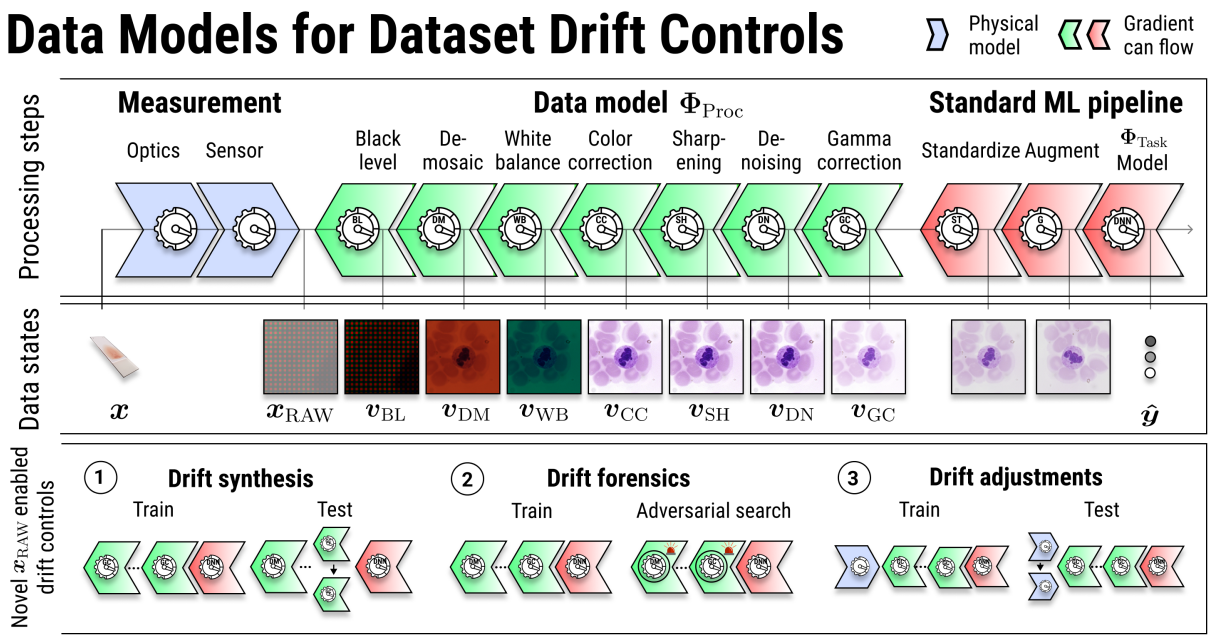

Figure 3.1: Schematic illustration of an optical imaging pipeline, the data states and novel, raw-enabled drift controls. Data $\boldsymbol{x}$ transitions through different representations. The measurement process yields metrologically accurate raw data $\boldsymbol{x}_{\text{RAW}}$, where the errors on each pixel are uncorrelated and unbiased. From the RAW sensor state, data undergoes stages of Image Signal Processing (ISP) $\boldsymbol{\Phi}_{\text{Proc}}$, the data model we consider here. Finally, the data is consumed by a machine learning task model $\boldsymbol{\Phi}_{\text{Task}}$ which outputs $\hat{\boldsymbol{y}}$. Combining raw data with the standard machine learning pipeline and a differentiable data model $\boldsymbol{\Phi}_{\text{Proc}}$ enables useful controls for dataset drift comprising ① drift synthesis, ② drift forensics, and ③ processing adjustments under drift.

to an already processed image [243, 90]. For illustration, assume we carry out augmentation testing to test the robustness of the task model wrt. to the dataset drift (3.1). Let $\boldsymbol{\xi} \sim \mathcal{D}_{noise}$ be a noise sample additively applied to the the view resulting in $\boldsymbol{v} + \boldsymbol{\xi}$. Doing so, the task models robustness is tested wrt. the distribution $\mathbb{P} \circ (V + \Xi)^{-1}$ that might not approximate $\mathcal{D}_s$ well. Since $\mathbb{P}$ is unknown, this is difficult to resolve but at least we could require that a sample used for robustness testing is an element of the image $\tilde{\boldsymbol{\Phi}}_{\text{Proc}}\left[\boldsymbol{\mathcal{X}}_{RAW}\right]$ of $\boldsymbol{\mathcal{X}}_{RAW}$ under $\tilde{\boldsymbol{\Phi}}_{\text{Proc}}$. Following this argumentation, we define a *physically faithful* data point wrt. the dataset drift (3.1) as a view $\tilde{\boldsymbol{v}}$ that satisfies $\tilde{\boldsymbol{v}} \in \tilde{\boldsymbol{\Phi}}_{\text{Proc}}\left[\boldsymbol{\mathcal{X}}_{RAW}\right]$. In augmentation testing, the test samples are not restricted to physically faithful data points wrt. to any dataset drift, since $\boldsymbol{v} + \boldsymbol{\xi} \in \tilde{\boldsymbol{\Phi}}_{\text{Proc}}\left[\boldsymbol{\mathcal{X}}_{\text{RAW}}\right]$ might not hold true for any data model.

A physically faithful alternative to *augmentation testing* is what we call *catalogue testing*. It involves the collection of datasets from different cameras, which are then used as hold-out robustness validation datasets [102, 5, 141, 125]. It does not allow for as flexible and fast in-silico simulation of test cases as augmentation testing because cataloguing requires expensive data collection, after which the test cases are "locked-in". Notwithstanding, catalogue testing comes with the appealing guarantee that test samples conform to the processing physics of the different

	Augmentation testing	Catalogue testing	Data models
Simulation of test samples	✓	✗	✓
Physically faithful test samples	✗	✓	✓
Differentiable data model	✗	✗	✓

Table 3.1: A conceptual comparison of different empirical approaches to dataset drift validation for machine learning task models. While augmentation testing allows the flexible, ad-hoc synthesis of test cases, they are not guaranteed to be physically faithful in contrast to catalogue testing. Pairing qualified raw data with explicit data models allows for the flexible synthesis of physically faithful test cases. In addition, the differentiable data model opens up novel drift controls, including drift forensics and drift adjustments.

cameras they were gathered from, ensuring that only physically faithful data points are used for testing.

However, the root of input data variations – the data model of images – has received little attention in machine learning robustness research to date. While machine learning practitioners are acutely aware of the dependency between data generation and downstream machine learning model performance, as 75% of respondents to a recent study confirmed [197], data models are routinely treated as a black-box in the robustness literature. This blind spot for explicit data models is particularly surprising since they are standard practice in other scientific communities, in particular optics and metrology [12, 185, 227, 21], as well as advanced industry applications, including microscopy [72, 164, 261] or autonomous vehicles [258, 95, 186].

3.1.2 Scope and limitation of the proposed methods

The systems infrastructure for optical imaging produced by large industry vendors such as ZEISS, Hamamatsu, Teledyne-Photometrics, Andor, Yokogawa or Perkin-Elmer allow raw sensor readouts. The same applies to consumer-grade cameras by market-dominating vendors such as Samsung or Apple. Concurrently, ISP-processed data is predominantly used in practice - both in the application domains considered here and machine learning. The reasons are often downstream workload dependencies. In most settings, data is acquired to be human readable for a specific task, for example diagnostics or environmental surveying. Variations in the ISPs, between different vendors or acquisition sites, then lead to the drifts of 3.1 outlined above. This work is targeted at the current imaging infrastructure that (i) makes widespread use of ISPs that lead to drift and (ii) simultaneously allows access to raw sensor readouts.

To illustrate the practical challenges associated with ISP-induced data drift, consider a scenario in which a machine learning model is trained on images captured with an iPhone. These images are not only processed through Apple's proprietary ISP but are also optimized for enhanced visual

appeal, involving specific color corrections and dynamic range adjustments. When this trained model is then applied to images captured by a Samsung device, which uses a different ISP with its own unique processing characteristics, the performance can significantly degrade. This disparity arises because the model has learned the specific data characteristics—such as color balance and texture details—encoded by the iPhone's processing algorithms. This example underlines the need for models that can adapt to or accommodate the variations in raw data processing between devices. The ability to emulate and control these ISP-specific characteristics within a data model would not only help in reducing the time and cost associated with retraining models for each new device but also improve the robustness and accuracy of applications across different hardware platforms.

For this setting, we propose data models that allow engineers to explore, emulate and control different data generating processes related to the ISP at low cost in a physically faithful way. In terms of practical benefits, data models can save time and money (drift synthesis) as additional acquisitions, on the order of days or even weeks, can be avoided. They also open up completely new applications for integrated data-model quality management (drift forensics, drift optimization) which are impossible without differentiable data models. There are other important sources of drift, such as the sensor or optical components of the camera, which cannot be captured by this framework but will be investigated further in 4. For example, in microscopy images, factors such as the choice of the colour filter array, the point-spread function, and mechanical drift can influence the final image quality. Similarly, in aerospace imagery, the choice of lens, f-number, PSF circle diameter, ISO, and the gain applied to pixel values can affect the acquired images. These factors introduce variations contributing to dataset drift. The data models presented in this study aim to account for changes during the ISP. While extensions beyond this setting are opportune, raw-only, as sketched out in the drift optimization experiments, would require a shift in the dominant imaging workflows of the application domains considered here. In summary, the current data models offer rich utility to capture ISP as a dominant source of data drift, but are limited to the ISP scope and require an extension to model additional factors outside that scope in a physically faithful manner.

3.2 Methods

3.2.1 A data model for images

Before proceeding with a description of the methods we use to obtain the data models $\mathbf{\Phi}_{\text{Proc}}$ in this study, let us briefly review the distinction between raw data $\boldsymbol{x}_{\text{RAW}}$, processed image $\boldsymbol{v}$ and the mechanisms $\mathbf{\Phi}_{\text{Proc}} : \mathbb{R}^{H,W} \to \mathbb{R}^{C,H,W}$ by which image data transitions between these states.

We recommend [185] for a good introduction to the physics of digital optical imaging. Image acquisition has traditionally been optimized for the human perception of a scene [87, 185]. Human eyes detect only the visible spectrum of electromagnetic radiation, hence imaging cameras in different application domains such as medical imaging or remote sensing are usually calibrated to aid the human eye perform a downstream task. This process that gives rise to optical image data, which ultimately forms the backbone for any machine learning downstream, is rarely considered in the machine learning literature. Conversely, most research to date has been conducted on processed RGB image representations.

The *raw sensor image* $\boldsymbol{x}_{\text{RAW}}$ obtained from a camera differs substantially from the processed image that is used in conventional machine learning pipelines. The $\boldsymbol{x}_{\text{RAW}}$ state appears like a grey scale image with a grid structure (see $\boldsymbol{x}_{raw}$ in 3.1). This grid is given by the Bayer color filter mosaic, which lies over sensors [12]. The final *RGB image* $\boldsymbol{v}$ is the result of a series of transformations applied to $\boldsymbol{x}_{\text{RAW}}$. For many steps in this process different possible algorithms exist. Starting from a single $\boldsymbol{x}_{\text{RAW}}$, all those possible combinations can generate an exponential number of possible images that are slightly different in terms of colors, lighting and blur - variations that contribute to dataset drift. In 3.1 a conventional pipeline from $\boldsymbol{x}_{\text{RAW}}$ to the final RGB image $\boldsymbol{v}$ is depicted. Here, common and core transformations are considered. Note that depending on the application context it is possible to reorder or add additional steps. The symbol $\boldsymbol{\Phi}_{\text{i}}$ is used to denote the i^{th} transformation and $\boldsymbol{v}_{\text{i}}$ (*view*) for the output image of $\boldsymbol{\Phi}_{\text{i}}$. The first step of the pipeline is *black level* correction $\boldsymbol{\Phi}_{\text{BL}}$, which removes any constant offset. The image $\boldsymbol{v}_{\text{BL}}$ is a grey image with a Bayer filter pattern. A *demosaicing* algorithm $\boldsymbol{\Phi}_{\text{DM}}$ is applied to construct the full RGB color image [123]. Given $\boldsymbol{v}_{\text{DM}}$, intensities are adjusted to obtain a neutrally illuminated image $\boldsymbol{v}_{\text{WB}}$ through a *white balance* transformation $\boldsymbol{\Phi}_{\text{WB}}$. By considering color dependencies, a *color correction* transformation $\boldsymbol{\Phi}_{\text{CC}}$ is applied to balance hue and saturation of the image. Once lighting and colors are corrected, a *sharpening* algorithm $\boldsymbol{\Phi}_{\text{SH}}$ is applied to reduce image blurriness. This transformation can make the image appear more noisy. For this reason a *denoising* algorithm $\boldsymbol{\Phi}_{\text{DN}}$ is applied afterwards [68, 231]. Finally, *gamma correction*, $\boldsymbol{\Phi}_{\text{GC}}$, adjusts the linearity of the pixel values. For a closed form description of these transformations see 3.2. Compression may also take place as an additional step. It is not considered here as the input image size is already small. Furthermore, the effect of compression on downstream task model performance has been thoroughly examined before [41, 94, 260, 170, 168]. However, users of the code provided can add this step or reorder the sequence of steps in the modular processing object class per their use case needs. See `pipeline_torch.py` and `pipeline_numpy.py` in our code.

3.2.2 The data model

The second ingredient of this study is the data models of image processing. Let $(\boldsymbol{X}_{RAW}, Y) : \Omega \to \mathbb{R}^{H,W} \times \mathcal{Y}$ be the raw sensor data generating random variable on some probability space $(\Omega, \mathcal{F}, \mathbb{P})$, with $\mathcal{Y} = \{0,1\}^K$ for classification and $\mathcal{Y} = \{0,1\}^{H,W}$ for segmentation. Let $\boldsymbol{\Phi}_{\text{Task}} : \mathbb{R}^{C,H,W} \to \mathcal{Y}$ be the task model determined during training. The inputs that are given to the task model $\boldsymbol{\Phi}_{\text{Task}}$ are the outputs of the data model $\boldsymbol{\Phi}_{\text{Proc}}$. We distinguish between the raw sensor image $\boldsymbol{x}_{\text{RAW}}$ and a *view* $\boldsymbol{v} = \boldsymbol{\Phi}_{\text{Proc}}(\boldsymbol{x}_{\text{RAW}})$ of this image, where $\boldsymbol{\Phi}_{\text{Proc}} : \mathbb{R}^{H,W} \to \mathbb{R}^{C,H,W}$ models the transformation steps applied to the raw sensor image during processing.

The objective in supervised machine learning is to learn a task model $\boldsymbol{\Phi}_{\text{Task}} : \mathbb{R}^{C,H,W} \to \mathcal{Y}$ within a fixed class of task models $\mathcal{H}$ that minimizes the expected loss wrt. the loss function $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \to [0, \infty)$, that is to find $\boldsymbol{\Phi}^{\star}_{\text{Task}}$ such that

$$\inf_{\boldsymbol{\Phi}_{\text{Task}} \in \mathcal{H}} \mathbb{E}\left[\mathcal{L}\left(\boldsymbol{\Phi}_{\text{Task}}(\boldsymbol{V}), Y\right)\right] \tag{3.2}$$

is attained. Towards that goal, $\boldsymbol{\Phi}_{\text{Task}}$ is determined during training such that the empirical error

$$\frac{1}{N} \sum_{n=1}^{N} \mathcal{L}\left(\boldsymbol{\Phi}_{\text{Task}}(\boldsymbol{v}_n), y_n\right) \tag{3.3}$$

is minimized over a sample $\mathcal{S} = \left((\boldsymbol{v}_1, y_1), ..., (\boldsymbol{v}_N, y_N)\right)$ of views. Modelling in the conventional machine learning setting begins with the image data generating random variable $(\boldsymbol{V}, Y) = \left(\boldsymbol{\Phi}_{\text{Proc}}(\boldsymbol{X}_{RAW}), Y\right)$ and the target distribution $\mathcal{D}_t = \mathbb{P} \circ (\boldsymbol{V}, Y)^{-1}$. Given a dataset drift $\mathcal{D}_s = \mathbb{P} \circ (\tilde{\boldsymbol{V}}, Y)^{-1} \neq \mathcal{D}_t$, as specified in Equation (3.1), without a data model we have little recourse to disentangle reasons for performance drops in $\boldsymbol{\Phi}_{\text{Task}}$. To alleviate this underspecification, an explicit data model is needed. We consider two such models in this study: a static model $\boldsymbol{\Phi}^{\text{stat}}_{\text{Proc}}$ and a parametrized model $\boldsymbol{\Phi}^{\text{para}}_{\text{Proc}}$.

In the following, we denote by $\boldsymbol{x}_{\text{RAW}} \in [0,1]^{H,W}$ the normalized raw image, that is a greyscale image with a Bayer filter pattern normalized by $2^{16} - 1$, i.e.

$$\boldsymbol{x}_{\text{RAW}} = \begin{bmatrix} \boldsymbol{A}_{1,1} & . & . & . & \boldsymbol{A}_{1,\frac{W}{2}} \\ . & . & & & . \\ . & & . & & . \\ . & & & . & . \\ \boldsymbol{A}_{\frac{H}{2},1} & . & . & . & \boldsymbol{A}_{\frac{H}{2},\frac{W}{2}} \end{bmatrix} \quad \textit{with} \quad \boldsymbol{A}_{h,j} = \begin{bmatrix} r_{2h+1,2w+1} & g_{2h+1,2w} \\ g_{2h,2w+1} & b_{2h,2w} \end{bmatrix}, \tag{3.4}$$

where the values $r_{2h+1,2w+1}, g_{2h+1,2w}, g_{2h,2w+1}, b_{2h,2w}$ correspond to the values measured through

Data models	Used functions		
bi,s,me	$\mathbf{\Phi}_{\text{DM}}^{\text{Bil}}$	$\mathbf{\Phi}_{\text{SH}}^{\text{SF}}$	$\mathbf{\Phi}_{\text{DN}}^{\text{MD}}$
bi,s,ga	$\mathbf{\Phi}_{\text{DM}}^{\text{Bil}}$	$\mathbf{\Phi}_{\text{SH}}^{\text{SF}}$	$\mathbf{\Phi}_{\text{DN}}^{\text{GD}}$
bi,u,me	$\mathbf{\Phi}_{\text{DM}}^{\text{Bil}}$	$\mathbf{\Phi}_{\text{SH}}^{\text{UM}}$	$\mathbf{\Phi}_{\text{DN}}^{\text{MD}}$
bi,u,ga	$\mathbf{\Phi}_{\text{DM}}^{\text{Bil}}$	$\mathbf{\Phi}_{\text{SH}}^{\text{UM}}$	$\mathbf{\Phi}_{\text{DN}}^{\text{GD}}$
me,s,me	$\mathbf{\Phi}_{\text{DM}}^{\text{Men}}$	$\mathbf{\Phi}_{\text{SH}}^{\text{SF}}$	$\mathbf{\Phi}_{\text{DN}}^{\text{MD}}$
me,s,ga	$\mathbf{\Phi}_{\text{DM}}^{\text{Men}}$	$\mathbf{\Phi}_{\text{SH}}^{\text{SF}}$	$\mathbf{\Phi}_{\text{DN}}^{\text{GD}}$
me,u,me	$\mathbf{\Phi}_{\text{DM}}^{\text{Men}}$	$\mathbf{\Phi}_{\text{SH}}^{\text{UM}}$	$\mathbf{\Phi}_{\text{DN}}^{\text{MD}}$
me,u,ga	$\mathbf{\Phi}_{\text{DM}}^{\text{Men}}$	$\mathbf{\Phi}_{\text{SH}}^{\text{UM}}$	$\mathbf{\Phi}_{\text{DN}}^{\text{GD}}$
ma,s,me	$\mathbf{\Phi}_{\text{DM}}^{\text{Mal}}$	$\mathbf{\Phi}_{\text{SH}}^{\text{SF}}$	$\mathbf{\Phi}_{\text{DN}}^{\text{MD}}$
ma,s,ga	$\mathbf{\Phi}_{\text{DM}}^{\text{Mal}}$	$\mathbf{\Phi}_{\text{SH}}^{\text{SF}}$	$\mathbf{\Phi}_{\text{DN}}^{\text{GD}}$
ma,u,me	$\mathbf{\Phi}_{\text{DM}}^{\text{Mal}}$	$\mathbf{\Phi}_{\text{SH}}^{\text{UM}}$	$\mathbf{\Phi}_{\text{DN}}^{\text{MD}}$
ma,u,ga	$\mathbf{\Phi}_{\text{DM}}^{\text{Mal}}$	$\mathbf{\Phi}_{\text{SH}}^{\text{UM}}$	$\mathbf{\Phi}_{\text{DN}}^{\text{GD}}$

Table 3.2: Abbreviations of the twelve configurations of the static data model $\mathbf{\Phi}_{\text{Proc}}^{\text{stat}}$ used in the drift synthesis experiments.

the different sensors and normalized by $2^{16} - 1$. We provide here a precise description of the transformations that we consider in our static model $\mathbf{\Phi}_{\text{Proc}}^{\text{stat}}$, followed by a description how to convert this static model into a differentiable, parametrized model $\mathbf{\Phi}_{\text{Proc}}^{\text{para}}$. For simplicity, if not stated otherwise, writing the equation $v_{c,h,w} = a_{c,h,w} + b_{c,h,w}$ defines $v_{c,h,w}$ for all $1 \leq c \leq 3$, $1 \leq h \leq H$ and $1 \leq h \leq W$.

The static data model $\mathbf{\Phi}_{\text{Proc}}^{\text{stat}}$

Following common steps in ISP, the *static data model* is defined as the composition

$$\mathbf{\Phi}_{\text{Proc}}^{\text{stat}} = \mathbf{\Phi}_{\text{GC}} \circ \mathbf{\Phi}_{\text{DN}} \circ \mathbf{\Phi}_{\text{SH}} \circ \mathbf{\Phi}_{\text{CC}} \circ \mathbf{\Phi}_{\text{WB}} \circ \mathbf{\Phi}_{\text{DM}} \circ \mathbf{\Phi}_{\text{BL}}, \tag{3.5}$$

mapping a raw sensor image to a RGB image. We note that other data model variations, for example by reordering or adding steps, are feasible. The static data models allow the controlled synthesis of different, physically faithful views from the same underlying raw sensor data by manually changing the configurations of the intermediate steps. Fixing the continuous features, but varying $\mathbf{\Phi}_{\text{DM}}$, $\mathbf{\Phi}_{\text{SH}}$ and $\mathbf{\Phi}_{\text{DN}}$ results in twelve different views for the configurations considered here. Samples for each of the twelve data models are provided in Fig. 3.2. An overview of the data model configurations and their corresponding abbreviations can be found alongside processed samples in 3.2. Here, we define the individual functions of the composition $\mathbf{\Phi}_{\text{Proc}}^{\text{stat}}$:

Black level correction (BL) removes thermal noise and readout noise generated from the

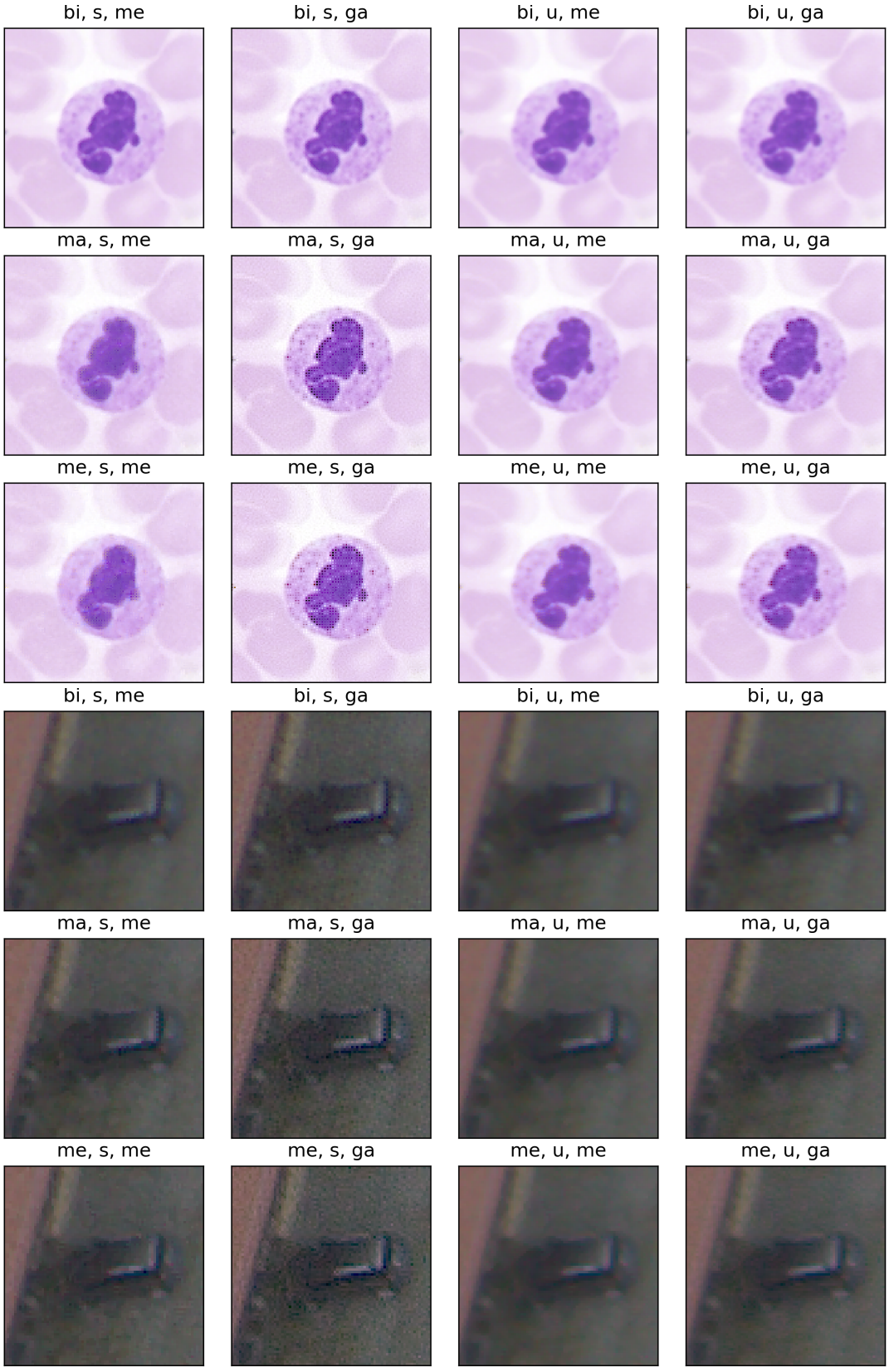

Figure 3.2: Samples for both datasets, Raw-Microscopy and Raw-Drone, from all twelve pipelines used in the drift synthesis experiments. The legend for abbreviations can be found in Table 3.2.

camera sensor. The transformation is given by

$$\boldsymbol{\Phi}_{\mathrm{BL}} : [0,1]^{H,W} \to [0,1]^{H,W}, \boldsymbol{x}_{\mathrm{RAW}} \mapsto \boldsymbol{v}_{\mathrm{BL}}, \tag{3.6}$$

with

$$\begin{aligned}(v_{\mathrm{BL}})_{2h+1,2w+1} &= x_{2h+1,2w+1} - bl_1\\ (v_{\mathrm{BL}})_{2h,2w+1} &= x_{2h,2w+1} - bl_2\\ (v_{\mathrm{BL}})_{2h+1,2w} &= x_{2h+1,2w} - bl_3\\ (v_{\mathrm{BL}})_{2h,2w} &= x_{2h,2w} - bl_4.\end{aligned}$$

By design of $\boldsymbol{bl} \in \mathbb{R}^4$, black level correction ensures that $\boldsymbol{v}_{\mathrm{BL}}$ is again an element of $[0,1]^{H,W}$.

Demosaicing (DM) is applied to reconstruct the full RGB color image through interpolation. We use one out of the three demosaicing algorithms BayerBilinear ($\boldsymbol{\Phi}_{\mathrm{DM}}^{\mathrm{Bil}}$), Menon2007 ($\boldsymbol{\Phi}_{\mathrm{DM}}^{\mathrm{Men}}$) and Malvar2004 ($\boldsymbol{\Phi}_{\mathrm{DM}}^{\mathrm{Mal}}$) from the Python package `color-demosaicing` and denote this transformation by the map

$$\boldsymbol{\Phi}_{\mathrm{DM}} : [0,1]^{H,W} \to [0,1]^{3,H,W}, \boldsymbol{v} \mapsto \boldsymbol{v}_{\mathrm{DM}}. \tag{3.7}$$

White balance (WB) is applied to obtain a neutrally illuminated image. The transformation is given by

$$\boldsymbol{\Phi}_{\mathrm{WB}} : [0,1]^{3,H,W} \to [0,1]^{3,H,W}, \boldsymbol{v} \mapsto \boldsymbol{v}_{\mathrm{WB}}, \tag{3.8}$$

where $\boldsymbol{wb} \in [0,1]^3$ adjusts the intensities by

$$(v_{\mathrm{WB}})_{c,h,w} = wb_c \cdot (v_{\mathrm{DM}})_{c,h,w}. \tag{3.9}$$

Color correction (CC) balances the saturation of the image by considering color dependencies. Let $\boldsymbol{M} \in \mathbb{R}^{3,3}$ be the color matrix. The transformation is defined by

$$\boldsymbol{\Phi}_{\mathrm{CC}} : [0,1]^{3,H,W} \to \mathbb{R}^{3,H,W}, \boldsymbol{v} \mapsto \boldsymbol{v}_{\mathrm{CC}}, \tag{3.10}$$

where

$$\boldsymbol{v}_{\mathrm{CC}} = \begin{bmatrix}(v_{\mathrm{CC}})_{1,h,w}\\ (v_{\mathrm{CC}})_{2,h,w}\\ (v_{\mathrm{CC}})_{3,h,w}\end{bmatrix} = \boldsymbol{M}\begin{bmatrix}(v_{\mathrm{WB}})_{1,h,w}\\ (v_{\mathrm{WB}})_{2,h,w}\\ (v_{\mathrm{WB}})_{3,h,w}\end{bmatrix}. \tag{3.11}$$

The entries of the resulting $\boldsymbol{v}_{\mathrm{CC}}$ are no longer restricted to $[0,1]$.

Sharpening (SH) reduces the blurriness of an image. We use the two methods sharpening filter ($\boldsymbol{\Phi}_{\mathrm{SH}}^{\mathrm{SF}}$) and unsharp masking ($\boldsymbol{\Phi}_{\mathrm{SH}}^{\mathrm{UM}}$) that are applied after a transformation of the view $\boldsymbol{v}_{\mathrm{CC}}$ to

the YUV-color space. To convert the view to the YUV-color space we use the `skimage.color` function `rgb2yuv` ($\mathbf{\Phi}_{YUV}$). The sharpening filter

$$SF : \mathbb{R}^{3,H,W} \to \mathbb{R}^{3,H,W}, \tag{3.12}$$

is defined by a channel-wise convolution

$$(SF(\boldsymbol{v}))_{c,h,w} = \left((\boldsymbol{v}_c \star \boldsymbol{k})_{h,w}\right)_c \quad with \quad \boldsymbol{k} := \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \tag{3.13}$$

of the view

$$\boldsymbol{v} = \mathbf{\Phi}_{YUV}(\boldsymbol{v}_{\mathrm{CC}}). \tag{3.14}$$

For unsharp masking we use the `ski.filters` function `unsharp_mask` modeled by UM. To formally define the sharpening we write

$$\mathbf{\Phi}_{\mathrm{SH}} : \mathbb{R}^{3,H,W} \to \mathbb{R}^{3,H,W}, \boldsymbol{v} \mapsto \boldsymbol{v}_{\mathrm{SH}} \tag{3.15}$$

where

$$\boldsymbol{v}_{\mathrm{SH}} = algo \circ \mathbf{\Phi}_{YUV}(\boldsymbol{v}_{\mathrm{CC}}) \quad with \quad algo \in \{SH, UM\}. \tag{3.16}$$

Denoising (DN) reduces the noise in an image that is (partly) introduced by SH and transforms the YUV-color space view back to the RGB-color space. For the latter transformation, the `skimage.color` function `yuv2rgb` $\left(\mathbf{\Phi}_{YUV}^{-1}\right)$ is used. We apply one out of the two methods Gaussian denoising $\left(\mathbf{\Phi}_{\mathrm{DN}}^{\mathrm{GD}}\right)$ and Median denoising $\left(\mathbf{\Phi}_{\mathrm{DN}}^{\mathrm{GD}}\right)$. For Gaussian denoising, we apply a Gaussian filter (GF) with standard deviation of $\sigma = 0.5$ from the `scipy.ndimage` package. For median denoising we apply a median filter (MF of size 3 from the `scipy.ndimage` package. Formally, this reads as

$$\mathbf{\Phi}_{\mathrm{DN}} : \mathbb{R}^{3,H,W} \to \mathbb{R}^{3,H,W}, \boldsymbol{v} \mapsto \boldsymbol{v}_{\mathrm{DN}} \tag{3.17}$$

where

$$\boldsymbol{v}_{\mathrm{DN}} = \mathbf{\Phi}_{YUV}^{-1} \circ algo(\boldsymbol{v}_{\mathrm{SH}}) \quad with \quad algo \in \{GF, UM\}. \tag{3.18}$$

Gamma correction (GC) equilibrates the overall brightness of the image. First, the entries of the view $\boldsymbol{v}_{\mathrm{DN}}$ are clipped to $[0, 1]$ leading to

$$(v_{CP})_{c,h,w} = \left(v_{\mathrm{DN}}\right)_{c,h,w} \mathbb{1}_{\{0 \le (v_{\mathrm{DN}})_{c,h,w} \le 1\}} + \mathbb{1}_{\{(v_{\mathrm{DN}})_{c,h,w} > 1\}}. \tag{3.19}$$

Second, the brightness adjusting transformation is defined by

$$\boldsymbol{\Phi}_{\mathrm{GC}} : \mathbb{R}^{3,H,W} \to [0,1]^{3,H,W}, \boldsymbol{v} \mapsto \boldsymbol{v}_{\mathrm{GC}} = \left(\boldsymbol{v}_{CP}\right)^{\frac{1}{\gamma}} \tag{3.20}$$

for some $\gamma > 0$ applied element-wise. Note that zero-clipping is necessary for $\boldsymbol{v}_{\mathrm{GC}}$ to be well-defined.

The parametrized data model $\boldsymbol{\Phi}^{\mathbf{para}}_{\mathbf{Proc}}$

For a fixed raw sensor image, the *parametrized data model* $\boldsymbol{\Phi}^{\mathrm{para}}_{\mathrm{Proc}}$ maps from a parameter space Θ to a RGB image. It is similar to the static data model with the notable difference that each processing step is differentiable wrt. its parameters $\boldsymbol{\theta}$. This allows for backpropagation of the gradient from the output of the task model $\boldsymbol{\Phi}_{\mathrm{Task}}$ through the data model $\boldsymbol{\Phi}_{\mathrm{Proc}}$ all the way back to the raw sensor image $\boldsymbol{x}_{\mathrm{RAW}}$ to perform drift forensics and drift adjustments. Hence, we aim to design a data model $\boldsymbol{\Phi}^{\mathrm{para}}_{\mathrm{Proc}} : \mathbb{R}^{H,W} \times \Theta \to \mathbb{R}^{C,H,W}$ that is differentiable in $\boldsymbol{\theta} \in \Theta$ satisfying

$$\boldsymbol{\Phi}^{\mathrm{stat}}_{\mathrm{Proc}} = \boldsymbol{\Phi}^{\mathrm{para}}_{\mathrm{Proc}}\left(\cdot, \boldsymbol{\theta}^{\mathrm{stat}}\right) \tag{3.21}$$

for some choice of parameters $\boldsymbol{\theta}^{\mathrm{stat}}$ and some fixed configuration of the static pipeline $\boldsymbol{\Phi}^{\mathrm{stat}}_{\mathrm{Proc}}$. We define for $\boldsymbol{\theta} = \left(\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_7\right) \in \Theta$ the parametrized processing model

$$\boldsymbol{\Phi}^{\mathrm{para}}_{\boldsymbol{Proc}} : [0,1]^{3,H,W} \times \Theta \to [0,1]^{3,H,W}, (\boldsymbol{x}_{\mathrm{RAW}}, \boldsymbol{\theta}) \mapsto \boldsymbol{v} \tag{3.22}$$

by the composition

$$\boldsymbol{v} = \left(\boldsymbol{\Phi}^{\mathrm{para}}_{\mathrm{GC}}\left(\cdot, \boldsymbol{\theta_7}\right) \circ \boldsymbol{\Phi}^{\mathrm{para}}_{\mathrm{DN}}\left(\cdot, \boldsymbol{\theta_6}\right) \circ \boldsymbol{\Phi}^{\mathrm{para}}_{\mathrm{SH}}\left(\cdot, \boldsymbol{\theta_5}\right) \circ \boldsymbol{\Phi}^{\mathrm{para}}_{\mathrm{CC}}\left(\cdot, \boldsymbol{\theta_4}\right) \circ \boldsymbol{\Phi}^{\mathrm{para}}_{\mathrm{WB}}\left(\cdot, \boldsymbol{\theta_3}\right) \circ \boldsymbol{\Phi}^{\mathrm{para}}_{\mathrm{DM}}\left(\cdot, \boldsymbol{\theta_2}\right) \circ \boldsymbol{\Phi}^{\mathrm{para}}_{\mathrm{BL}}\left(\cdot, \boldsymbol{\theta_1}\right)\right)\left(\boldsymbol{x}_{\mathrm{RAW}}\right). \tag{3.23}$$

The operations used above are differentiable except for the clipping operation in the GC that is *a.e.*-differentiable[5], since the set $\{0,1\}$ of non-differentiable points has measure zero. Assuming in addition that $\mathbb{P}\left((v_{\mathrm{DN}})_{c,h,w} \in \{0,1\}\right) = 0$ holds true for the entries of $\boldsymbol{v}_{\mathrm{DN}}$ results in an *a.e.*-differentiable processing model. We further say that $\boldsymbol{\Phi}^{\mathrm{para}}_{\mathrm{Proc}}$ is differentiable, noting that this holds only *a.e.* under the aforementioned assumption. We designed the individual functional components of eq. 3.23 as follows

Black level correction (BL) For the parametrized black level correction define the map

$$\boldsymbol{\Phi}^{\mathrm{stat}}_{\mathrm{BL}} : [0,1]^{H,W} \times \mathbb{R}^4 \to \mathbb{R}^{H,W}, (\boldsymbol{x}_{\mathrm{RAW}}, \boldsymbol{\theta}_1) \mapsto \boldsymbol{v}_{\mathrm{BL}} = \boldsymbol{\Phi}_{\mathrm{BL}}(\boldsymbol{x}_{\mathrm{RAW}})|_{\boldsymbol{bl}=\boldsymbol{\theta}_1}. \tag{3.24}$$

and set $\Theta_1 := \mathbb{R}^4$.

[5] *a.e.* stands for almost everywhere

Demosaicing (DM) We first convert $\boldsymbol{v}_{\mathrm{BL}}$ to a three channel image $[\boldsymbol{R}, \boldsymbol{G}, \boldsymbol{B}] \in \mathbb{R}^{3,H,W}$ where the entries of $\boldsymbol{R}, \boldsymbol{G}$ and $\boldsymbol{B}$ are zero except

$$\begin{aligned} R_{2h+1,2w+1} &= \boldsymbol{v}_{BL_{2h+1,2w+1}}, \quad B_{2h,2w} = v_{BL_{2h,2w}}, \\ G_{2h+1,2w} &= v_{BL_{2h+1,2w}}, \quad G_{2h,2w+1} = v_{BL_{2h,2w+1}}. \end{aligned}$$

To parametrize $\boldsymbol{\Phi}_{\mathrm{DM}}^{\mathrm{Bil}}$ define the map

$$\boldsymbol{\Phi}_{\mathrm{DM}}^{\mathrm{para}} : [0,1]^{H,W} \times \mathbb{R}^{3,3,3} \to \mathbb{R}^{3,H,W}, (\boldsymbol{v}_{\mathrm{BL}}, \boldsymbol{\theta}_2) \mapsto \boldsymbol{v}_{\mathrm{DM}}, \tag{3.25}$$

with $\boldsymbol{\theta}_2 = [\boldsymbol{k}_1, \boldsymbol{k}_2, \boldsymbol{k}_3]$, where the kernels $\boldsymbol{k}_1, \boldsymbol{k}_2, \boldsymbol{k}_3 \in \mathbb{R}^{3,3}$ are separately applied to each color channel resulting in

$$\begin{aligned} \boldsymbol{v}_{DM_{1,h,w}} &= \left(R \star \boldsymbol{k}_1\right)_{h,w} \\ \boldsymbol{v}_{DM_{2,h,w}} &= \left(G \star \boldsymbol{k}_2\right)_{h,w} \\ \boldsymbol{v}_{DM_{3,h,w}} &= \left(B \star \boldsymbol{k}_3\right)_{h,w}. \end{aligned}$$

The source code of BayerBilinear shows that the parameter choice

$$\boldsymbol{k}_1 = \boldsymbol{k}_3 = \begin{bmatrix} 0 & 0.25 & 0 \\ 0.25 & 1 & 0.25 \\ 0 & 0.25 & 0 \end{bmatrix} \quad \textit{and} \quad \boldsymbol{k}_2 = \begin{bmatrix} 0.25 & 0.5 & 0.25 \\ 0.5 & 1 & 0.5 \\ 0.25 & 0.5 & 0.25 \end{bmatrix} \tag{3.26}$$

leads to

$$\boldsymbol{\Phi}_{\mathrm{DM}}^{\mathrm{Bil}} = \boldsymbol{\Phi}_{\mathrm{DM}}^{\mathrm{para}}(\cdot, \boldsymbol{\theta}_2). \tag{3.27}$$

Towards the definition of the parameter space set $\Theta_2 := \mathbb{R}^{3,3,3} \times \Theta_1$.

White balance (WB) For the parametrized white balance define the map

$$\boldsymbol{\Phi}_{\mathrm{WB}}^{\mathrm{para}} : \mathbb{R}^{3,H,W} \times \mathbb{R}^3 \to \mathbb{R}^{3,H,W}, (\boldsymbol{v}_{\mathrm{DM}}, \boldsymbol{\theta}_3) \mapsto \boldsymbol{v}_{\mathrm{WB}} = \boldsymbol{\Phi}_{\mathrm{WB}}(\boldsymbol{v}_{\mathrm{DM}})|_{\boldsymbol{wb}=\boldsymbol{\theta}_3} \tag{3.28}$$

and set $\Theta_3 := \mathbb{R}^3 \times \Theta_2$.

Color correction (CC) For the parametrized color correction define the map

$$\boldsymbol{\Phi}_{\mathrm{CC}}^{\mathrm{para}} : \mathbb{R}^{3,H,W} \times \mathbb{R}^{3,3} \to \mathbb{R}^{3,H,W}, (\boldsymbol{v}_{\mathrm{WB}}, \boldsymbol{\theta}_4) \mapsto \boldsymbol{v}_{\mathrm{CC}} = \boldsymbol{\Phi}_{\mathrm{CC}}(v_{\mathrm{WB}})|_{\boldsymbol{M}=\boldsymbol{\theta}_4} \tag{3.29}$$

and set $\Theta_4 := \mathbb{R}^{3,3} \times \Theta_3$

Sharpening (SH) We parametrize the sharpening filter configuration of the static pipeline, by

	Classification	**Segmentation**
$\mathbf{\Phi}_{\text{Task}}$	ResNet18 based on [74] trained with Adam [99] for 100 epochs learning rate: 10^{-4} mini-batch size: 128	U-Net++ based on [184] trained with Adam for 100 epochs learning rate: $7.5 \cdot 10^{-5}$ mini-batch size: 12

Table 3.3: Summary of the training procedure for both task models.

using the entries of $\boldsymbol{k} \in \mathbb{R}^{3,3}$ defined in (3.13) as parameters leading to

$$\mathbf{\Phi}_{\text{SH}}^{\text{para}} : \mathbb{R}^{3,H,W} \times \mathbb{R}^{3,3} \to \mathbb{R}^{3,H,W}, (\boldsymbol{v}_{\text{CC}}, \boldsymbol{\theta}_5) \mapsto \boldsymbol{v}_{\text{SH}} = \mathbf{\Phi}_{\text{SH}}(v_{\text{CC}})|_{\boldsymbol{k}=\boldsymbol{\theta}_5} \tag{3.30}$$

and $\Theta_5 := \mathbb{R}^{3,3} \times \theta_4$.

Denoising (DN) We parametrize the configuration where the Gaussian denoising method is applied. Applying the Gaussian filter from `scipy.ndimage` with $\sigma = 0.5$ is equivalent to a convolution of the view in the YUV-color space with a specific $\boldsymbol{k}_{gauss} \in \mathbb{R}^{5,5}$. For the specific values of $\boldsymbol{k}_{gauss}$ see `K_BLUR` at the code of the parametrized pipeline. Therefore, to parametrize DN we define the map

$$\mathbf{\Phi}_{\text{DN}}^{\text{para}} : \mathbb{R}^{3,H,W} \times \mathbb{R}^{5,5} \to \mathbb{R}^{3,H,W}, (\boldsymbol{v}_{\text{SH}}, \boldsymbol{\theta}_6) \mapsto \boldsymbol{v}_{\text{DN}} = \mathbf{\Phi}_{\text{DN}}(v_{\text{SH}})|_{\boldsymbol{k}_{gauss}=\boldsymbol{\theta}_6} \tag{3.31}$$

and set $\Theta_6 := \mathbb{R}^{5,5} \times \Theta_5$.

Gamma correction (GC) Define the parametrized gamma correction by

$$\mathbf{\Phi}_{\text{GC}}^{\text{para}} : \mathbb{R}^{3,H,W} \times \mathbb{R} \to [0,1]^{3,H,W}, (\boldsymbol{v}_{\text{DN}}, \boldsymbol{\theta_7}) \mapsto \boldsymbol{v} = \boldsymbol{v}_{\text{GC}} = \mathbf{\Phi}_{\text{GC}}(\boldsymbol{v}_{\text{DN}})|_{\gamma=\theta_7}. \tag{3.32}$$

3.2.3 Task models $\mathbf{\Phi}_{\text{Task}}$

Finally, two task models are employed in the experiments. For the classification task on the Raw-Microscopy dataset a 18-layer residual net (ResNet18) [74] was used as reference task model. This model is designed to classify images from ImageNet [190] and has therefore an output dimension of 1000. In order to use the model to classify images from Raw-Microscopy, we changed the output dimension of the fully-connected layer to nine. The model was trained for 100 epochs using pre-trained ResNet features. Hyperparameters were kept constant across all runs to isolate the effect of varying image processing pipelines. For implementation, the code provided at `https://pytorch.org/hub/pytorch_vision_resnet/` was used. The model consists of 34 layers with approximately 11.2 million trainable parameters. The storage size of the model is 44.725 MB.

To segment cars from the Raw-Drone dataset the convolutional neural network proposed in [184] (U-Net) was used. The model was trained for 100 epochs using pretrained ResNet features as the encoder of the U-Net++. Hyperparameters were kept constant across all runs to isolate the effect of varying image processing pipelines. For implementation, we used the code provided at `https://github.com/qubvel/segmentation_models.pytorch`. The model has approximately 26.1 million trainable parameters. The storage size of the model is 104.315 MB.

Both task models were trained using common data augmentations on processed views $\boldsymbol{v}$ of the image measurements to avoid naive robustness failures. For a summary of the training procedure see 3.3.

3.2.4 Raw dataset acquisition

In order to obtain advanced data models for images, raw sensor data is required. In many industry domains such as microscopy, biomedicine, autonomous vehicles or remote sensing raw sensor data is processed at scale for machine vision tasks. Most existing digital camera systems on the market today, including consumer smartphones, can be configured to access the raw sensor measurements. Next, we explain how to obtain raw sensor data from existing optical hardware. We collected two datasets for two representative machine learning tasks. Both datasets are made available for free, public use at `https://zenodo.org/record/5235536`. As public, scientifically calibrated and labelled raw data is, to the best of our knowledge, currently not available, we acquired two raw datasets as part of this study: Raw-Microscopy and Raw-Drone. Raw-Microscopy consists of expert annotated blood smear microscope images. Raw-Drone comprises drone images with annotations of cars. Our motivation behind the acquisition of these particular datasets was threefold. First, we wanted to ensure that the acquired datasets provide good coverage of representative machine learning tasks, including classification (Raw-Microscopy) and regression (Raw-Drone). Second, we wanted to collect data on applications that, to our minds, are disposed towards positive welfare impact in today's world, including medicine (Raw-Microscopy) and environmental surveying (Raw-Drone). Third, we wanted to ensure the downstream machine learning application contexts are such where errors can be costly, here patient safety (Raw-Microscopy) and autonomous vehicles (Raw-Drone), hence necessitating extensive robustness and dataset drift controls.

Since data collection is an expensive project in and of itself we did not aspire to provide extensive benchmark datasets for the respective applications, but to collect enough data to demonstrate the advanced data modelling and dataset drift controls that raw data enables. In 3.2.5 we provide detailed information on the two datasets and the calibration setups of the acquisition process. Samples of both datasets can be inspected in Fig. 3.3 and 3.2.5.

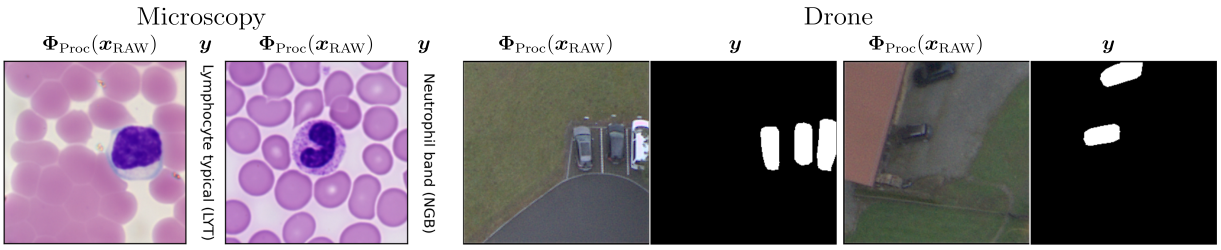

Figure 3.3: Processed samples and labels of the two datasets, Raw-Microscopy (columns one to four) and Raw-Drone (columns five and eight), that were acquired for the dataset drift study presented here.

RGB to raw An alternative goal is to attempt to reconstruct raw images from processed images [22, 150]. As we laid out earlier, when an image is captured by a digital camera, the sensor records raw image data in the form of an array of pixel values. This raw image data is usually processed by an ISP before being used in a downstream task. The ISP performs various adjustments such as color correction, noise reduction, and sharpening to enhance the quality of the image. However, these adjustments are not physically faithful to the original raw image data and result in a loss of information. Therefore, it is generally not possible to reconstruct the exact raw image data from the ISP processed images. While the processed images may look better to the human eye, they do not accurately represent the physical reality of the original scene. For example, a recent paper by Nam et al. [150] propose a content-aware metadata approach to sRGB-to-Raw RGB de-rendering, but acknowledges that the resulting approximations are not perfectly accurate and still suffer from limitations. Similarly, another study by Brooks et al. [22] presents a method for recovering raw data from processed images, but also notes that the approach is not able to recover all of the original data with perfect accuracy. These findings highlight the fundamental challenge of reconstructing raw data from processed images. Empirical approximations are possible but not exact, that is not physically faithful, and hence orthogonal to our goal here. However, one should note that these reconstruction approaches can offer interesting value propositions outside the physically precise drift regime. The proposed technique by [22] "unprocesses" images and offers interesting gains during training, enabling a convolutional neural network with 14-38% lower error rates and 9-18× faster performance, while generalizing to other sensors as well. This approach can further be calibrated using joint learning of sampling and reconstruction, offering better raw reconstructions by adapting to image content, with an additional online fine-tuning strategy for enhanced results [150].

3.2.5 Datasets details

In the following, core information on the two acquired datasets is provided.

Raw-Microscopy Assessment of blood smears under a light microscope is a key diagnostic technique for many healthcare services such as cancer treatment and kidney failure as well as blood disorder detection [10]. The creation of image datasets and machine learning models on them has received wide interest in recent years [112, 143, 8]. Variations in the image processing can affect the downstream task model performance [229]. Dataset drift controls can thus help to specify the perimeter of safe application for a task model. A raw dataset was collected for that purpose. A bright-field microscope was used to image blood smear cytopathology samples. The light source is a halogen lamp equipped with a 0.55 NA condenser, and a pre-centred field diaphragm unit. Filters at 450 nm, 525 nm and 620 nm were used to acquire the blue, green and red channels respectively. The condenser is followed by a 40× objective with 0.95 NA (Olympus UPLXAPO40X). Slides can be moved via a piezo with 1 nm spatial resolution, in three directions. Focus was achieved by maximizing the variance of the pixel values[6]. Images are acquired at 16 bit, with a 2560 × 2160 pixels CMOS sensor (PCO edge 5.5). The point-spread function (PSF) was measured to be 450 nm with 100 nm nanospheres. Mechanical drift was measured at 0.4 pixels per hour. Imaging was performed on de-identified human blood smear slides (Ma190c Lieder, J. Lieder GmbH & Co. KG, Ludwigsburg/Germany). All slides were taken from healthy humans without known hematologic pathology. Imaging regions were selected to contain single leukoytes in order to allow unique labelling of image patches, and regions were cropped to 256 × 256 pixels. All images were annotated by a trained hematological cytologist using the standard scheme of normal leukocytes comprising band and segmented neutrophils, typical and atypical lymphocytes, monocytes, eosinophils and basophils [131]. To soften class imbalance, candidates for rare normal leukocyte types were preferentially imaged, and enrich rare classes. Additionally, two classes for debris and smudge cells, as well as cells of unclear morphology were included. Labelling took place for all imaged cells from a particular smear at a time, with single-cell patches shown in random order. Raw images were extracted using JetRaw Data Suite features. Blue, red and green channels are metrologically rescaled independently in intensity to simulate a standard RGB camera condition. Some pixels are discarded complementary on each channel in order to obtain a Bayer filter pattern.

Raw-Microscopy for segmentation comes with 940 raw images, twelve differently processed variants totalling 11280 images and six additional raw intensity levels totalling 5640 samples.

Raw-Drone Automated processing of drone data has useful applications including precision agriculture [109] or environmental protection [92]. Variation in image processing has been shown to affect task model performance [139, 243], underlining the need for drift controls. For the purposes of this study, a raw car segmentation dataset was created for the drone image modality. A DJI Mavic 2 Pro Drone was used, equipped with a Hasselblad L1D-20c camera (Sony IMX183

[6] 3.4 in 3.2.5 provides an illustration of the imaging setup.

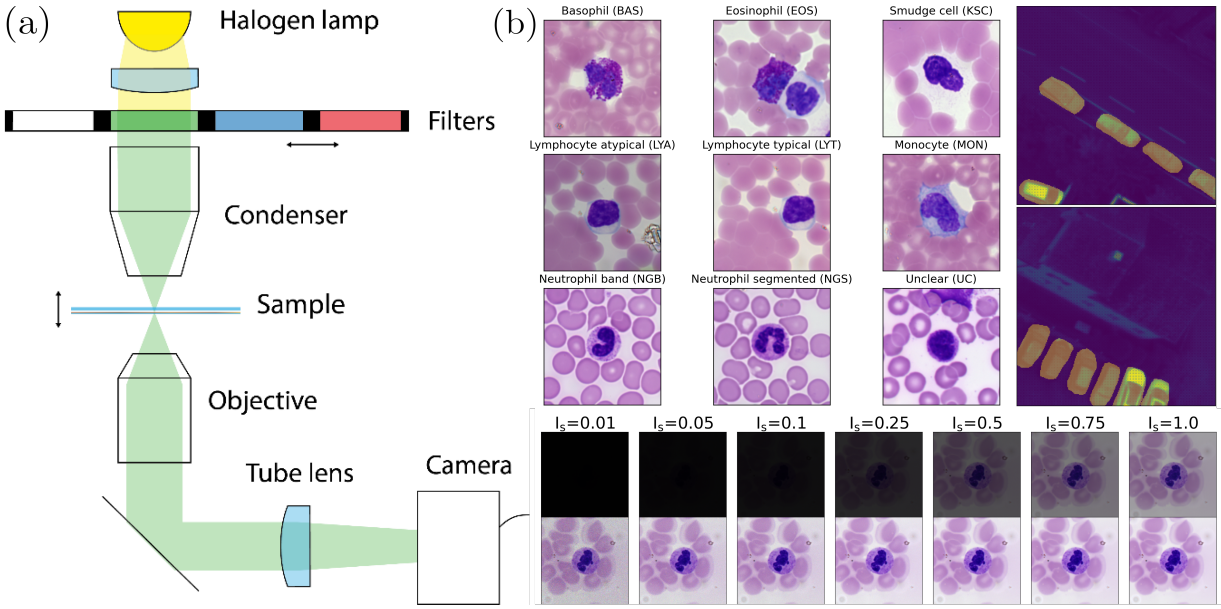

Figure 3.4: (a) An illustration of the imaging setup. (b) Datasets visualization. (Top-left) Processed raw microscopy classes are shown. (Top-right) Drone raw images are shown with the segmentation masks applied over them. (Bottom) Different intensity realizations are shown for the microscopy case. Images on the top are directly print out on the same scale as the original image. Images in the bottom row are normalized on their own min and max values to highlight the role of noise levels on low-intensity images.

sensor) having 2.4 µm pixels in a Bayer filter array. The lens has a focal length of 10.3 mm. The f-number was set to $N = 8$, to emulate the PSF circle diameter relative to the pixel pitch and ground sampling distance (GSD) as would be found on images from high-resolution satellites. The PSF was measured to have a circle diameter of 12.5 µm. This corresponds to a diffraction-limited system, within the uncertainty dominated by the wavelength spread of the image. Images were taken at 200 ISO, a gain of 0.528 DN/e^-. The 12-bit pixel values are however left-justified to 16-bits, so that the gain on the 16-bit numbers is 8.448 DN/e^-. The images were taken at a height of 250 m, so that the GSD is 6 cm. All images were tiled in 256×256 patches. Segmentation masks were created to identify cars for each patch. From this mask, classification labels were generated to detect if there is a car in the image. The dataset is constituted of 548 images for the segmentation task.

Raw-Drone for segmentation comes with 548 raw images, twelve differently processed variants totalling 6576 images and six additional raw intensity levels totalling 3288 samples.

Composition of Raw-Microscopy	
Type of instances	Image and label
Objects on images	White blood cells
Type of classes	Morphological classes
Number of instances	940
Number of classes	9
Image size	256 by 256 pixels
Image format	`.tif`
Raw image format	`.tif`

Class	Proportion in %
Basophil (BAS)	1.91
Eosinophil (EOS)	5.74
Smudge cell / debris (KSC)	17.34
atypical Lymphocyte (LYA)	3.19
typical Lymphocyte (LYT)	24.47
Monocyte (MON)	20.32
Neutrophil (band) (NGB)	0.85
Neutrophil (segmented) (NGS)	22.98
Image that could not be assigned a class (UNC)	3.19

Composition of Raw-Drone	
Type of instances	Image and mask
Objects on images	Landscape shots from above
Number of instances	548
Number of original images	12
Image size	256 by 256 pixels
Mask size	256 by 256 pixels
Original image size	3648 by 5472
Image format	`.tif`
Mask format	`.png`
Raw image format	`.DNG`

Table 3.4: Summaries of the compositions of Raw-Microscopy and Raw-Drone

3.3 Applications

With data models, raw data and task models in place, we are now able to demonstrate the advanced dataset drift controls comprising ① drift synthesis, ② modular drift forensics and ③ drift optimization.

3.3.1 Drift synthesis

The static data model enables physically faithful synthesis of drift test cases: individual data model components can be swapped out, allowing the controlled creation of different, physically faithful processed views from one raw reference dataset. A typical use case of drift synthesis for machine learning researchers and practitioners is the prospective validation of their task model to drift from different camera devices, for example, microscopes across different labs, without having to collect measurements from the different devices. This simulation can be done in-silico as software because the hardware specific processing that takes place on optical measurement devices after the sensor reading is also in-silico. Thus, the extraction of raw sensor readings from one device allows the emulation of different processing variations present on other devices. A typical workflow of this data synthesis starts with an engineer constructing a data model of interest, then passing raw measurements through it and finally getting emulated data to test how the downstream task model would fare on processing variations from different devices. We provide twelve possible example data models in the following experiments. For each of the twelve data models laid out in 3.2, the task models were trained for 100 epochs on image data processed through the training data model. Hyperparameters were kept constant across all runs to isolate the effect of varying the data models. Then, dataset drift test cases were synthesized by processing the raw test data through the remaining eleven data models. The task models were then evaluated on test data from all twelve data models. All results that follow are reported as the mean with error bars over a 5-fold cross-validation. You can find a full description of task model hyperparameters and experimental setup in Appendix A.1. The metrics used to evaluate the task models are accuracy for classification and IoU for segmentation.

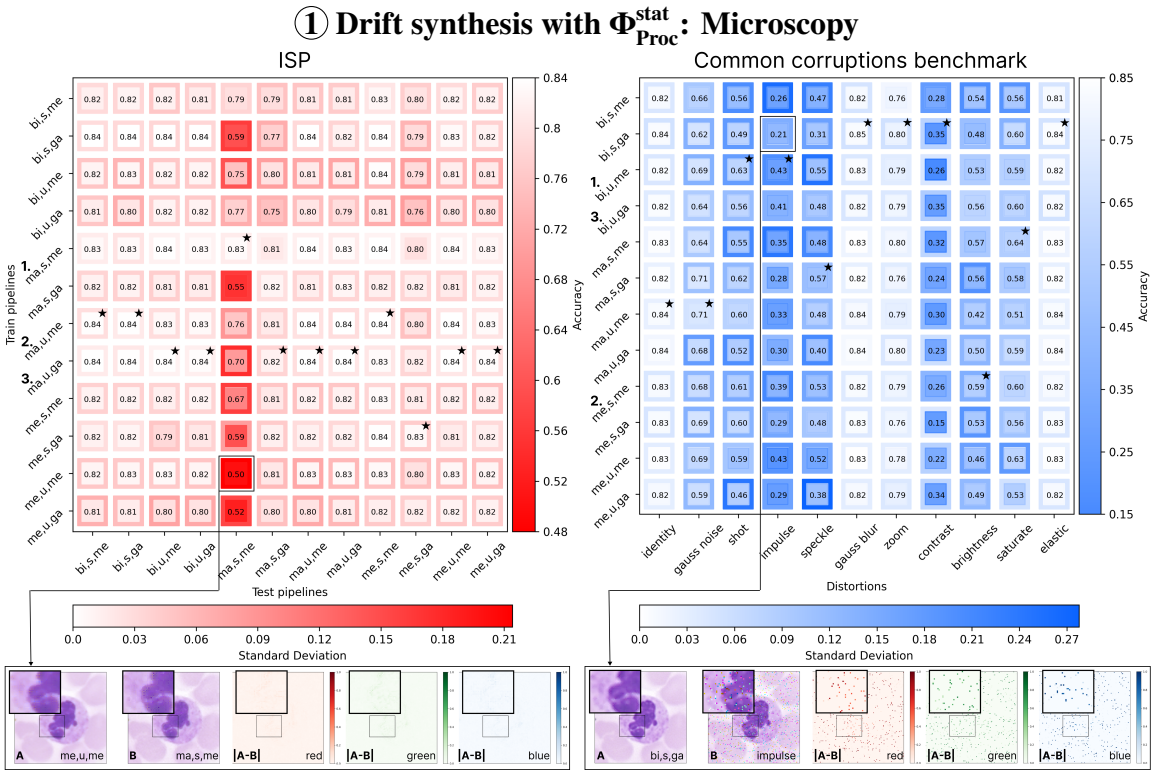

Figure 3.5: 5-fold cross-validation results of the Raw-Microscopy drift synthesis experiments. Each cell contains the average accuracy with a color coded border for the standard deviation. Task models were trained on the data models on the vertical axis and then tested on processed data as indicated on the horizontal axis. Numbers 1-3 left to the vertical axis denote the ranking of task models according to their average accuracy across all test pipelines (respective corruptions). Stars denote the train pipeline under which the task model performed best on the respective test pipeline/corruption. Full ranking results can be found in Tables A.1 to A.3 of Appendix A.2. Top-left: Varying the data model leads to mild performance drops except (ma,s,me). Diagonal is $\mathbf{\Phi}_{\text{Proc}} = \tilde{\mathbf{\Phi}}_{\text{Proc}}$. Top-Right: Comparison to the corruption benchmark at medium severity (level 3). The average performance drop is more than thirteen times higher compared to data model variations. First column is $\mathbf{\Phi}_{\text{Proc}} = \tilde{\mathbf{\Phi}}_{\text{Proc}}$. Bottom: Visual inspection of worst case (globally worst scoring) train/test pipelines.

Physically faithful versus physically unfaithful robustness validation

Physically faithful The leukocyte classification model, as displayed in the left matrix of Figure 3.5, has a critical drop for few configurations, suggesting that it is relatively robust to processing induced dataset drift except for the (ma,s,me) configuration. Note that diagonal elements serve as references corresponding to test data that was processed in the same way as the training data. The segmentation task model (left matrix in Figure 3.6) displays a more heterogeneous pattern with symmetries for certain combinations of data models, such as (bi, u, me/ga) and (me, s, me/ga), which are mutually destructive to the task model performance. The average performance of the task models drops from 0.82 to 0.8 between train and test data models for classification and from 0.71 to 0.65 for segmentation. That is the average change from train to test data environment

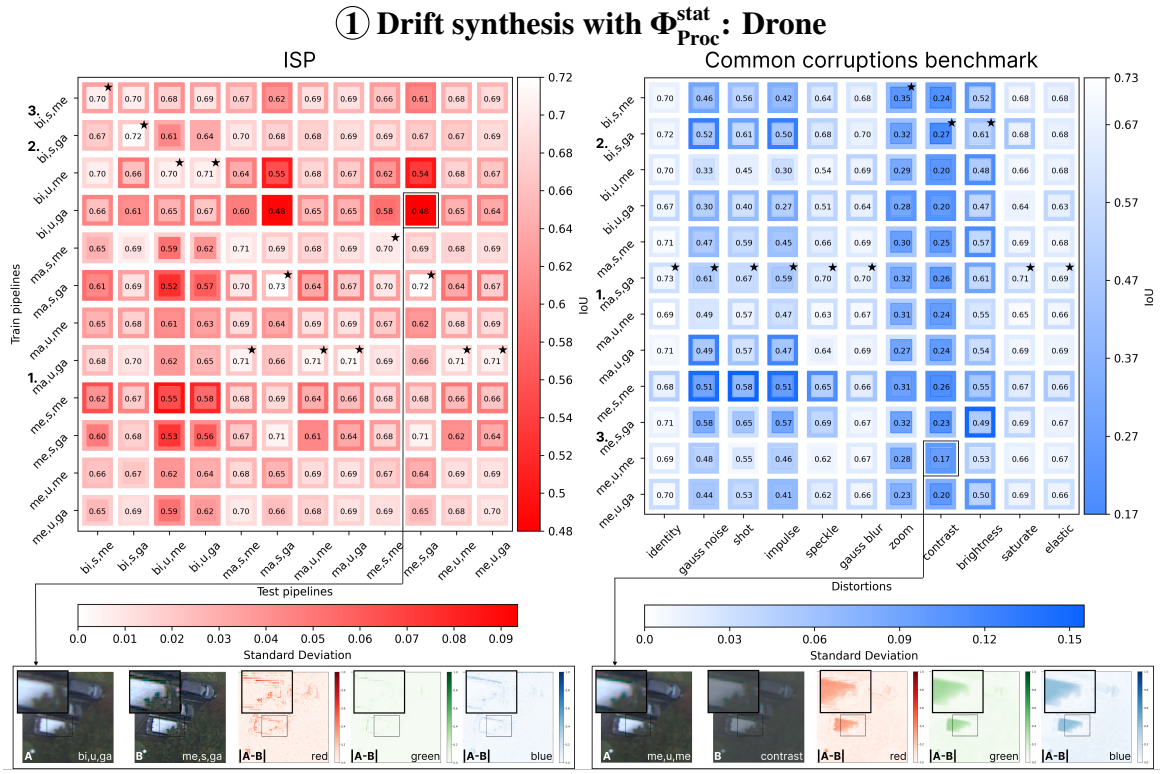

Figure 3.6: 5-fold cross-validation results of the Raw-Drone drift synthesis experiments. Each cell contains the average IoU with a color coded border for the standard deviation. Task models were trained on the data model on the vertical axis and then tested on processed data as indicated on the horizontal axis. Numbers 1-3 left to the vertical axis denote the ranking of task models according to their average IoU across all test pipelines respective corruptions. Stars denote the train pipeline under which the task model performed best on the respective test pipeline/corruption. Full ranking results can be found in Tables A.1, A.4 and A.5 of Appendix A.2. Left: Varying the data model leads to mixed performance drops. Diagonal is $\mathbf{\Phi}_{\text{Proc}} = \tilde{\mathbf{\Phi}}_{\text{Proc}}$. Right: Comparison to the corruption benchmark at medium severity (level 3). The average performance drop is more than four times higher compared to data model variations. First column is $\mathbf{\Phi}_{\text{Proc}} = \tilde{\mathbf{\Phi}}_{\text{Proc}}$. Bottom: Visual inspection of worst case (globally worst scoring) train/test pipelines.

calculated across all configurations for ISP as well as Common Corruptions. The results for individual components of the data models can also be directly compared in Figures 3.5 and 3.6. For example, to understand how changes in the demosaicing algorithm affect the segmentation model, we can look at the left box in Figure 3.6 and focus on the column combinations 1-5-9, 2-6-10, 3-7-11 and 4-8-12 where the demosaicing is varied but the other components of the data model stay fixed. Considering the training condition with the (bi,s,me) data model using Bilinear demosaicing (row 1), the task model performance drops from 0.7 (column 1) to 0.67 (column 5) IoU in response to Malvar2004 demosaicing and to 0.66 (column 9) IoU when using Menon2007.

Physically unfaithful To demonstrate the limitation of post-hoc augmentations, we compare the drift synthesis results to a popular augmentation testing framework known as Common Corruptions

Benchmark [76]. For this use case, We are only referring to limitations relating to robustness testing and model selection. Augmentations have important empirically validated benefits in other applications such as regularization during training or semi- and self-supervised learning. In machine learning practice, augmentation users often assume that applying a corruption, for example 'blur', to a processed image will emulate the noise from a real-world camera system, for example blur from the lens or the denoising component in the camera. However, this is not the case. It should not come as a surprise given the composition of the optical data generating process (see Figure 3.1 and Section 3.2), that is $\boldsymbol{v} + \boldsymbol{\xi} \in \tilde{\boldsymbol{\Phi}}_{\text{Proc}} \left[\boldsymbol{\mathcal{X}}_{\text{RAW}} \right]$ might not hold true for any data model as we explain in Section 3.1.1. This has also been empirically demonstrated in previous work [228, 243, 90]. Here we go one step further to show that physically *un*faithful augmentation testing can lead to wrong conclusions in model selection. As we note in Appendix A.2, a direct apple-to-apple comparison is impossible due to the fundamental limitation of post-hoc corruptions' physical unfaithfulness. However, we make the comparison as plausible as possible by only including corruptions that can be related to the ISP data model. Others, such as Fog, Spatter, Motion, Snow, Frost were excluded. A comparative overview of included and excluded corruptions can be found in Figure 3.7. In contrast to physically faithful test data, the performance drops under corruptions are more severe across the board: from 0.82 to 0.55 for classification and from 0.71 to 0.49 for segmentation. Results at additional severity levels for the common corruptions can be found in Appendix A.2. This is more than thirteen and four times as much as for the physically faithful drifts synthesized with the data models considered here. We see similar gaps when considering the best models. For example, the best performing microscopy data-task-model combo selected across all test ISPs ((ma,s,me) with 0.83 average accuracy) has more than 20 percentage points gap compared to Common Corruptions (0.62 average accuracy). For the segmentation task we make a similar observation where the best performing drone data-task-model combo selected across all test ISPs ((ma,u,ga) with 0.68 average IoU) has more than 10 percentage points gap compared to Common Corruptions (0.55 average IoU). See Appendix A.2 for full tables.

Qualitative comparison The qualitative difference between physically faithful drift test cases and augmentation testing can also be appreciated in the samples of the bottom rows of Figures 3.5 and 3.6. For each task, we display a sample from the drift test configuration with the worst case performance drop between train and test data conditions. We show the sample viewed from the training data model (**A**), the test data model (**B**), and the difference between both (|**A-B**|) along the red, green and blue channel. For both tasks, the drift artifacts (|**A-B**|) are more localized than the artifacts obtained from augmentation testing. This makes sense, as changes in the composition of the test data model $\boldsymbol{\Phi}_{\text{Proc}}$ maintain the physical faithfulness of the remaining data model, whereas augmentation testing spreads noise globally across all pixels, which are not guaranteed to be physically faithful.

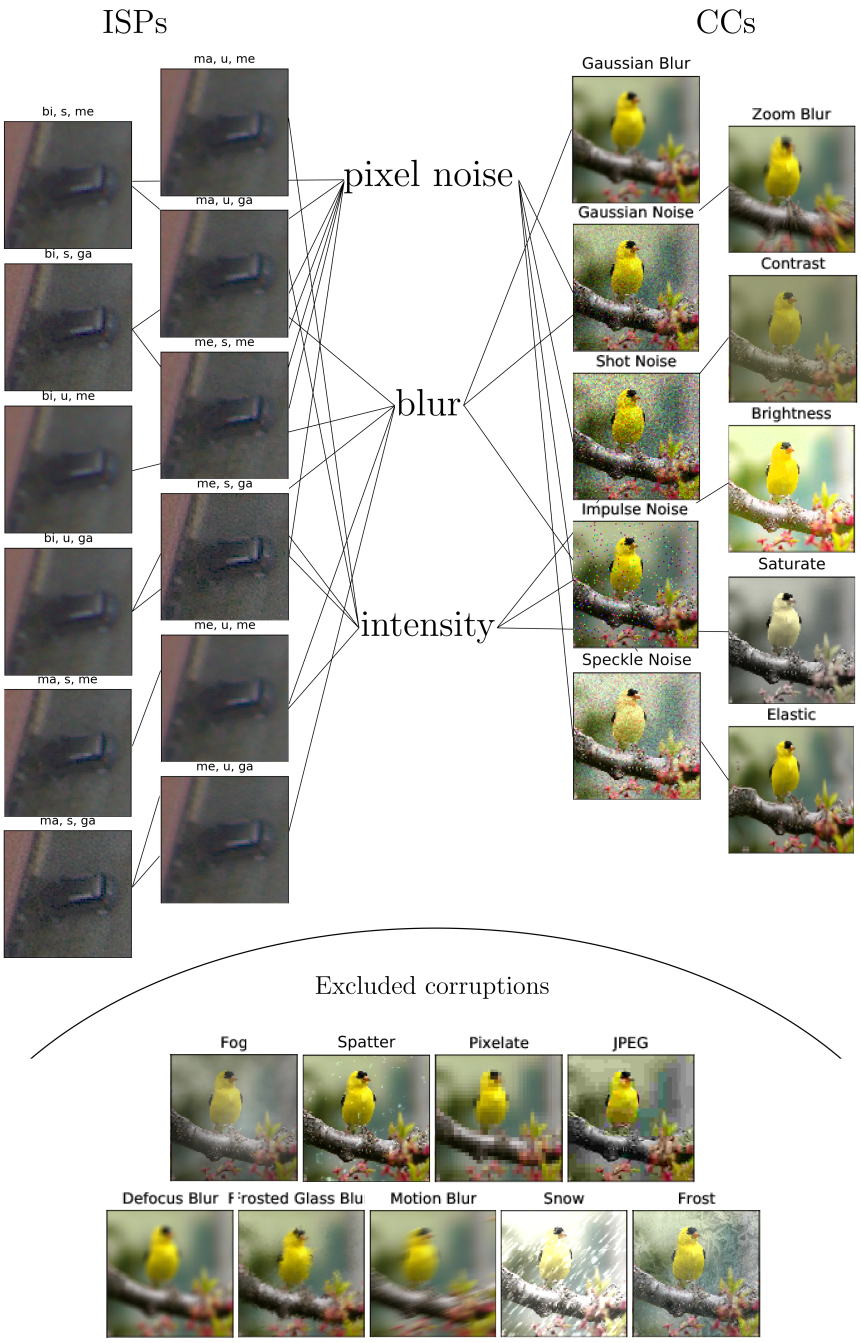

Figure 3.7: A comparative overview of physically faithful data models (ISPs, top-left) and Common Corruptions (CC, top-right) used in the drift synthesis experiments of 3.3.1. A matching heuristic based on possible visual perception of the drift artifacts (top-middle) is provided for readers who would like to relate specific data models to specific corruptions. However, we emphasize that this is a *purely qualitative heuristic* and has no metrological basis. Since CCs are not physically faithful it is not clear how to relate them to actual variations in the optical data generating process. Finally, corruptions that were excluded from the experiments in Section 3.3.1 are displayed (bottom). The CC examples where stitched from the original paper [75] for authenticity.

Implications for model selection

Similarly, the conclusions for model selection diverge depending on whether physically faithful data or corruptions are used. In terms of the average performance across all test conditions, none of the top-3 training data models, denoted by the numbers 1-3 alongside the rows of the matrices in Figures 3.5 and 3.6, overlap between ISP and common corruptions on the classification task. For segmentation, only one of the training data models (bi,s,ga) overlaps in the top-3 under ISP and common corruptions. Similarly, the training data models under which task models perform best in individual testing conditions vary widely between ISP and common corruptions, both for classification and segmentation. Why does physical faithfulness matter in dataset drift testing? A test result is only as reliable as its constituting parts. If we are to rely on robustness test results to decide whether to use a task model in a certain data environment or not, we need to ensure the test cases represent real-world data models. If the test cases are not physically faithful, the results based on them are of limited use to make decisions.

Data models and targeted generalization

Recent advances in learning theory by Krikamol Muandet conjecture the impossibility to design rational learning algorithms that have the ability to learn across heterogeneous environments successfully [146]. Explicit data models allow us to rethink the problem of generalization in a similar vein. With data models it is possible to i) precisely specify individual environments and ii) observe what combinations of environments and task model play together nicely. Rather than selecting task models with best average performance across all heterogeneous test environments, we can serve the task model with the right data model depending on which environment it is deployed in. When we analyze the columns of the matrices in Figures 3.5 and 3.6 we can observe under what training data model (or 'environments' in Muandet's language) the task model performs best in which testing environment. These configurations are marked by a star ($\star$). For example, in the case of classification, we can observe that for a task model to perform well in (bi,u,me) and (bi,u,ga) test data environments, the (ma,u,ga) training data environment is best (left matrix, Figure 3.5). However, for the segmentation task, to perform best in the same testing data environments, the (bi,u,me) training data environment is preferable (left matrix, Figure 3.6).

Use cases and limitations of drift synthesis

The most immediate use case for drift synthesis is physically faithful, prospective validation without measurement. In this scenario, an engineer will have a task model as well as reference raw measurements. She would then construct data models of interest, for example for two different

microscopes across laboratory sites $\boldsymbol{s}$ and $\boldsymbol{t}$. She would then pipe the reference measurements through data models $\boldsymbol{s}$ and $\boldsymbol{t}$ to obtain two different datasets and test the task models on them. She could observe the effect of the processing in lab site $\boldsymbol{t}$ from a computer without ever having to take expensive measurements on-site. Building up a catalogue of data models, as we demonstrated in the experiments, further allows us to perform model selection or targeted generalization management where the task model is paired with suitable data models during deployment. All these applications presuppose access to raw data as well as knowledge of the data model specification so that they can be constructed accordingly in software.

3.3.2 Drift forensic

Clear and precise specification of the limitations of use is a mandated requirement for many products that can potentially contain machine learning components, such as software as a medical device [70, 234] or autonomous vehicles [156]. Without knowledge and control over the data acquisition process in practice, this requirement cannot be met. An explicit, differentiable data model paired with raw data offers a viable solution to this problem. $\boldsymbol{\Phi}^{\text{para}}_{\text{Proc}}$ enables the analysis of the task model's susceptibility to dataset drift in an interpretable manner using adversarial search. Related work, such as [165] also uses a differentiable raw processing pipeline to propagate the gradient information back to the raw image. There, however, the signal is used in a classical adversarial setup, to optimize adversarial noise on a per-image basis. In our work here, gradient updates are not applied to individual images, but to the data model parameters. The goal of such an analysis is to identify the parameter configurations of the data model under which the task model should not be operated. The resulting adjustments correspond to plausible changes which reflect changes in data model, for example, due to changing camera ISPs. In order to limit the parameter ranges, we chose an explicit constraint in the RGB space.

$$\underset{\widetilde{\theta}\in\Theta}{\text{minimize}} \quad \lambda\|\boldsymbol{V}-\widetilde{\boldsymbol{V}}\|_2^2 - \mathcal{L}(\widetilde{\boldsymbol{V}},\boldsymbol{Y})\,, \tag{3.33}$$

where $\boldsymbol{V} = \boldsymbol{\Phi}^{\text{para}}_{\text{Proc}}\left(\boldsymbol{X}_{\text{RAW}},\theta\right)$ are the RGB images obtained from the original data model and $\widetilde{\boldsymbol{V}} = \boldsymbol{\Phi}^{\text{para}}_{\text{Proc}}\left(\boldsymbol{X}_{\text{RAW}},\widetilde{\theta}\right)$ are the RGB images obtained from adversarial search on the data model parameters. Equation (3.33) maximizes the classification loss under a relaxed ℓ_2 constraint controlled by the hyperparameter $\lambda \geq 0$. This procedure yields data model parameters that deteriorate the task model performance while keeping the measured distortion minimal and the within constraints of physical faithfulness. All of the pipeline's parameters are optimized jointly to search for a task model's overall data model related weaknesses. Targeting select parameters is also possible and provides insight into a parameter's effect on the task model's performance.

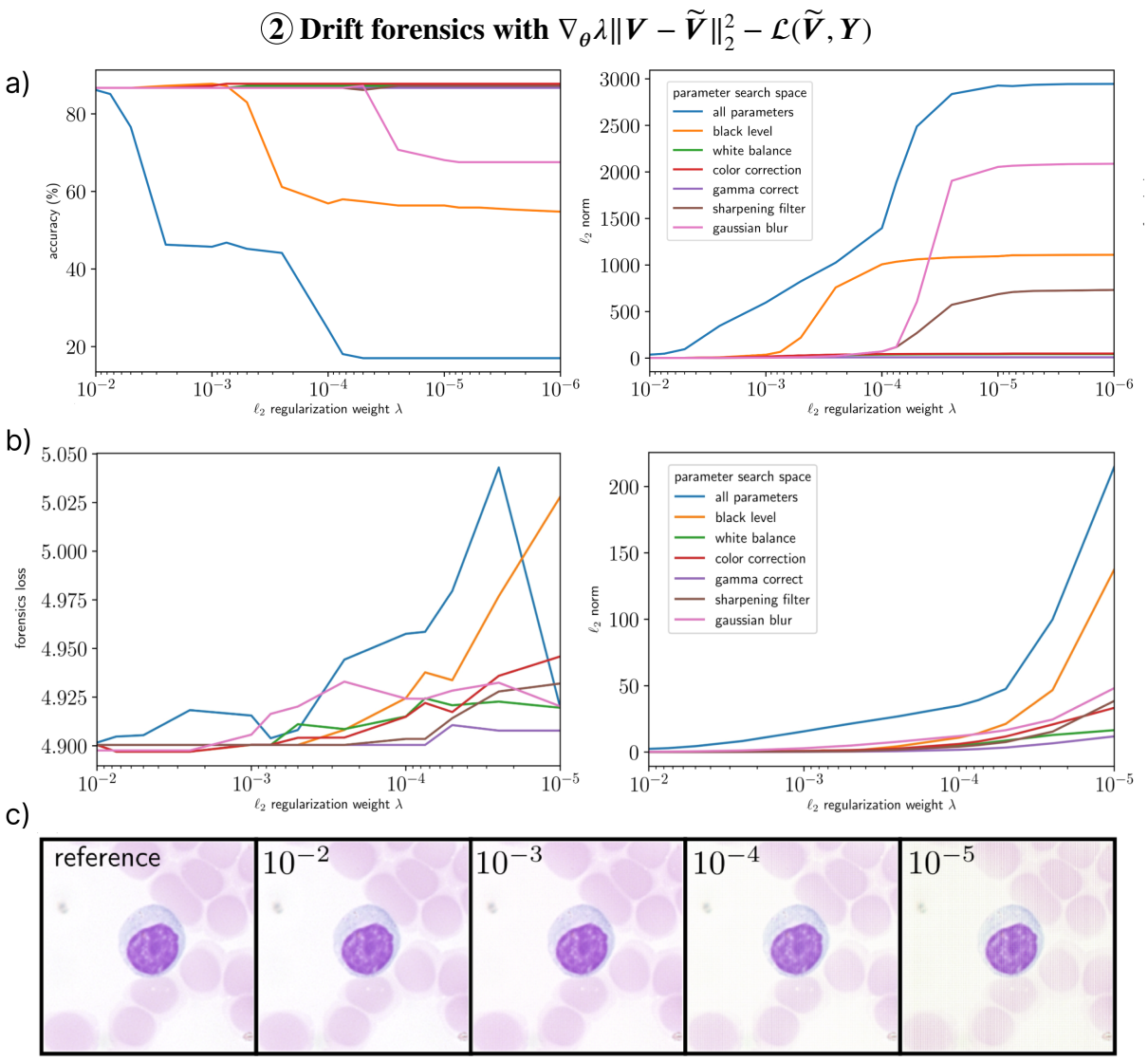

Figure 3.8: (a): Test accuracy on the Raw-Microscopy test set after 20 epochs of adversarial search in the data model for varying regularization weight parameters λ. The individual plots depict the various pipeline parameter selections (left plot). Plot showing ℓ_2-norm (of processed images between the adversarially trained $\widetilde{\boldsymbol{\Phi}}^{\text{para}}_{\text{Proc}}$ and the default $\boldsymbol{\Phi}^{\text{para}}_{\text{Proc}}$) versus attained accuracy of the task model (right plot). The metrics are evaluated on the test set after 20 epochs of adversarial optimization for varying regularization weight parameter λ. (b): Same for Raw-Drone. The individual plots depict the various data model parameter selections. A lower regularization results in a bigger search space for adversarial optimization. Forensics loss refers to the binary cross entropy and Dice loss used as the optimization objective for the segmentation task model. (c): Processed samples from the drift forensics after 20 epochs with varying regularization weights λ.

Sensitivity to data models

The left plot in block (a) of Figure 3.8 shows sensitivities of the classification task model to changes in the data model parameters. With increased relaxation of the ℓ_2 regularization, the accuracy declines exposing configurations under which the task model deteriorates. As to be expected,

the setting allowing for all parameters to be altered shows the biggest effect on the resulting performance. Individually, changes in the black level configuration Φ_{BL}^{para} and the denoising parameters Φ_{DN}^{para} pose the greatest risk for task model performance under a relaxed regularization weight. In contrast to the classification model, the performance drops for the segmentation model are less severe (left plot in block (b) of Figure 3.8). We hypothesize that this is because classification problems are inherently discontinuous while inverse problems inherently allow for more stable solutions [59], thus being less susceptible to instabilities.

Sensitivity in relation to magnitude

For comparison, the right plot in block (a) of Figure 3.8 shows the regularization weight λ against the resulting ℓ_2. Interestingly, a higher norm in the resulting RGB images does not directly translate to the most severe performance degradation of the task model. At $\ell_2 = 10^{-5}$, changes in the Gaussian blur parameters induce a norm almost twice as large as the changes in the black level parameters. However, the corresponding drop in accuracy caused by Gaussian blur is around one third less relative to the black level. Similarly, at $\ell_2 = 10^{-5}$, the sharpening filter parameters incur a norm but do not lead to accuracy drops of the task model. This underscores the importance of precise data models for dataset drift validation. Physically faithful yet small changes, as visible in the samples in the bottom row of Figure 3.8, in processed images can have a larger impact on the performance than large changes.

Use cases and limitations of drift forensics

A practical use-case of drift forensics looks as follows: party ***s*** develops and trains a model and then licenses it to party ***t*** for use. Party ***t*** wants to know what the data conditions are under which the task model performs well and under which conditions it should not be used. Party ***s*** runs drift forensics and provides party ***t*** with a forensic signature, as seen in Figure 3.8, detailing which parameters in the data model can be changed and which should not be touched to maintain task model performance. Party ***t*** can use this information to calibrate their data processing and knows which data settings to avoid for the specific task model. As with the other drift controls, access to the raw data, as well as data models, is required to perform drift forensics.

3.3.3 Drift optimization

In the previous two experiments, we demonstrated how raw data and a differentiable data model can be used to identify and then modularly test for unfavourable data models that should be

avoided during the deployment of the machine learning task model. The same mechanics can also be exploited to optimize the data itself, effectively creating a beneficial drift. In the drift optimization setting, the gradient from the task model $\mathbf{\Phi}_{\text{Task}}$ is propagated into the data model $\mathbf{\Phi}_{\text{Proc}}$ to jointly optimize both of them.

In the *learned* setting, the data model parameters are jointly optimized with the task model parameters. In the *frozen* setting, only the task model parameters are optimized and the data model parameters are kept fixed. The initialization of $\mathbf{\Phi}_{\text{Proc}}^{\text{para}}$ (both *frozen* and *learned*) is set to standard values which can be found in Appendix A.1 as well as in `pipeline_torch.py` of the code.

Convergence and stability

In the left column (a) of Figure 3.9 these two scenarios are compared. The *learned* data model creates a drift that improves the stability of the learning trajectory. This is indicated by the blue line which displays the validation accuracy against optimization steps for the first half of training (step 1439 corresponds to epoch 60). It exceeds that of the *frozen* data model (red line) by up to 25 percentage points in accuracy at a lower variance. For the segmentation task (bottom row Figure 3.9) the stabilization effect cannot be observed. This could be due to the low resolution of the problem itself as the drift optimization may not have a large effect on enhancing the solid blocks of cars in the raw data. Other evidence further suggests that inverse problems are inherently less unstable [59]. The results of the convergence and stability behaviour under the different settings can also be found as a tabular summary in Table 3.5.

Helpful artifacts

In fact, the processed image from a *learned* data model with optimized drift (see *learned* column in block (a) of Figure 3.9 for an example) can contain visible artifacts that *aid* stability and generalization vis-à-vis the image from the *frozen* baseline data model which, arguably, looks cleaner to the human eye.

A possible explanation for the improved learning trajectory could be that a varying optimized drift automatically generates samples akin to data augmentation. Such uses could further be explored in scarce data settings like fine tuning, semi-supervised or few-shot learning. Having gradient access to the data model thus offers the opportunity to optimize data generation itself for a given machine learning task. If learned data models are to be applied in real-world applications, it thus appears likely that a tradeoff has to be made between human perceived visual quality and artifacts that can be helpful to the task model.

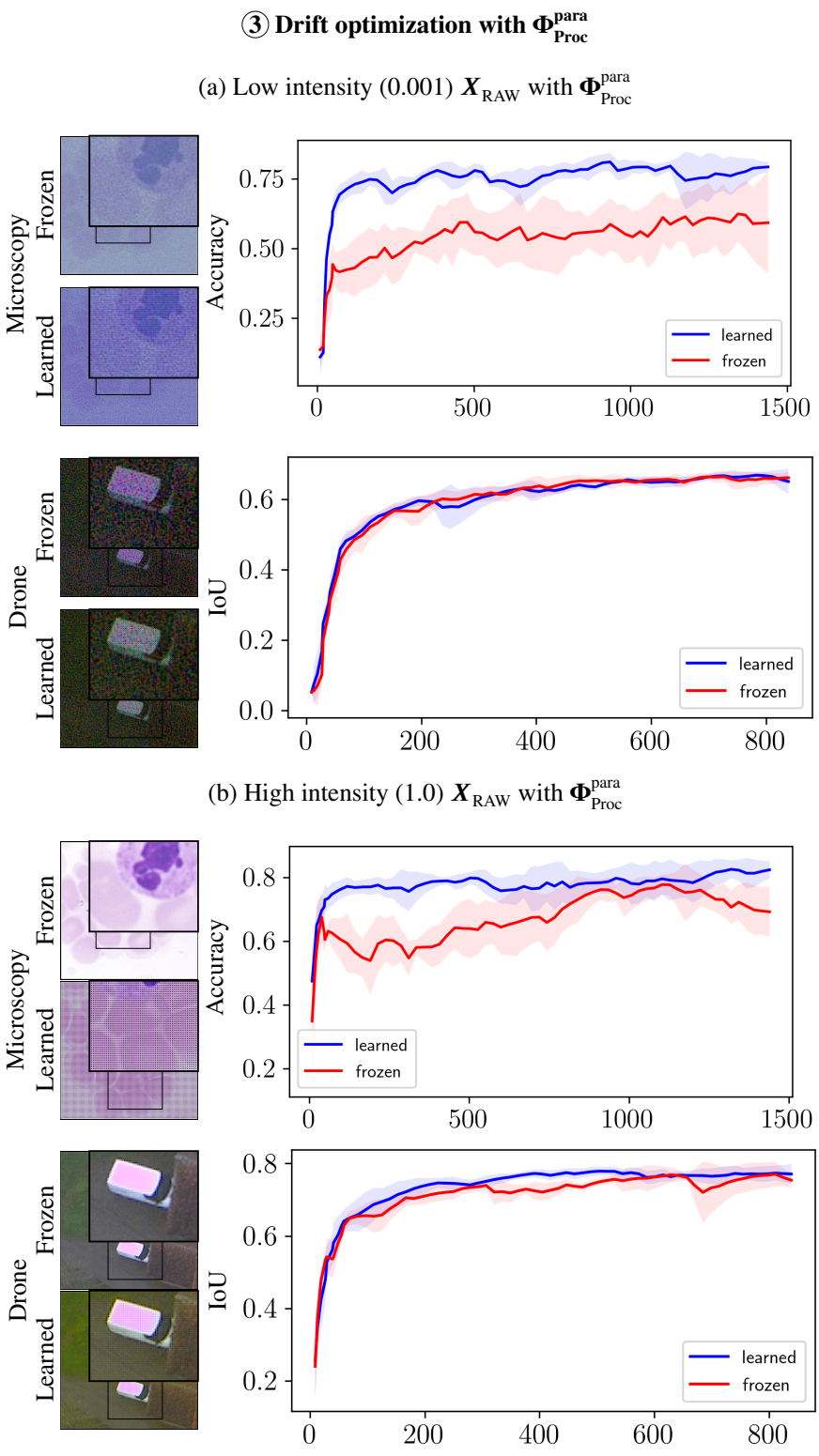

Figure 3.9: Low (a) and high (b) intensity images processed by a *frozen* and a *learned* pipeline. This type of drift optimization would not be possible with processed data. The plots columns three and six display the mean of validation metrics over five cross validation runs. Column seven shows additional results on raw data for comparison. Error bars are reported as one standard deviation. Optimization step 1439 and 915 correspond to epoch 60 into training.

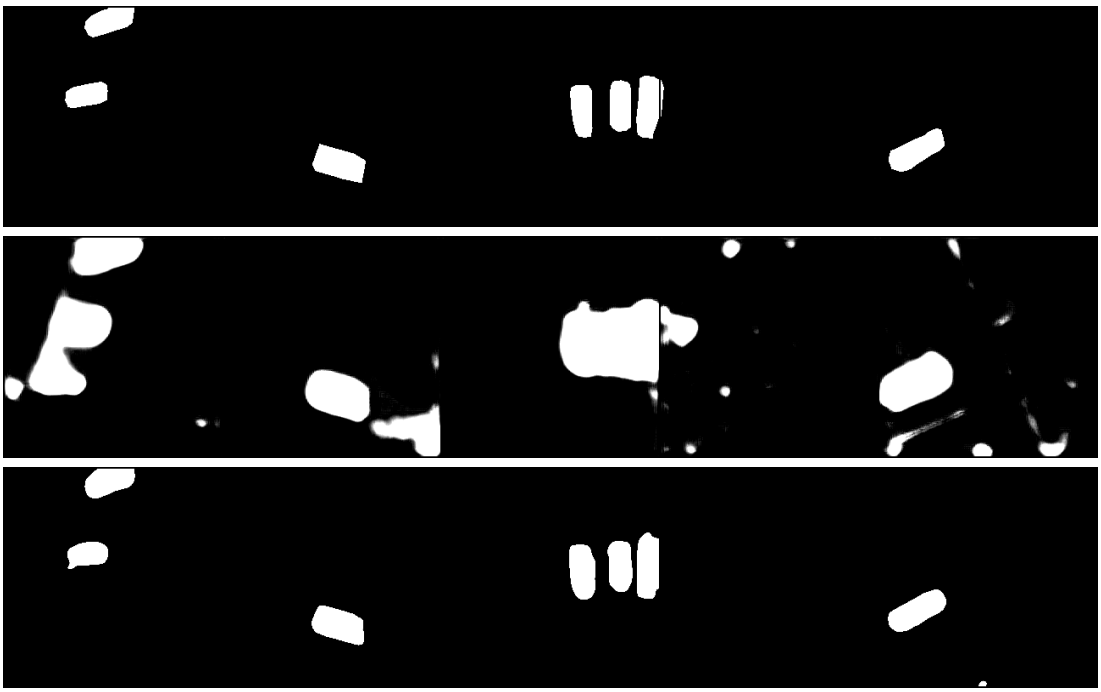

Figure 3.10: A set of random test samples for the segmentation task under learned processing. Top row: Targets, middle row: predictions of the task model after the first epoch, last row: predictions of the task model after the last epoch.

Similar outcomes for stability and artifacts can also be observed for the reverse situation (high intensity 1.0 $\boldsymbol{x}_{\mathrm{RAW}}$) in the right column (b) of Fig. 3.9. An example of the segmentation mask optimization during the learning process is shown in Fig. 3.10.

Raw and data models

We demonstrated how parametrized data models can be used to optimise drift under data model constraints. Going beyond physically faithful drift controls, an interesting extension to drift optimization is training directly on raw data to optimize task model performance. While in this chapter we are concerned with providing building blocks to emulate optical data models used in practice, training directly on raw opens up the possibility to learn purely machine-optimized optical data processing free of existing data model constraints. In the last column of Figure 3.9 an optimization directly on raw data is displayed for each task. The raw data is demosaiced using `class RawToRGB(nn.Module)` from `/processing/pipeline_torch.py` in the data model code. Then task models are tuned to raw data under the same regimes described above. Results are reported across threefold cross-validation with error bars of one standard deviation. Like in the other experiments the task model parameters are tuned as well. For classification, a performance similar to the *learned* setting is achieved with a more volatile optimization trajectory. For the segmentation task, the performance is not on par with either the *learned* or *frozen* setting, but

	Microscopy Average accuracy	**Drone** Average IoU
Learned (low)	0.75 ± 0.09	0.59 ± 0.05
Frozen (low)	0.54 ± 0.21	0.59 ± 0.05
Learned (high)	0.78 ± 0.08	0.74 ± 0.04
Frozen (high)	0.67 ± 0.14	0.71 ± 0.05
Direct raw	0.75 ± 0.07	0.60 ± 0.07

Table 3.5: Tabular summary of the drift optimization results. The average accuracy and standard deviations over cross-validation runs and training steps are displayed, summarizing both the stability and converge trajectory for each setting.

it appears plausible that this gap can be substantially reduced with further finetuning. Learning directly on raw data thus appears as a promising direction for data model-free machine vision.

Use cases and limitations of drift optimization

Drift optimization can be used to squeeze out performance from a task model by creating drift that is helpful. A common use case is the adjustment of imaging pipelines, such as microscopes, that have traditionally been optimized for human end users, for example, medical staff, which are increasingly being used in conjunction with automated machine learning models, for example for cell detection. By adjusting the parameters in the data models of existing optical laboratory infrastructure performance gains can be achieved. Due to the improved convergence and stability of the optimization trajectory, it can also potentially be used in situations where computing is expensive or scarce altogether. However, these benefits do not hold across all tasks, as we saw in the case of the segmentation model, and for cases where no data models are present, such as novel or custom optical hardware, learning directly on raw data offers a promising extension. This work targets imaging infrastructure that uses ISPs causing data drift while also allowing access to raw sensor readouts. The proposed data models enable engineers to emulate and control different data generating processes related to ISPs in a cost-effective, physically faithful manner. These models save time and money and enable new applications for data-model quality management. However, they only capture ISP-related drifts and require extensions to model other factors. The ultimate goal could be to train on RAW data, with the current pipeline serving as an interim solution until RAW data becomes more widely available.

3.4 Discussion

The main message we hope to convey in this chapter is that black-box data models for images neither have to be the norm in machine learning research nor in engineering. Leveraging established knowledge from physical optics enables us to push the modelling goalpost further towards machine learning's core ingredient: the data. Paired with raw data, precise differentiable data models for images allow for advanced controls of dataset drift, a common and far reaching challenge across many machine learning disciplines. Applications beyond robustness validation in areas of machine learning that are also held back by black-box data, such as federated learning and formal model certifications, appear opportune, too.

Drift synthesis allows the creation of physically faithful of drift test cases. In contrast to augmentation testing, the performance drops for physically faithful test cases are less severe across the board for both uses cases in our experiments, changing the conclusions we arrive at for model selection and enabling new ways to think about generalization with targeted, data model specific deployment of task models. A plausible practical application scenario of drift synthesis for machine learning researchers and practitioners is the prospective validation of their task model to drift from different camera devices, for example microscopes across different lab sites or autonomous vehicles, without having to collect measurements from the different devices. Drift synthesis could also be interesting for other application domains that rely on data synthesis (semi- [161, 253, 17] and self-supervised learning [232, 69]) or on precise data models (aleatoric uncertainty quantification [155, 207, 53, 114, 56, 135, 241, 129, 159, 147, 6], out-of-distribution detection [39, 218, 118, 149, 187, 45]). While we cross-validated a substantial number of data model variations in our experiments, it should be noted that further variations, for example by reordering or adding steps, are possible. Furthermore, it should not be overlooked that dataset drift can also be caused by factors outside the ISP data model, for example the optical components of a camera. These *data models* are not yet capable of capturing factors that go beyond the ISP. Integrating work from lens manufacturing [238] to expand the reach explicit data models offers a promising next step for drift synthesis, explored in more detail in the next chapter. Drift forensics allows the precise specification of data model limitations of use for a given machine learning task model. Data models under which the task model should not be operated can be identified by gradient search and then documented. In our demonstration, the setting allowing for all parameters to be altered shows the biggest effect on the resulting performance. Individually, changes in the black level configuration and the denoising parameters pose the greatest risk for performance of the task model at hand. Interestingly, a higher norm in the resulting RGB images does not directly translate to the most severe performance degradation of the task model. This underscores the importance of precise data models for dataset drift validation. In practice, clear specification of the limitations of use is a mandated requirement for many products that can potentially contain

machine learning components, such as software as a medical device [70, 234] or autonomous vehicles [156]. Drift forensics with explicit data models can help to utilize the precision of machine learning and data engineering to satisfy such regulatory constraints. Explicit data models combined with gradient search may also be interesting to explore in areas such as formal model verification [254, 121, 262, 91, 196, 67, 195, 40, 33, 257, 18, 64, 63, 221, 28, 36, 57] to obtain tighter error bounds. Other constraints beyond ℓ_2 are feasible, depending on the particular use case to be analyzed, and can be plugged into our code[7]

We also showed how differentiable data models can be used for drift optimization where the data generating process is jointly optimized with the task model parameters. It leads to improved stability of the learning trajectory on the classification task in both low and high intensity measurements.

Interestingly, the processed image from a *learned* data model can contain visible artifacts that *aid* stability and generalization vis-à-vis the image from the *frozen* baseline data model which arguably looks cleaner to the human eye. In practice, the extension of the gradient connection from the task model $\mathbf{\Phi}_{\text{Task}}$ to the data model $\mathbf{\Phi}_{\text{Proc}}$ enables the extension of machine learning right into the data generating process. Thus, data generation itself can be optimized to best suit the task model at hand. Furthermore, the stabilization effect could prove useful for learning problems where training is costly and speedup precious (for example large models or large datasets). This capacity could also be exploited in other areas that deal with heterogenous training or deployment environments, such as different clients in federated learning [199, 200, 246] or domain adaptation techniques [15]. However, the above drift adjustment benefits could only be observed for the classification task, not the regression task, possibly due to the low resolution of the segmentation problem. How far we can push the gradient into the real world is an interesting future direction for data modelling. Including more parts of the data acquisition hardware into the data model and consequently the machine learning optimization pipeline appears feasible [252] and represents an important next step in aligning machine learning with real world data infrastructures.

Finally, raw data, which is already routinely used in optical industries [72, 164, 261, 258, 95, 186], for representative machine learning tasks has to become more accessible to researchers to align robustness research with physically faithful data models and infrastructures. While most optical imaging devices support the extraction of raw data and this procedure is well established in industry and physics, data collection procedures for machine learning robustness research still have to catch up in order to make raw datasets and their benefits more widely available. Norms around established benchmarking datasets of processed images, such as CIFAR or ImageNet, can slow down this progress. To that end, we collected and publicly release two raw image datasets in the camera sensor state. The assumptions with respect to the practicality of the procedures

[7] Argument `args.adv_aux_loss` in `train.py`

we propose here are mild in our eyes. Raw subsets of data could be stored and then pulled in-code from cloud storage, as demonstrated in the code that we provide, for the purposes of drift synthesis or drift forensics. Learned data models obtained from drift adjustments could be calibrated directly on hardware such that the bandwidth requirements would not change compared to current image acquisition and transmission. Better APIs to optical hardware would allow more researchers and industries to make their raw data accessible and service a culture of data modelling that can help overcome the limitations of machine learning in the pure task model regime. Machine learning risk management, such as drift controls, can make ML deployment possible and safer. More deployment translates to increases in automation.

Chapter 4.

Data-centric workflow based on raw images

Due to confidentiality issues this chapter is not available for viewing.

Chapter 5.

DiffInfinite: Large Mask-Image Synthesis via Parallel Random Patch Diffusion

In this chapter, we investigate how to generate synthetic medical images with biological plausibility, given information on the long-range spatial correlation. We introduce a novel hierarchical diffusion model framework to generate arbitrarily large images along with its segmentation mask. The proposed method generates high-fidelity histological images while preserving long-range correlation structural information. The model can be parallelized more efficiently in inference than previous large-content generation methods while avoiding tiling artefacts. This framework alleviates unique challenges in histopathological imaging practice like large-scale information, costly manual annotation, and protective data handling. The clinical relevance of DiffInfinite synthetic data is evaluated in a survey by ten experienced pathologists as well as a downstream classification and segmentation task. Samples from the model score strongly on anti-copying metrics, indicating that it does not reproduce patterns identical to those in the input training data, which is relevant for the protection of patient data.

5.1 Synthetic data in medical imaging

Deep learning (DL) models are promising auxiliary tools for medical diagnosis [3, 163, 219]. Applications like segmentation and classification have been refined and pushed to the limit on natural images [248]. However, neural networks trained on rich datasets still have limited applications in medical data. While segmentation models rely on sharp object contours when applied to natural data, in medical imaging, the model struggles to detect a specific feature because it has a "limited ability to handle objects with missed boundaries" and often "miss tiny and low-contrast objects" [71, 134]. Therefore, task-specific medical applications require their own specialised and fine-grained annotation. Data labelling is arguably one of the most critical

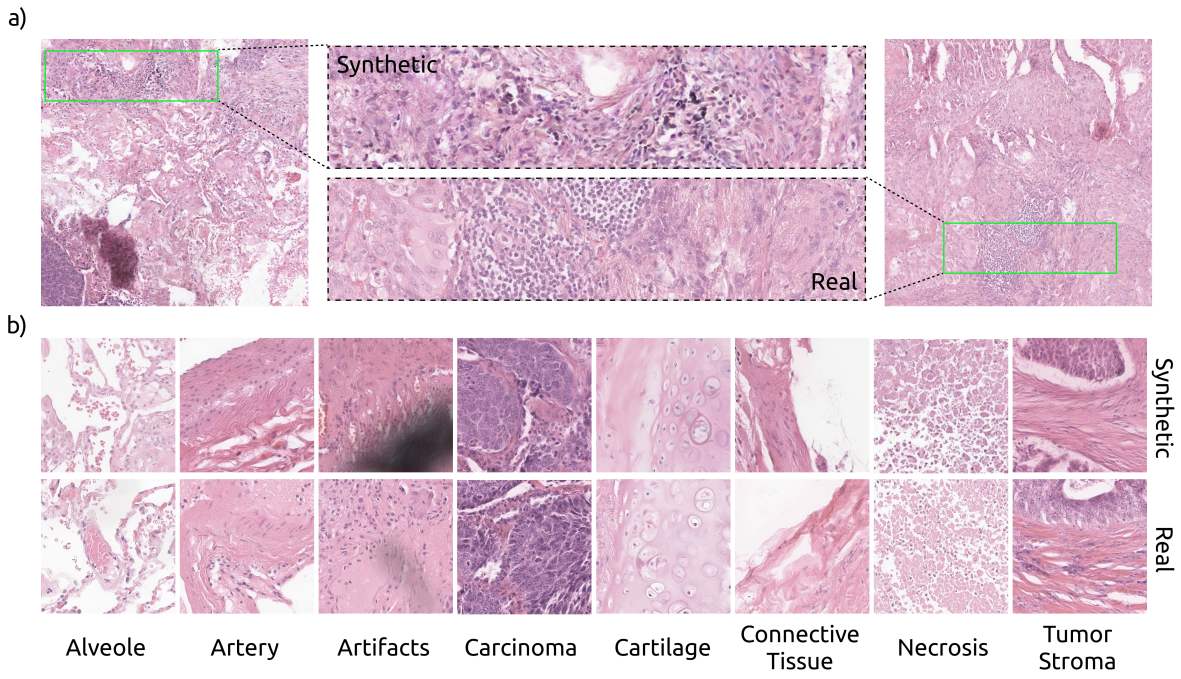

Figure 5.1: a) Examples of synthetic and real 2048×2048 images. b) Pairs of 512×512 synthetic tiles (top) with the closest real images found with Inception-v3 near-neighbour (bottom).

bottlenecks in healthcare machine learning (ML) applications. In histopathology, pathologists examine the histological slide at multiple levels, starting with a lower magnification to analyse the tissue architecture and cellular arrangement and gradually proceeding to a higher magnification to examine cell morphology, including aspects such as alterations in the nucleus-to-cytoplasm ratio, anisonucleosis, and the presence of mitotic figures. Annotating features within gigapixel whole slide images (WSIs) with this level of detail demands effort and time, often leading to sparse, limited annotated data. In addition, due to privacy regulations and ethics [180, 157], having access to medical data can be challenging since it has been shown that it is possible to extract patients' sensitive information [205] from it.

In histopathology, state-of-the-art ML models require the context of the entire WSIs, with features at different scales, in order to distinguish between different tumor sub-types, grades and stages [30]. Despite the demonstrated effectiveness of diffusion models (DMs) in generating natural images compared to other approaches, they still have rarely been applied in medical imaging. Existing generative models in histopathology can generate images of relatively small resolution compared to WSIs. To give a few examples, the application of Generative Adversarial Networks (GANs) in cervical dysplasia detection [256], glioma classification [83], and generating images of breast and colorectal cancer [172], generate images with 256×128 px, 384×384 px and 224×224 px, respectively. In spite of their current limitations in generating images at scales necessary to fully address all medical concerns, the use of synthetic data in medical imaging can provide a valuable solution to the persistent issue of data scarcity [171, 26, 25, 160]. Models

generally improve after data augmentation and synthetic images are equally informative as real images when added to the training set [120, 48]. Data augmentation could also help with the underrepresentation in data sets of rare cancer subtypes. By adding synthetic images to the training set, Chen et al. [29] demonstrated that their model had better accuracy in detecting chromophobe renal cell carcinoma, which is a rare subtype of renal cell carcinoma. Furthermore, Doleful et al. [44] showed how synthetic histological images could be used for educational purposes for pathology residents. Regarding the challenges highlighted before, we present a novel sampling method to generate large histological images with long-range pixel correlation (see Fig. 5.1), aiming to extend up to the resolution of the WSI.

In this chapter, the primary contribution is the introduction of DiffInfinite, a novel hierarchical generative framework that exhibits the capability to generate images of arbitrary sizes, alongside their corresponding segmentation masks. This framework not only expands the horizons of generative modeling but also has the potential to revolutionize medical imaging applications by providing high-quality, large-scale data that is invaluable for research and clinical practice. Furthermore, generative models can be implicitly used to learn features within the data. By employing the diffusion model to generate synthetic data mirroring the distribution of the training data, we can extract significant insights from the actual datasets. This process involves adding noise to an image and subsequently denoise it to reconstruct the exact same image, during which we retrieve self-attention masks from the model that highlight the most import information in the data, for example using "carcinoma" as prompt the model would highlight in the self-attention mask the carcinoma in the generated image. Although synthetic data are not yet broadly utilized for downstream tasks, their potential to enhance model performance on previously unseen data is considerable. The rapid advancement in high-fidelity generation opens the way for creating new synthetic datasets from a minimal number of examples, which could further enhance model robustness.

Another key contribution of our work is the development of a fast outpainting method designed for efficient parallelization. This method addresses the challenge of generating image content beyond the boundaries of existing data, ensuring that our generative model can produce coherent and contextually relevant extensions of medical images. The speed and parallelization capabilities of this approach are crucial for scaling up the generation process, making it practical for real-world applications.

Furthermore, we rigorously evaluate the quality of the data generated by DiffInfinite through a comprehensive assessment. We engage ten experienced pathologists to distinguish between the generated and real images and to provide feedback on the generated samples. Additionally, we examine the utility of DiffInfinite data in downstream machine learning tasks, including classification and segmentation. To further address concerns regarding patient data privacy, we

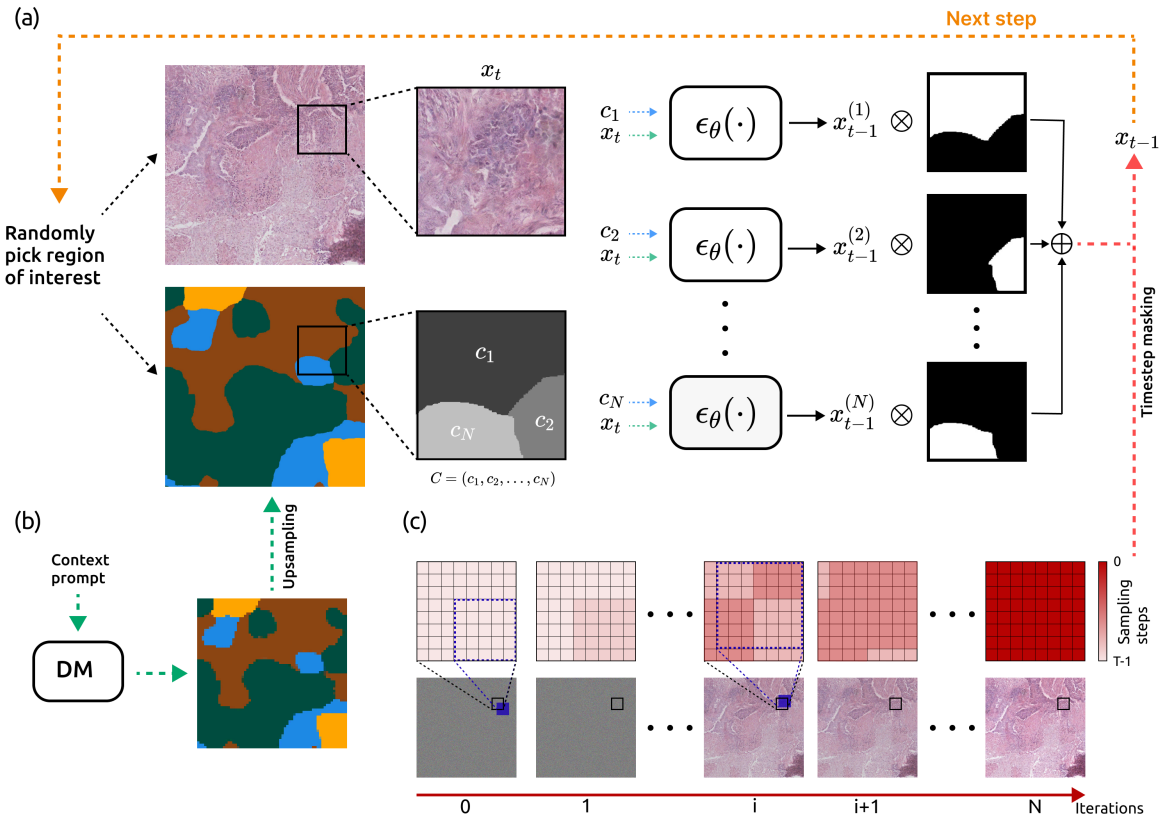

Figure 5.2: DiffInfinite generation method. a) Diffusion steps on large images. Given a random position, we select a sub-tile with its segmentation mask. A diffusion model generates in parallel the next step conditioned on each conditional label, or prompt, found in the mask. The outputs are masked individually with the corresponding label. The next step is the union of all the sub-patches. b) Large-scale context mask generation. A diffusion model conditioned on a large-scale conditional prompt (e.g. Adenocarcinoma subtype) generates a low-resolution mask. The mask is upsampled via linear interpolation to the desired image size. c) Tracking timesteps pixel-wise. We keep track of the time step of each pixel in the large image. The model evolves only the pixels with the higher timestep on each iteration.

employ anti-duplication metrics to assess the potential leakage of sensitive patient information from the training dataset. This comprehensive evaluation ensures that our generative framework not only produces high-quality data but also adheres to privacy and ethical standards.

5.2 Infinite diffusion

The DiffInfinite approach we present here,[1] is a generative algorithm to generate arbitrarily large images without imposing conditional independence, allowing for long-range correlation structural information. The method overcomes this limitation of DMs for large-content generation by

[1]Code available at `https://github.com/marcoaversa/diffinfinite`

deploying multiple realizations of a DM on smaller patches. In this section, we first define a mathematical description of this hierarchical generation model and then describe the sampling method paired with a masked conditioned generation process.

5.2.1 The method

Let $X \sim \mathcal{X}$ be a large-content generating random variable taking values in $\mathbb{R}^{KD}$. Using the approach of latent diffusion models [181], the high-dimensional content is first mapped to the latent space $\mathbb{R}^D$ by $\Phi(X) = Y \sim \mathcal{Y}_\Phi$. For simplicity, we assume throughout this work the existence of an ideal encoder-decoder pair (Φ, Ψ) such that $\Psi(\Phi(X)) = X$ is the identity on $\mathbb{R}^{KD}$. Assume further, to have a reverse time model $(SM_\theta, \epsilon_\theta)$ at hand consisting of a sampling method SM_θ and a learned model ϵ_θ trained on small patches $Z \sim \mathcal{Z}_\Phi$ taking values in $\mathbb{R}^d$. The reverse time model transforms $z_T \sim \mathcal{N}(0, I_d)$ over the time steps $t \in \{T, T-1, ..., 1\}$ recursively by

$$z_{t-1} = SM_\theta(z_t) \tag{5.1}$$

to an approximate instance of $\mathcal{Z}_\Phi$. We aim to sample instances from $\mathcal{Y}_\Phi$ by deploying multiple realizations of the reverse time model $(SM_\theta, \epsilon_\theta)$. Towards that goal, define the set of projections

$$\mathcal{C} := \{proj_I : \mathbb{R}^D \to \mathbb{R}^d \mid I \subset \mathbb{N} \text{ correspond to } d \text{ indices of connected pixels in } \mathbb{R}^D\}, \tag{5.2}$$

where $proj \in \mathcal{C}$ models a crop $proj(Y) \in \mathbb{R}^d$ of d connected pixels from the latent image Y. Since the model ϵ_θ is trained on images taking values in $\mathbb{R}^d$ the standing assumption is

Assumption 1. *Any projection $proj \in \mathcal{C}$ maps Y to the same distribution $proj(Y) \sim \mathcal{Z}_\Phi$ in $\mathbb{R}^d$.*

Since the goal is to approximate an instance of $\mathcal{Y}_\Phi$, we initialize the sampling method by $y_T \sim \mathcal{N}(0, I_D)$ and proceed in the following way: Given y_t, randomly choose $proj_{I_1}, ..., proj_{I_m} \in \mathcal{C}$ independent of the state y_t such that $proj_{I_1}, ..., proj_{I_m}$ are non equal crops that cover all latent pixels in $\mathbb{R}^D$. To be more precise, for every $i \in \{1, ..., D\}$ we find at least one $j \in \{1, ..., m\}$ with $i \in I_j$. For every projection $proj_{I_1}, ..., proj_{I_m}$ we calculate the crop $z_t^j = proj_{I_j}(y_t)$ of the current state y_t and perform one step of the reverse time model following the sampling scheme

$$z_{t-1}^j = SM_\theta(z_t^j), \quad j \in \{1, ..., m\}. \tag{5.3}$$

This results in overlapping estimates $z_{t-1}^1, ..., z_{t-1}^m$ of the subsequent state $t-1$ and we simply assign to every pixel in the latent space the first value computed for this pixel such that

$$[y_{t-1}]_i = [z_{t-1}^j]_l, \quad \text{where } j = \min\left\{j' \mid i \in I_{j'}\right\} \tag{5.4}$$

and l refers to the entry in z^j_{t-1} corresponding to i with $[proj_{I_j}(y_{t-1})]_l = [y_{t-1}]_i$. Hence, starting from $y_T \sim \mathcal{N}(0, I_D)$ we sample in the first step from a distribution

$$y_{T-1} \sim p_{T-1,\theta}(y \mid y_T, proj_{I_1}, ..., proj_{I_m}). \tag{5.5}$$

Using Bayes' theorem, this distribution simplifies to

$$p_{T-1,\theta}(y \mid y_T, proj_{I_1}, ..., proj_{I_m}) = p_{T-1,\theta}(y \mid y_T), \tag{5.6}$$

since we sample the projections independently from y_T. Repeating the argument, we sample in every step from a distribution $y_{t-1} \sim p_{t-1,\theta}(y|y_t, ..., y_T)$ over $\mathbb{R}^D$ instead of sampling from $z_{t-1} \sim q_{t-1,\theta}(z|z_t, ..., z_T)$ over $\mathbb{R}^d$. Hence, we approximate the true latent distribution $\mathcal{Y}_\Phi$ by the approximate distribution with density $p_{0,\theta}(y|y_1, ..., y_T)$. In contrast to [264], our method does not use the assumption of conditional independence and the method can be applied to a wide range of DMs, without an adjustment of the training method. As the authors of [264] point out in their section on limitations, the assumption of conditional independence is not well-suited in cases of a data distribution with long-range dependence. For image generation in the medical context, we aim to circumvent this assumption as we do not want to claim that the density of a given region depends only on one neighboring region. The drawback of dropping the assumption is that we only approximate the reverse time model of the latent image distribution $\mathcal{Y}_\Phi$ indirectly, by multiple realizations of a reverse time model that approximates $\mathcal{Z}_\Phi$.

5.2.2 Semi-supervised guidance

In order to generate diverse high-fidelity data, DMs require lots of training data. Perhaps, training on a few samples still extracts significant features but it lacks variability, resulting in simple replicas. Here, we show how to enhance synthetic data diversity using classifier-free guidance as a semi-supervised learning method. In the classifier-free guidance [79], a single model is trained conditionally and unconditionally on the same dataset. We adapt the training scheme using two separate datasets. The model is guided by a small and sparsely annotated dataset q_1, used for the conditional training step, while extracts features by the large unlabelled dataset q_0, used on the unconditional training step (see Alg.1)

$$(z_0, c) = \begin{cases} (z_0, \varnothing) \sim q_0(z_0) & \text{if } u \geq p_{unc} \\ (z_0, c) \sim q_1(z_0, c) & \text{otherwise} \end{cases}, \tag{5.7}$$

where u is sampled from a uniform distribution in [0,1], p_{unc} is the probability of switching from the conditional to the unconditional setting and $\varnothing$ is a null label. During the sampling, a tradeoff

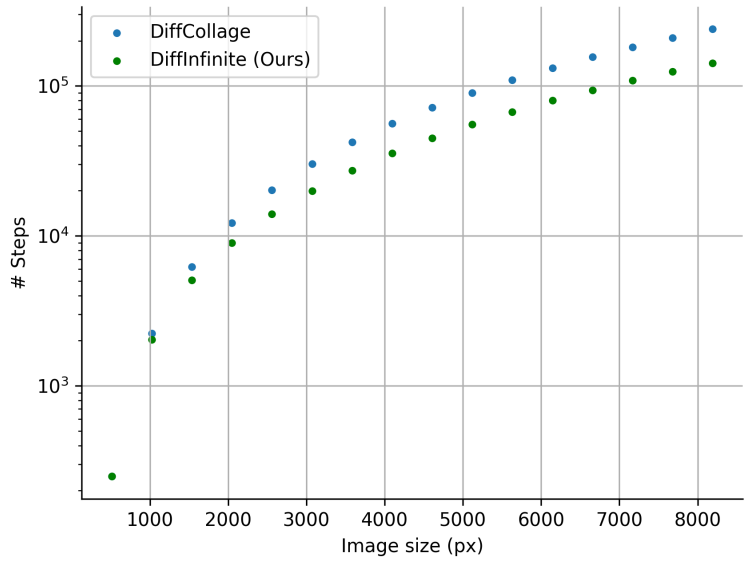

Figure 5.3: Comparison of sampling speed for DiffCollage and DiffInfinite, measuring diffusion steps required for image sampling. Demonstrating increased efficiency of DiffInfinite for larger images.

between conditioning and diversity is controlled via the parameter ω in eq.2.7.

5.2.3 Sampling

High-level content generation The outputs of DMs have pixel consistency within the training image size. Outpainting an area with a generative model might lead to unrealistic and odd artifacts due to poor long-range spatial correlations. Here, we show how to predict pixels beyond the image's boundaries by generating a hierarchical mapping of the data. The starting point is the generation of the highest-level representation of the data. In our case, it is the sketch of the cellular arrangement in the WSI (see Figure 5.2a). Since higher-frequency details are unnecessary at this stage, we can downsample the masks until the clustering pattern is still recognizable. The diffusion model, conditioned on the context prompt (e.g. Adenocarcinoma subtype), learns the segmentation masks which contain the cellular macro-structures information.

Random patch diffusion Once the long-range correlation content in the segmentation mask M is generated, we can proceed with the large image sampling according to Section 5.2.1 in the latent space $\mathbb{R}^D$ of $Y = \Phi(X)$ (see Alg.2). Since we trained a conditional diffusion model with conditions $c_1, ..., c_N$, the learned model takes the form

$$\epsilon_\theta(z_t, t) = (\epsilon_\theta(z_t, t|c_1), ..., \epsilon_\theta(z_t, t|c_N)). \tag{5.8}$$

Given y_t, we first sample projections $proj_{I_1}, ..., proj_{I_m} \in C$, corresponding to different crops of d connected pixels up to the m-th projection with $\cup_{j=1}^{m} I_j = \{1, ..., D\}$ and $\cup_{j=1}^{m} I_j \setminus \cup_{j=1}^{m-1} I_j \neq \emptyset$ (see the left hand-side of Figure 5.2b). Note that m is not fixed, but varies over the sampling steps and is upper bounded by the number of possible crops of d connected pixels. The random selection of the projection is implemented such that regions with latent pixels of low projection coverage are more likely. Secondly, we calculate for every projection $j \in \{1, ..., m\}$ the crop $proj_{I_j}(y_t)$ and perform one step of the DDIM sampling procedure using the classifier-free guidance model

$$(1 + \omega)\epsilon_\theta(z_t^j, t, c) - \omega\epsilon_\theta(z_t^j, t, \varnothing), \tag{5.9}$$

where ϵ_θ is the learned model and $z_t^j = proj_{I_j}(y_t)$. This results for every pixels $i \in I_j$ in N values $DDIM_{\theta,c_1}(proj_{I_j}(y_t)), ..., DDIM_{\theta,c_N}(proj_{I_j}(y_t))$, one for every condition c_i (see the right hand-side of Figure 5.2b). If $i \notin I_{j'}$ for all j', the pixel i has not been considered yet and we assign i the value $[y_{t-1}]_i = [DDIM_{\theta,M_i}(proj_{I_j}(y_t))]_l$, where l corresponds to the pixel i under the projection I_j and M_i is the value of i in the mask M.

Time tracking Since we are updating random projections of the overall image, in the t-th step pixels either have the time index t or $t + 1$, resulting in a reversed diffusion process of differing time states. We initialize a tensor L_t, with the same size D as the latent variable, to keep track of the time index for each pixel. Each element is set to $L_T \equiv T$. In the j-th iteration of the t-th step we only update the pixels that have not been considered in one of the previous iterations of the t-th diffusion step, hence all the pixels in $i \in I_j$ with $proj_{I_j}(L_t)_i = t + 1$, similarly to the inpainting mask in the Repaint sampling method [132]. To restore the pixels that already received an update, i.e. every pixel $i \in I_j$ with $proj_{I_j}(L_t)_i = t$, we store a replica of the previous diffusion step for every pixel. Finally, we update all the time states in L_t that received an update in the j-th iteration to t resulting in $proj_{I_j}(L_t)_i = t$ for all $i \in I_j$. See the top row of Fig.5.2c for an illustration of the evolution of L_t. The random patch diffusion can also be applied to mask generation, where the only condition is the context prompt. This method can generate segmentation masks of arbitrary sizes with the correlation length bounded by two times the training mask image size.

Parallelization The sampling method proposed has several advantages. In [264] each sequential patch is outpainted from the previous one with 50% of the pixels shared. Here, the randomization eventually leads to every possible overlap with the neighboring patches. This introduces a longer pixel correlation across the whole generated image, avoiding artifacts due to tiling. In Figure 5.3, we show that the number of steps in the whole large image generation process is drastically reduced with the random patching method with respect to the sliding window one. Moreover, in the sliding sampling method, the model can be paralleled only 2 or 4 times, depending if we are outpainting the image horizontally or on both axis. In our approach, we can parallelize the

Algorithm 1 DiffInfinite training

Repeat

Randomly train on labelled or unlabelled data with probability p_{unc},
$u \sim Uniform[0, 1]$

$$(z_0, c) = \begin{cases} (z_0, \varnothing) \sim q_0(z_0) & \text{if } u \geq p_{unc} \\ (z_0, c) \sim q_1(z_0, c) & \text{otherwise} \end{cases}$$

Sample random time step
$t \sim Uniform\{1, ..., T\}$
Sample noise, $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
Corrupt data, $z_t = \gamma_t x_0 + \sigma_t \epsilon$
Take gradient descent step:
$\nabla_\theta \|\epsilon - \epsilon_\theta(z_t, t, c)\|^2$
until converged

Algorithm 2 DiffInfinite sampling

Input: High-level segmentation mask $M \in \mathbb{R}^D$ and learned model ϵ_θ
Output: Synthetic image X with the mask size
Initialization:
$\mathbf{y}_T \sim \mathcal{N}(0, \mathbf{I})$, index set $I_0 = \emptyset$ and time state tensor $L_T \equiv T$
Repeat

for $t \in \{T - 1, ...0\}$ **do**
  while $\cup_{j=0}^{m} I_j \neq \{1, ..., D\}$ **do**
    $m \leftarrow m + 1$
    Select randomly $proj_{I_m} \in C \setminus \{proj_{I_1}, ..., proj_{I_{m-1}}\}$
    Crop $z_t^m = proj_{I_m}(y_t)$
    for all conditions $n \in \{1, ..., N\}$ **do**
      DDIM sampling with classifier-free guidance
      $z_{t-1}^m | c_n \sim p_{\theta,t}(z | z_t^m, c_n)$
    end for
    for all indices $i \in I_m$ **do**
      if $i \notin I_j$ for all $j < m$ **then**
        $[y_{t-1}]_i \leftarrow [z_{t-1}^m | M_i]_l$
        $proj_{I_m}(L_t)_i \leftarrow t$
        such that $[proj_{I_m}(y_{t-1})]_l = [z_{t-1}^m | M_i]_l$
      end if
    end for
  end while
end for
$X \leftarrow \Psi(y_0)$

sampling up to the computational resource limit.

Hann windows decoding After the diffusion model samples Z in the VAE's latent space, the latent variable Z needs to be decoded into the pixel space. However, due to computational

constraints, it is not feasible to decode Z all at once. Therefore, we tile it into smaller patches. Decoding smaller patches would introduce tiling effects. In order to reduce edge artifacts, we used an overlapping window method using Hann windows as weights [166]. In Fig. 5.4, we tile the image in four different configurations such that the edges and corners are overlapping, and then we perform a weighted sum over the upsampled outputs.

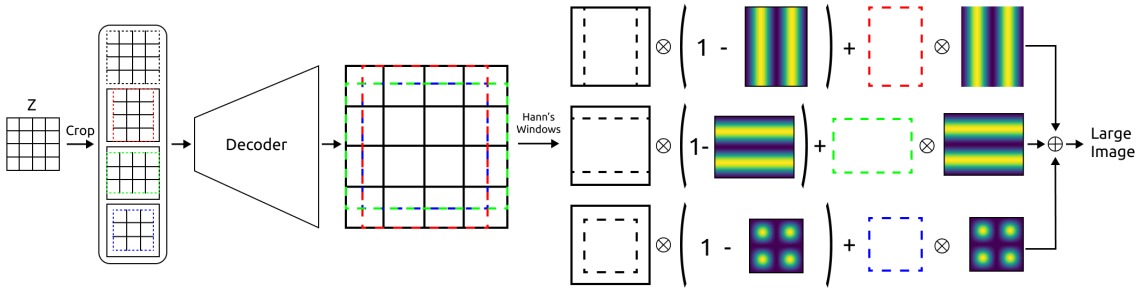

Figure 5.4: Hann window overlapping illustration.

5.3 Training details

Training on images The core model used in the diffusion process is a U-Net[2]. Every U-net's block is composed of two ResNet blocks, a cross-attention layer and a normalization layer. On each ResNet block, we feed the output x^l of the previous block l, the time t and the label c_i. The cross-attention is performed using the mask corresponding to the label c_i as query and the input x^l as key and value.

Training on masks We replace the cross-attention with a linear self-attention layer for mask generation. Here, the model is conditioned with binary labels $\{0, 1\}$, where 0 corresponds to adenocarcinoma and 1 corresponds to squamous cell carcinoma. The masks of size 512×512 is first downsampled to size $1 \times 128 \times 128$. We stack the downsampled mask to the size $(3, 128, 128)$ to make it compatible with a pre-trained VAE[3]. We repeated the same training for the larger masks 2048×2048, downsampling them to 128×128 as well.

Mask cleaning The diffusion model samples a latent mask in the VAE's latent space. After mapping the latent mask back to the pixel space we average over the channels to have a mask with one channel and round the pixel values to the integers $\{0, 1, ..., num\_values\}$. Since we note some boundary artifacts between regions of different values we first apply a method from skimage

[2]Baseline, `https://github.com/lucidrains/classifier-free-guidance-pytorch`
[3]`https://huggingface.co/stabilityai/stable-diffusion-2`

Table 5.1: Details of the parameters used for training

	Model parameters image generation
Image X shape	(3,512,512)
Latent Y shape	(4,64,64)
VAE	stabilityai/stable-diffusion-2-base
Num classes	5 and 10
Loss	L2
Diffusion steps	1000
Training steps	250000
Sampling steps	250
Heads	4
Heads channels	32
Attention resolution	32,16,8
Num Resblocks	2
Probability p_{unc}	0.5
Batch size	128
Number of workers	32
GPUs Training	4 NVIDIA GeForce RTX 3090, 24Gb each
GPUs Inference	1 NVIDIA GeForce RTX 3090
Training time	$\sim$ 1 week
Optimizer	Adam
Scheduler	OneCycleLR(max lr=1e-4)

	Model parameters mask generation
Mask M shape	(3,128,128)
Latent Y shape	(4,16,16)
VAE (repo id)	stabilityai/stable-diffusion-2-base
Num classes	2
Loss	L2
Diffusion steps	1000
Training steps	100000
Sampling steps	250
Heads	4
Heads channels	32
Attention resolution	32,16,8
Num Resblocks	2
Probability p_{unc}	0.5
Batch size	64
Number of workers	1
GPUs Training	2 Ampere A100, 40Gb each
GPUs Inference	1 NVIDIA GeForce RTX 3090
Training time	$\sim$ 4 hours
Optimizer	Adam
Scheduler	OneCycleLR(max lr=1e-4)

[4] to find these boundary artifacts and replace it by 0, corresponding to unknown area. Before resizing the mask to the full size, we apply a minpooling operation to erase labelled regions of small magnitude and replace it as well with unknowns.

5.3.1 Histological dataset

The real-world data used for training the generative model consisted of 41 high-resolution Hematoxylin and Eosin (H&E)-stained whole slide images of lung tissue biopsies from different cancer patients. These images were evenly split between cases diagnosed with adenocarcinoma of the

[4] `https://scikit-image.org/docs/stable/api/skimage.segmentation.html`

Table 5.2: Details of the histological dataset

	Histological dataset
Number of whole slide images	41
Image type	H&E-stained whole slide images
Whole slide image size	$\sim 100,000 \times 100,000$
Magnification	40x
Image scanner	Aperio scanner
Number of annotation categories	40
Annotation distribution	37% Carcinoma, 36% Stroma, 3.5% Necrosis, 23.5% Other
Resolution	0.5 microns per pixel
Number of patches (image training)	4,781
Patch size	512×512 px
Train/Test split	90/10 stratified by annotation categories
Number of patches (large mask training)	1,183
Patch size	2048×2048 px

lung and squamous cell carcinoma, representing the two most common sub-types in lung cancer. The images were scanned on an Aperio scanner at a resolution of 0.25 microns per pixel (40x). Different classes used for conditioning were annotated digitally by a pathologist using an apple pencil with the instruction to clearly demarcate boundaries between tissue regions. The pathologist could choose from a list of 40 distinct annotation categories, aiming to cover all possible annotation requirements. 37% of the annotations belonged to the Carcinoma category, 36% to Stroma, 3.5% to Necrosis and the remaining 23.5% to other smaller categories summarized as Other. All data handling was performed in strict accordance with privacy regulations and ethical standards, ensuring the protection of patient information at all times. For training the diffusion model, we utilized a patch dataset derived from expert annotations. In total, the dataset contained 4,781 patches of size 512×512 px. The dataset was split into train/ test sets with a ratio of 90/ 10, stratified by annotation categories. This test split was used for the generative model as well as to evaluate the downstream task. We also tiled the slides with size 2048×2048 from the same annotations, extracting 1,183 patches. These masks are used for training the mask generative model.

5.4 Synthetic data visualisation

In this section, we visually demonstrate the quality of the synthetic data generated. As an illustration of the mask-image guidance, we show the results on the patch level such it allows a more detailed examination of the model's synthetic data generation capabilities. We provide also synthetic data at different resolutions, showing the model's reliability at different scales. Inpainting examples provide additional validation of the data's authenticity.

Mask-image pairs In Fig. 5.5, we show the control on the mask-image generation for 512×512 patches. The *Unknown* class corresponds to pixels which were not annotated due to a sparse annotation strategy. The images show that the cross-attention layer controls mask conditioning effectively. As a proof of concept, we generated images at different scales (512×512,1024×1024,2048×2048) with a simple squares mask (see Fig. 5.6). In Figure 5.7, we see that for the small masks of size 512×512, the frequency of labels in the real masks are reproduced well by the generated masks. For the large masks of size 2048×2048, the labels that occur most frequently in the real masks are underrepresented in the generated masks, while all other labels are overrepresented in the generated masks.

Random patch advantages Sampling with the random patch (RP) method leads to several benefits compared to the sliding windows (SW) approach (see Fig. 5.8). First, the sliding window

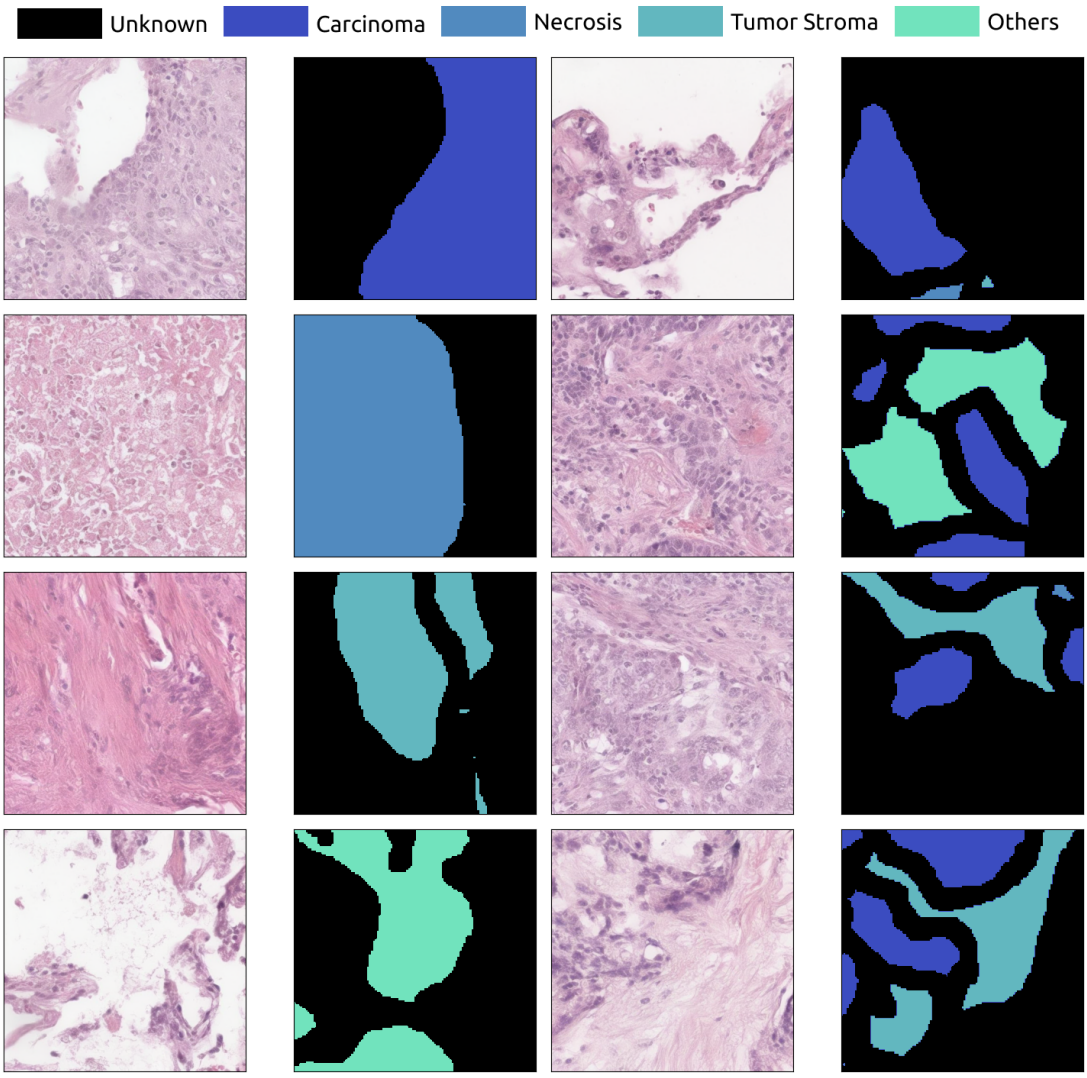

Figure 5.5: Generated images conditioned on the synthetic segmentation masks.

method starts from the centre of the image and outpaints in four directions. As a consequence, the model needs to condition on previously generated areas, leading to blurriness on the farther pixels. With the random patch method, every area is conditioned only on its neighbour, avoiding error propagation. Moreover, while SWs have only information on the closest neighbour, RPs consider long-range correlations. On every diffusion step, we have every possible overlap between near patches, extending correlation lengths to twice the diffusion model output size. Furthermore, this random overlap avoids any tiling effect.

Inpainting Using the segmentation images and masks of the test set, we inpainted the annotated areas with the same corresponding class (see Fig. 5.9). We show that the model generates new content with respect to the real one. We run the same experiment by inpainting one area with different classes (see Fig. 5.10). Keeping the same seed, we show how the generation changes while ω increases. By increasing ω, we enhance the diversity at the cost of losing some

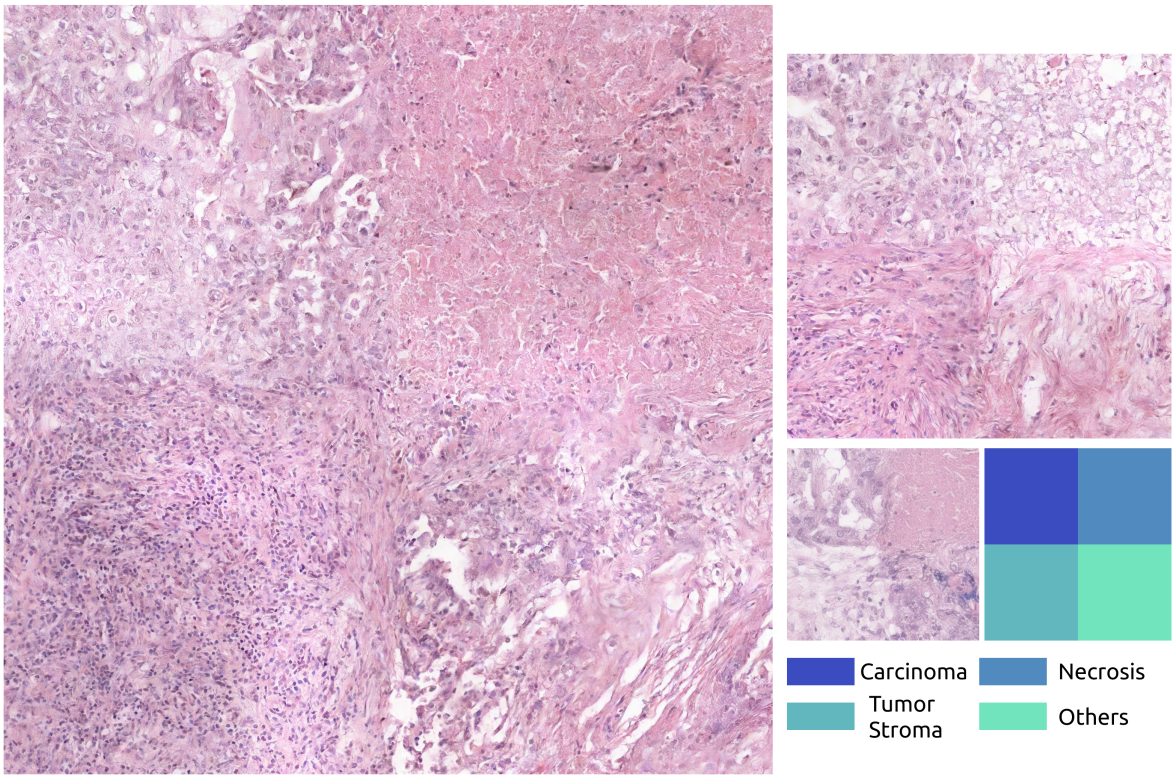

Figure 5.6: Conditioning visualization. All the images are conditioned with the squared mask shown. Left) 2048 × 2048 image. Top-Right) 1024 × 1024 image. Bottom-Right) 512 × 512 images.

conditioning.

5.5 Data assessment

To assess synthetic images for medical image analysis, we need to take various dimensions of data assessment into account. We extend traditional metrics from the natural image community with qualitative and quantitative assessments specific to the medical context. For the qualitative analysis, a team of pathologists evaluated the images for histological plausibility. The quantitative assessment entailed a proof-of-concept that a model can learn sensible features from the synthetically generated image patches for a relevant downstream task. As data protection is highly relevant regarding patient data, we performed evaluations to rule out the memorization effects of the generative model.

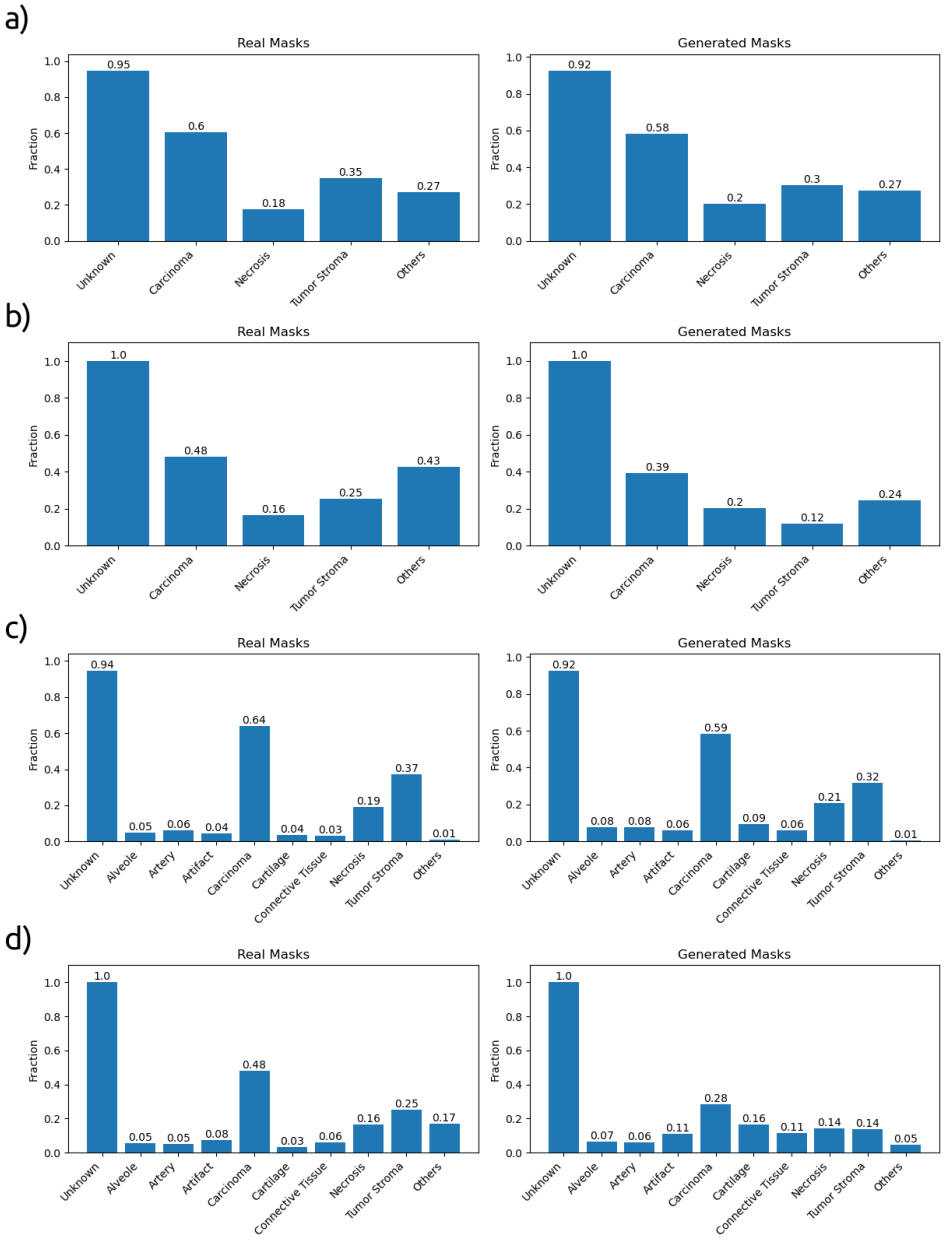

Figure 5.7: Fraction of label appearance in the segmentation masks with 5 classes in a,b) and 10 in c,d). Fractions estimated over a) 4205 real masks of size 512×512 and 20719 generated masks, b) 1183 real masks of size 2048×2048 and 22705 generated masks, c) 4205 real masks of size 512×512 and 22604 generated masks, d) 1183 real masks of size 2048×2048 and 22560 generated masks.

5.5.1 Traditional fidelity

In this section, we define and apply the metrics used to assess the fidelity and degree of memorization of DiffInfinite. Following the notation of Section 5.2.1, denote by $X_r \sim \mathcal{X}_r$ the real data distribution and by $X_g = \Psi(\hat{Y}) \sim \mathcal{X}_g$ the distribution from which the generative model samples. For the quantitative evaluation of the quality and the coverage of the data generated by DiffInfinite

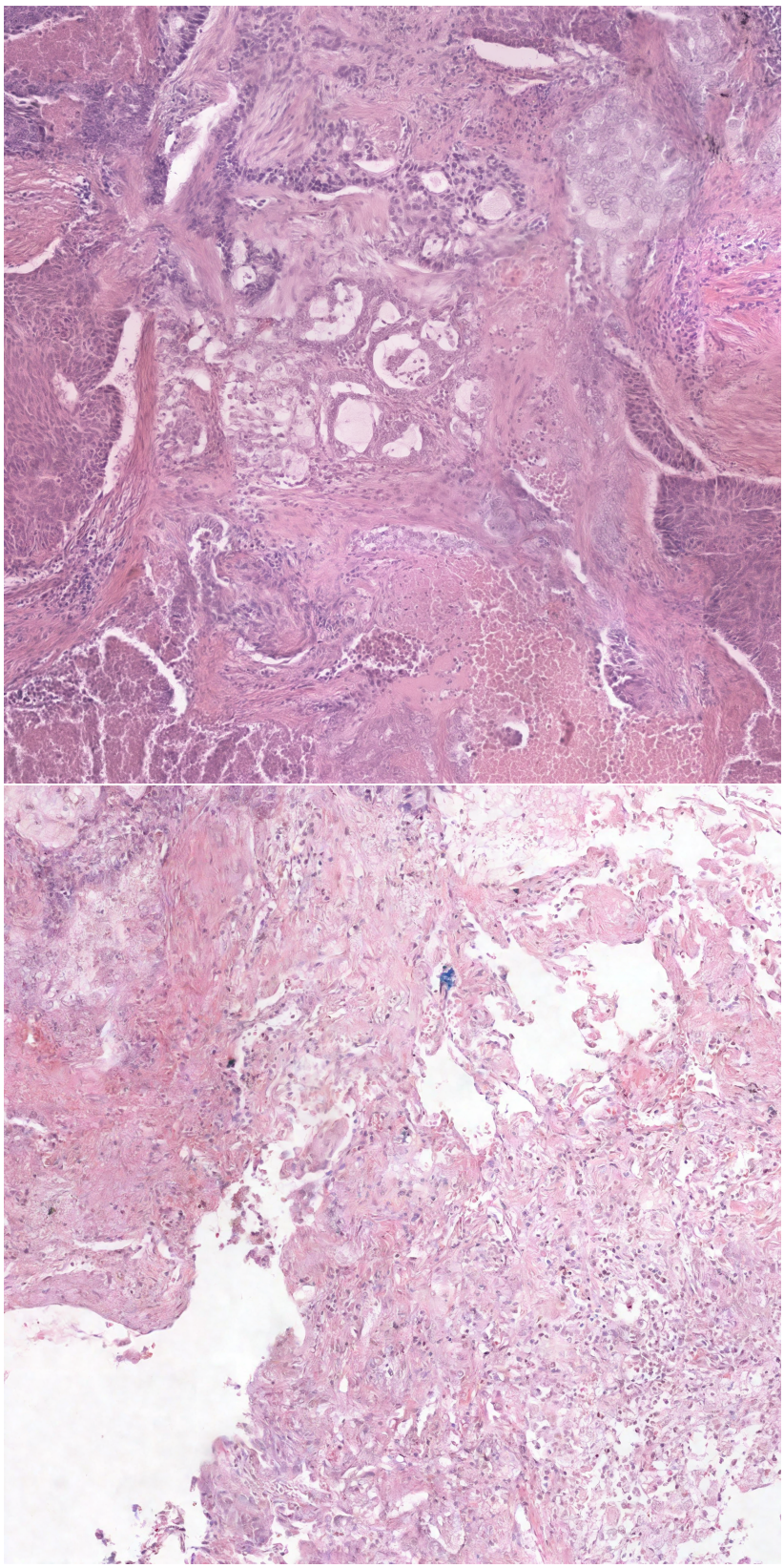

Figure 5.8: Comparison of different methods to generate large images (2048 × 2048). top) DiffCollage image generation using the grid graph [264]. Bottom) DiffInfinite (ours) image generation using the proposed random patch sampling.

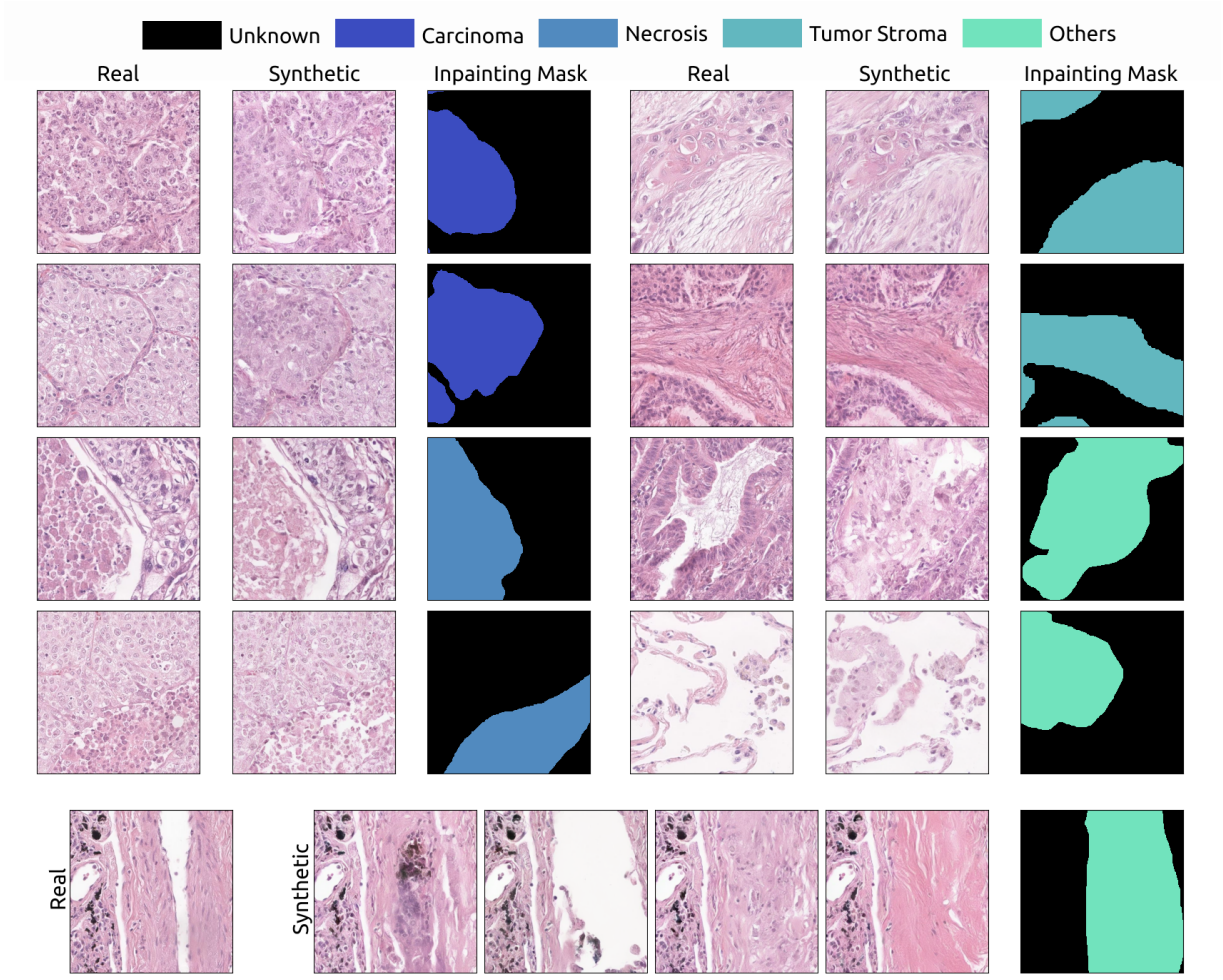

Figure 5.9: Inpainting test data with the corresponding masks. Top) Inpainting for different labels. Bottom) Different inpainted synthetic areas for the same mask.

we use the improved recall and improved precision, while the rate of DiffInfinite to innovate a new sample is approximated by the authenticity score. Finally, to test DiffInfinite for data-copying we compute the C_T score.

Improved recall and improved precision [110] A pre-trained classifier[5] maps the samples into a high-dimensional feature space resulting in the feature vectors $\mathbf{\Phi}_r$ and $\mathbf{\Phi}_g$. For $\mathbf{\Phi} \in \{\mathbf{\Phi}_r, \mathbf{\Phi}_g\}$ denote by $NN_k(\phi', \mathbf{\Phi})$ the kth nearest feature vector of ϕ' from set $\mathbf{\Phi}$ and define the binary function

$$f(\phi, \mathbf{\Phi}) = \begin{cases} 1, & \textit{if } \|\phi - \phi'\|_2 \leq \|\phi' - NN_k(\phi', \mathbf{\Phi})\|_2 \textit{ for at least one } \phi' \in \mathbf{\Phi} \\ 0, & \textit{otherwise} \end{cases} \tag{5.10}$$

[5]We use the pre-trained VGG-16 classifier from `https://github.com/blandocs/improved-precision-and-recall-metric-pytorch`.

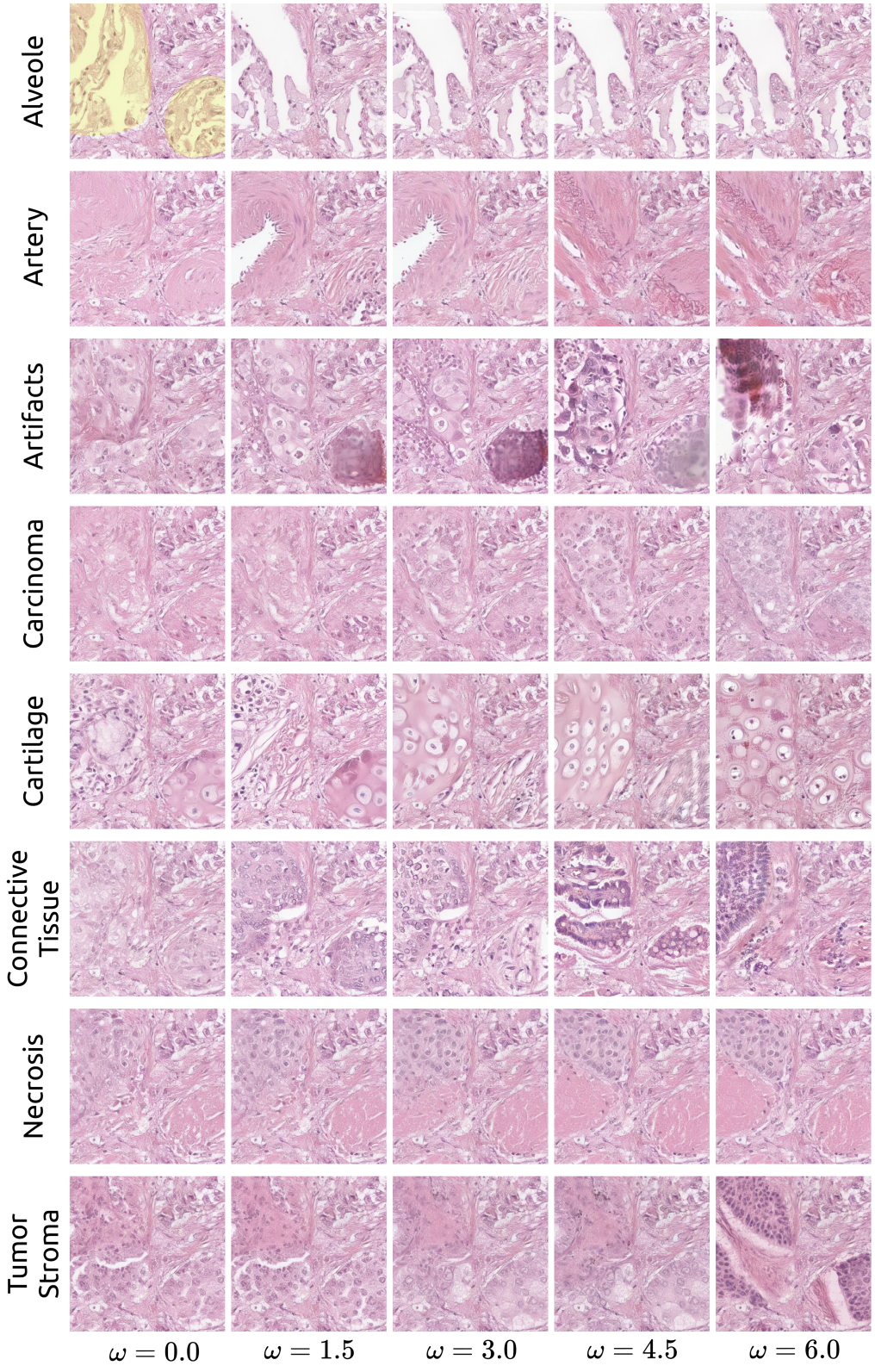

Figure 5.10: Proof of concept with inpainting. We inpainted the same base image with different classes and different strengths of conditioning (small ω corresponding to less diversity). The corresponding inpainting mask is displayed as an overlay on the top left patch (in yellow).

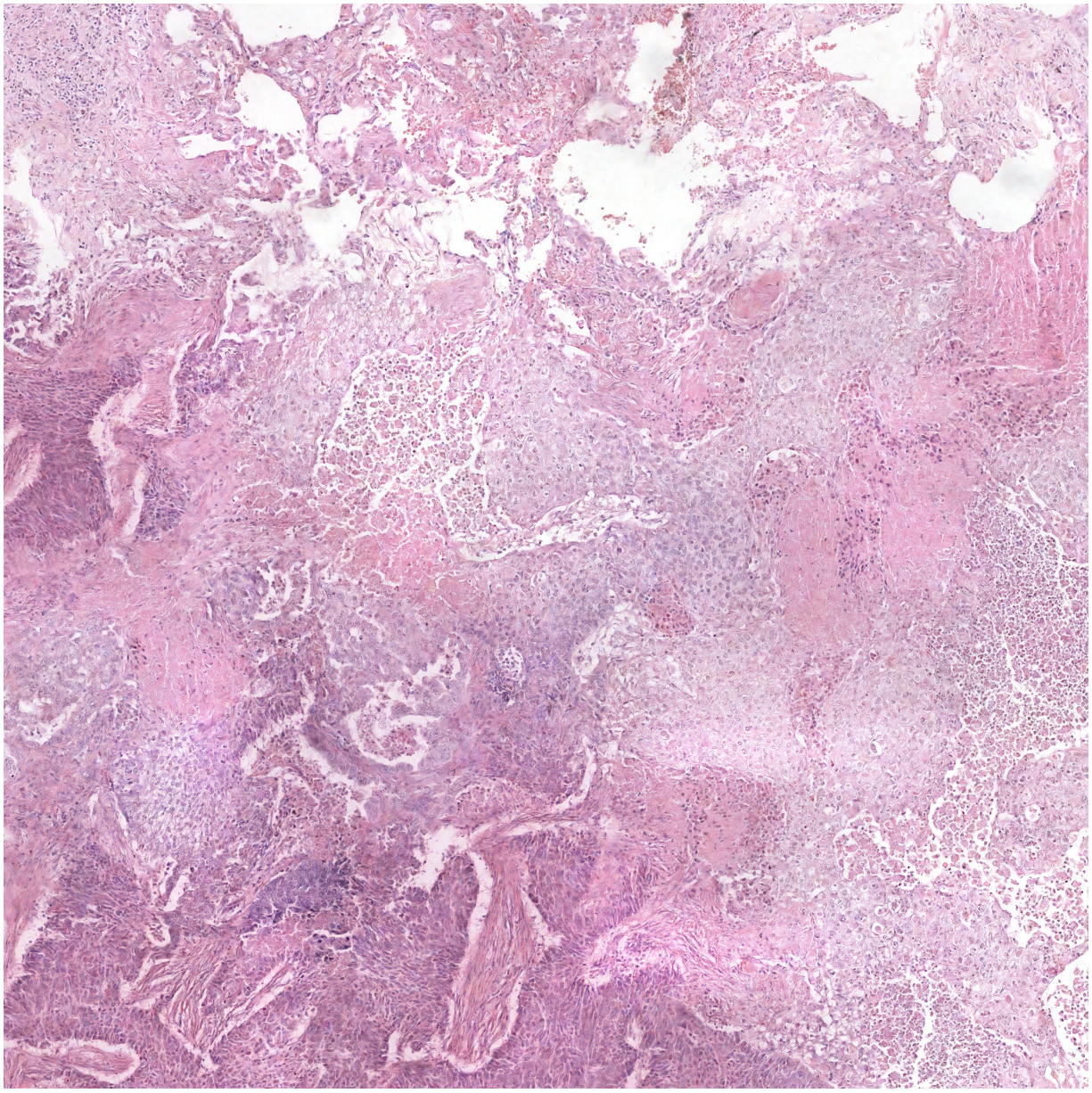

Figure 5.11: Large content synthetic image with a size of 4096×4096 pixels.

that identifies whether a given sample $\boldsymbol{\phi}$ is within the estimated manifold volume of $\boldsymbol{\Phi}$ corresponding to NN_k. To measure the similarity of $\boldsymbol{\Phi}_g$ to the estimated manifold of the real images, define improved precision (IP) by

$$\text{precision}(\boldsymbol{\Phi}_r, \boldsymbol{\Phi}_g) = \frac{1}{|\boldsymbol{\Phi}_g|} \sum_{\boldsymbol{\phi}_g \in \boldsymbol{\Phi}_g} f\left(\boldsymbol{\phi}_g, \boldsymbol{\Phi}_r\right) \tag{5.11}$$

and to measure the similarity of $\boldsymbol{\Phi}_r$ to the estimated manifold of the generated images, define improved recall (IR) by

$$\text{recall}(\boldsymbol{\Phi}_r, \boldsymbol{\Phi}_g) = \frac{1}{|\boldsymbol{\Phi}_r|} \sum_{\boldsymbol{\phi}_r \in \boldsymbol{\Phi}_r} f\left(\boldsymbol{\phi}_r, \boldsymbol{\Phi}_g\right). \tag{5.12}$$

Table 5.3: Quantitative memorization metrics for the variants of DiffInfinite described in Section 5.5.1. For consistency, we consider all methods from Table 5.4 in our evalution, including the comparison to DiffCollage. For the methods that output a large image of size 2048 we consider the *tiled* patches resulting in 16 patches per large image and the *resized* image resulting in 200 images of size 512×512. The best results are highlighted in bold.

	$\boldsymbol{A}\uparrow$		$\boldsymbol{C}_T \Downarrow\uparrow$	
	tiled	*resized*	*tiled*	*resized*
DiffCollage	**0.89**	0.97	11.02	**7.00**
DiffInfinite (a)	0.86	-	4.99	-
DiffInfinite (b)	0.86	0.97	**3.29**	8.11
DiffInfinite (c)	0.86	**0.98**	9.61	11.56
DiffInfinite (b) & (c)	0.87	0.95	5.31	10.96

Authenticity score [4] For the definition of the authenticity score $A \in [0, 1]$, assume that the probability measure $\mathbb{P}_g$ corresponding to $\mathcal{X}_g$ is a mixture of the probability measures

$$\mathbb{P}_g = A \cdot \mathbb{P}'_g + (1 - A) \cdot \delta_{g,\epsilon}, \tag{5.13}$$

where $\mathbb{P}'_g$ characterizes the generative distribution, excluding synthetic samples that are duplicates of training samples and $\delta_{g,\epsilon} = \delta_g * \mathcal{N}(0, \epsilon^2)$ is the noisy distribution over training data implied by an unknown discrete probability measure δ_g placing probability mass on each data point used for training.

$\boldsymbol{C}_T$ score [144] . For a set of training images $\mathcal{D}_{train} = \{x_1, ..., x_k | x_i \sim \mathcal{X}_r\}$ and $y \in \mathbb{R}^{KD}$ define the distance measure $d(y) = \min_{x \in \mathcal{D}_{train}} \|x - y\|_2^2$. Denote by $L(\mathcal{V})$ the one dimensional distribution $d(V)$ of any random variable $V \sim \mathcal{V}$ with the same instance space as $\mathcal{X}_r$. For the test set of the real data $\mathcal{D}_{test} = \{y_1, ..., y_n \mid y_i \sim \mathcal{X}_r\}$, define the fraction $P_n(\pi) = \frac{1}{n} \left| \{y \in \mathcal{D}_{test} \mid y \in \pi \in \Pi\} \right|$ of test points in cell $\pi \in \Pi$, where Π is a partition of $\mathbb{R}^{KD}$ resulting from applying the k-means algorithm on $\mathcal{D}_{train}$. Similar for a set of generated images $\mathcal{D}_{gen} = \{\hat{x}_1, ..., \hat{x}_m\}$ sampled from $\mathcal{X}_g$, define the fraction $Q_n(\pi)$ of generated samples in cell $\pi \in \Pi$. Denote by Z_U the z-scored Mann-Whitney U statistic from Section 3.1 of [144] with $L_\pi(\mathcal{D}) = \{d(x) \mid x \in \mathcal{D}, \pi \in \Pi\}$ for $\mathcal{D} \in \{\mathcal{D}_{test}, \mathcal{D}_{gen}\}$ and let Π_τ be the set of all cells in Π for which $Q_m(\pi) \geq \tau$ holds true. The $\boldsymbol{C}_T$ score is finally defined as the average

$$\boldsymbol{C}_T \left(P_n, Q_m\right) = \frac{\sum_{\pi \in \Pi_\tau} P_n(\pi) Z_U \left(L_\pi(P_n), L_\pi(Q_m); T\right)}{\sum_{\pi \in \Pi_\tau} P_n(\pi)}. \tag{5.14}$$

across all cells represented by $\mathcal{X}_g$.

Quantitative assessment We evaluate the fidelity of synthetic 512×512 images by calculating Improved Precision (IP) and Improved Recall (IR) metrics between 10240 real and synthetic images [110].[6] The IP evaluates synthetic data quality, while the IR measures data coverage. Despite their unsuitability for histological data [111, 11], Frechet-Inception Distance (FID) and Inception Score (IS) [78, 194] are reported for comparison with [145] and Shrivastava and Fletcher [209].[7] In Table 5.4 (left), we report an IP of 0.94 and an IR of 0.70, indicating good quality and coverage of the generated samples. However, we note that these metrics are only somewhat comparable due to the different types of images generated by MorphDiffusion [145] and NASDM [209]. For the large images of size 2048×2048, we rely solely on the IP and IR for quantitative evaluation due to the limited number of 200 generated large images. As shown in Figure 3(a) of [110], FID is unsuitable for evaluating such a small sample size, while IP and IR are more reliable. In Table 5.4 (right), we find that generating images first results in slightly higher IR, while generating the mask first achieves an IP of 0.98. For the sake of completeness we also report the scores then combining the two datasets. To compare our method to DiffCollage we generate 200 images using [264]. DiffInfinite performs better than DiffCollage wrt. to IP and IR. The drop of IR to 0.22 might be a result of the tiling artifacts observable in the LHS of Figure 5.8.

Table 5.4: Metrics to quantitatively evaluate the quality of the generated images. Left: scores for images of size 512×512. DiffInfinite (a) first generates a mask and secondly an image following Section 5.2.1. Right: scores for real and generated images of size 2048×2048 resized to 512×512. All methods use the same model trained on small patches of size 512×512. DiffCollage corresponds to the method proposed in [264]. DiffInfinite (b) uses the real masks, while DiffInfinite (c) first generates a mask and secondly the large image. DiffInfinite (b) & (c) refers to the mixture of the generated dataset from DiffInfinite (b) and DiffInfinite (c). The best results are highlighted in bold.

	IP ↑	**IR ↑**	**IS ↑**	**FID ↓**
Morph-Diffusion [145]	0.26	**0.85**	2.1	20.1
NASDM [209]	-	-	**2.7**	**15.7**
DiffInfinite (a)	**0.94**	0.70	**2.7**	26.7

	IP ↑	**IR ↑**
DiffCollage	0.94	0.22
DiffInfinite (b)	0.95	**0.48**
DiffInfinite (c)	**0.98**	0.44
DiffInfinite (b) & (c)	**0.98**	0.33

Metrics' limitations The concern about whether these metrics are the most meaningful for a specific problem domain is valid. Current research is increasingly focused on understanding the distinction between real and synthetic data from a quantitative perspective. Most metrics in use today, including FID and IS serve as benchmarks, while state-of-the-art metrics, including improved precision, improved recall, the C_T score, and the authenticity score provide valuable insights into the sampled distribution of generative models, but they should be considered complementary to

[6] https://github.com/blandocs/improved-precision-and-recall-metric-pytorch
[7] https://github.com/toshas/torch-fidelity

the specific usage of synthetic data. While these advanced metrics offer a robust assessment of the generative model's performance, they can be supplemented with domain-specific evaluations to ensure a comprehensive analysis. For instance, in the context of histopathology, traditional metrics such as cell counting and analysis of tissue architecture can provide additional relevant insights. These classical metrics help to validate that the generated data not only meets quantitative benchmarks but also aligns with the practical requirements of the domain. In our case, we specifically focus on the utility of synthetic data for downstream tasks (see Sec. 5.5.3). By conducting experiments that leverage synthetic data in real-world applications, we can directly quantify its usefulness and effectiveness. This approach ensures that our evaluation is not only grounded in advanced quantitative metrics but also reflects the practical value of the synthetic data in our specific problem domain.

5.5.2 Domain experts' assessment

To assess the histological plausibility of our generated images, we conducted a survey with a cohort of ten experienced pathologists, averaging 8.7 years of professional tenure. The pathologists were tasked with differentiating between our synthetized images and real image patches extracted from whole slide images. We included both small patches (512×512 px) as commonly used for downstream tasks as well as large patches (2048×2048 px). Including large patches enabled us to additionally evaluate the modelled long-range correlations in terms of transitions between tissue types as well as growth patterns which are usually not observable on the smaller patch sizes but essential in histopathology. In total the survey contained 60 images, in equal parts synthetic and real images as well as small and large patches. Fig.5.12 shows the setup of the survey. The presented images were shown in randomized order. The overall ability of pathologists to discern between real and synthetic images was modest, with an accuracy of 63%, and an average reported confidence level of 2.22 on a 1-7 Likert scale. While we observed high inter-rater variance, there was no clear correlation between experience and accuracy ($r(8) = 0.069$, $p=0.850$), nor between confidence level and accuracy ($r(8) = 0.446$, $p=0.197$). Furthermore there was no significant correlation between the participants' completion time of the survey and the number of correct responses ($r(8) = -0.08$, $p=0.826$).

Surprisingly, we found a similar performance for both, real and synthetic images. This indicates that, while clinical practice is mostly based on visual assessment, it is not a common task for pathologists to be restricted to parts of the whole slide image only. More detailed visualizations of the individual scores can be found in Fig. 5.14. Besides this satisfactory result, explore the limitations of our method by assessing the nuanced differences pathologists observed between synthetic and real images. While overall the structure and features seemed similar and hard to discern, they sometimes reported regions of inconsistent patterns, overly homogeneous chromatin

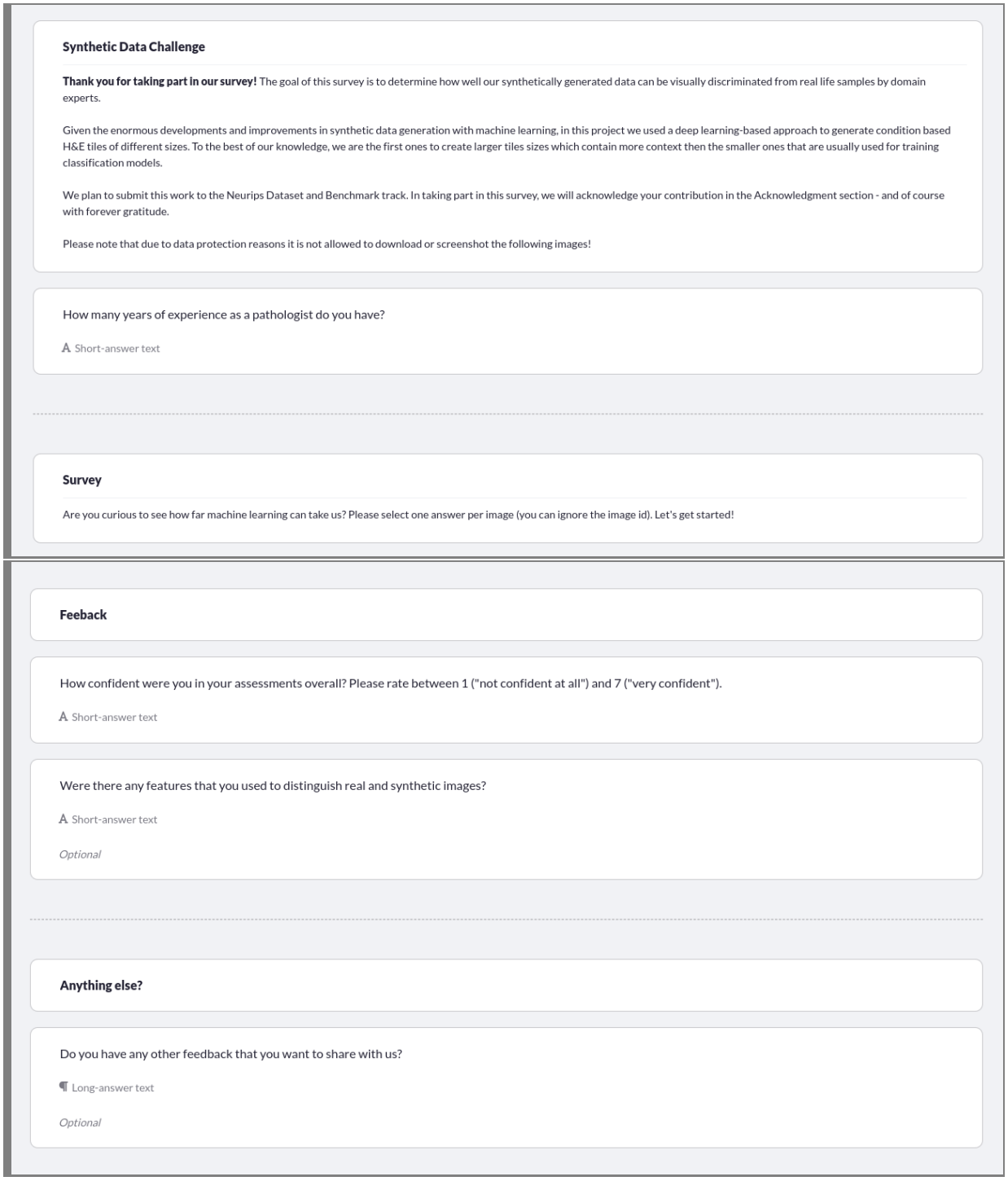

Figure 5.12: Survey interface for the domain expert assessment of real versus synthetic data.

in some of the synthetic nuclei, peculiarities in cellular and intercellular structures, and aesthetic elements. These seemed to be especially pronounced in tumorous regions where sometimes the tissue architecture appeared exaggerated, the transition to stroma or surrounding tissue was too abrupt and some cells lacked distinguishable nucleoli or cytoplasm. We attribute the nuanced effect of larger image size on the accuracy on this observation (cf. Fig. 5.14C). Overall the finding of the conducted survey demonstrates how complex the task of distinguishing between real and

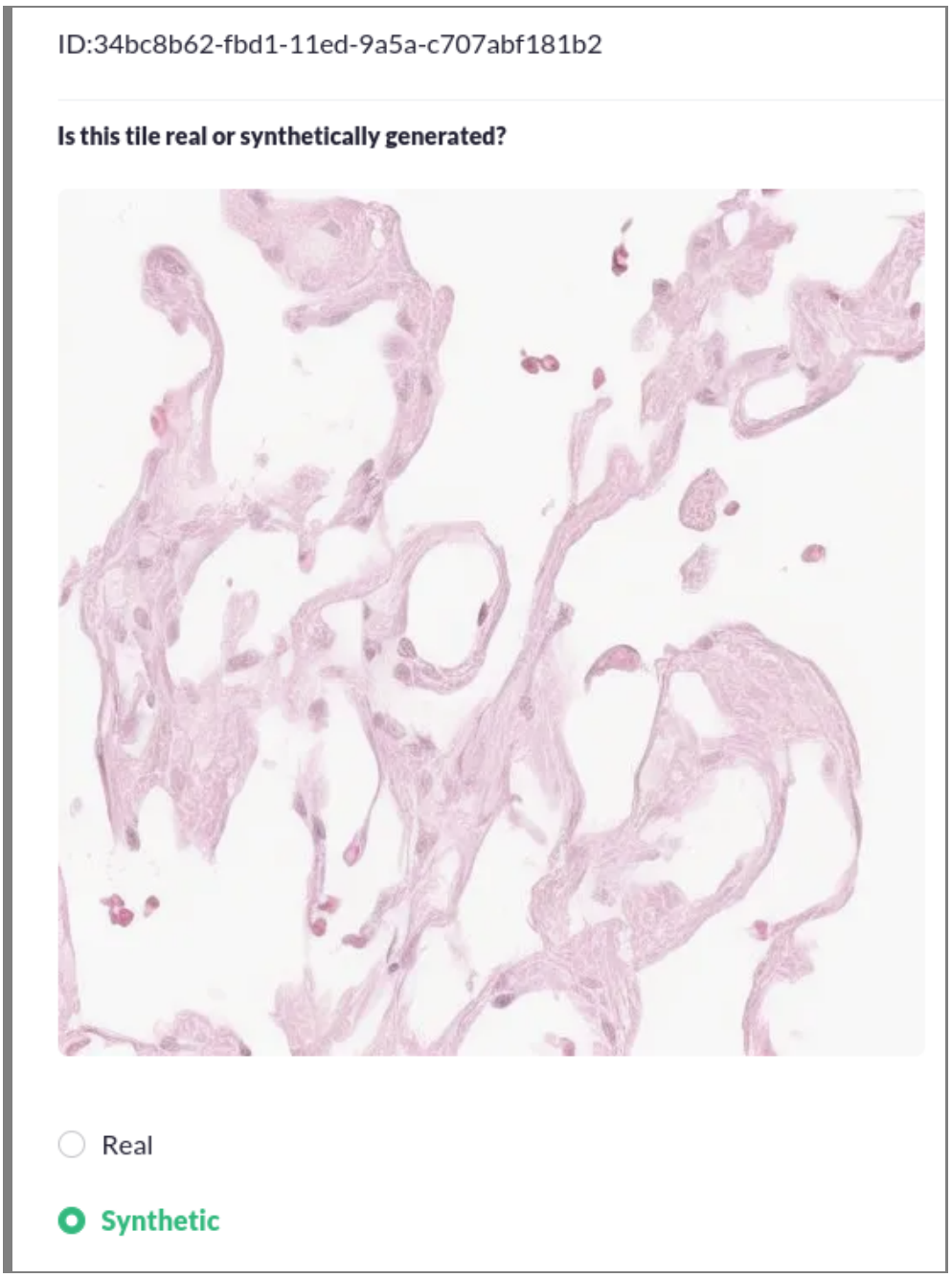

Figure 5.13: Example of data showed during the survey with domain experts.

synthetically generated data is even for experienced pathologists while still highlighting potential areas to improve the generative model.

5.5.3 Synthetic data for downstream tasks

A major interest in the availability of high quality labeled synthetic images is their use in downstream digital pathology applications. In this area, two primary challenges are the binary classifi-

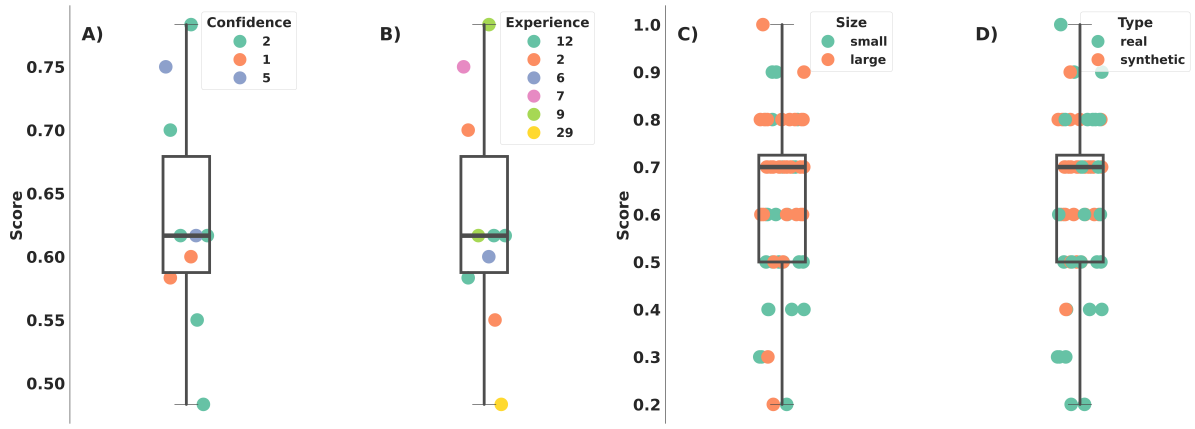

Figure 5.14: Results of the survey. Left: Accuracy per pathologist, color-coded by subjective confidence level (**A**) and years of experience (**B**). Right: Average accuracy across pathologists for each image patch, color-coded by path-size (**C**) and veracity (**D**).

Table 5.5: Zero-shot evaluation results of the downstream tasks, encompass both classification and segmentation scenarios. We employed three distinct models for each scenario: The first, "Trained Real", was trained using real data (in-house IH1), which also served as the training set for DiffInfinite. The second, "Trained Synthetic", was trained using samples generated from DiffInfinite, and the third, "Trained Augmented", utilized a combination of real and synthetic data. Our evaluation extends across separate lung cohorts (internal datasets IH2 and IH3) and additional indications (external datasets NCT, CRC, PCam), with varying degrees of data drift introduced. The best results are highlighted in bold.

	IH1	**IH2**	**IH3**	**NCT-100K**	**CRC-7K**	**PCam-327K**
Drift components	-	Patient change Different center		Patient change Different center Indication change		Patient change Different center Indication change Lower resolution
Trained Real	0.846 ± 0.005	0.733 ± 0.021	0.598 ± 0.049	**0.857** ± 0.009	**0.822** ± 0.034	0.628 ± 0.035
Trained Synthetic	0.747 ± 0.025	**0.753** ± 0.005	**0.699** ± 0.002	0.796 ± 0.023	0.753 ± 0.038	0.628 ± 0.012
Trained Augmented	**0.852** ± 0.007	0.732 ± 0.027	0.637 ± 0.025	0.847 ± 0.044	0.811 ± 0.057	**0.641** ± 0.035

(a) Classification results

	IH2
Trained Real	0.614 ± 0.009
Trained Synthetic	0.471 ± 0.039
Trained Augmented	**0.710** ± 0.021

(b) Segmentation results

cation of images into cancerous or healthy tissues and the segmentation of distinct tissue areas in the tumor microenvironment. The unique ability of our technique to generate images of different cancer subtypes through the context prompt as well as the ability to create new segmentation masks and their corresponding H&E images specifically addresses these two challenges. Notably, expert annotations are costly and time-consuming to acquire thus emphasizing the benefits of being able to train on purely synthetic datasets or augmenting annotated data in the low data

regime. To showcase these two usecases we performed a series of experiments in both classification and segmentation settings. For all experiments, we trained a baseline classier on a relatively small number of expert annotations IH1 (#patches = 3726)) — the same that were used to train DiffInfinite — and additionally trained one model purely on synthetic data (IH1-S, #patches = 9974, $\omega = 0$), and one model on the real data augmented with the synthetic images.

Binary classification To generate target labels for the classification experiments, we simplified the segmentation challenge by categorizing patches with at least 0.05% of pixels labelled as 'Carcinoma' in the segmentation masks as 'Carcinoma'. All other patches were labelled 'Non-Carcinoma'. We evaluated all three classification models on several out-of-distribution datasets. We utilized two proprietary datasets (from the same cancer type with similar attributes but from distinct patient groups: IH1 (# patients=13, # patches=704) and IH3 (# patients=2, # patches=2817). Moreover, we assessed the models using two public datasets (NCK-CRC [98] and PatchCamelyon [242]), both representing tissue from different organs with distinct morphologies. Our findings, summarized in Table 5.5a, suggest that a classifier's out-of-distribution performance, trained with limited sample size and morphological diversity, can vary significantly (ranging from 0.628 to 0.857 balanced accuracy). This variability cannot be attributed solely to morphology but may also be influenced by factors such as resolution and variations in scanning and staining techniques. Training exclusively with a larger set of synthetic images can enhance performance on some datasets (specifically IH2 and IH3), underscoring the advantages of leveraging the full training data in a semi-supervised manner within the generative model. Incorporating synthetic data as an augmentation to real data not only prevents the classifier's performance decline, as seen on NCT-CRC and Patchcamelyon, on similar datasets but also bolsters its efficiency on more distinct ones.

Multi-class segmentation For the more challenging segmentation task we again trained three segmentation models to differentiate between carcinoma, stroma, necrosis, and a miscellaneous class that included all other tissue types, such as artifacts. The baseline performance of the real data model on a distinct group of lung patients (dataset IH2) of a F_1 score of 0.614 ± 0.009 (across three random seeds) highlights the difficulty of generalizing out of distribution in this task. While the purely synthetic model was not able to fully recover the baseline performance (0.471 ± 0.039), augmenting the small annotated dataset with synthetic data enhanced predictive performance to an F_1 score of 0.710 ± 0.021. This boost of 10 percentage points in performance demonstrates that the synthetic data provide new, relevant information to the downstream task. In summary, our findings demonstrate the feasibility of meeting or surpassing baseline performance levels for both tasks using either entirely synthetic data or within an augmented context. Nevertheless, the advantages of employing synthetic data in downstream tasks continue to pose a challenge, not

only within the medical image domain but also across various other domains [222, 122, 210], thus requiring more comprehensive assessment and thorough examination.

5.5.4 Considerations on memorization

In medicine, adherence to privacy regulations is a sensitive requirement. While it is generally not possible for domain experts to infer patient identities from the image content of a histological tile or slide alone [82], developers and users of generative models are well advised to understand the risk of correspondence between the training data and the synthesized data. To this end, we evaluate the training and synthesized data against two memorization measures. The authenticity score $A \in [0, 1]$ by [4] aims to measure the rate by which a model generates new samples (higher score means more innovative samples). Similarly, [144] aims to estimate the degree of data copying C_T from the training data by the generative model. A $C_T \ll 0$ implies data copying, while a $C_T \gg 0$ implies an underfitting of the model. The closer to 0 the better. See Section 5.5.1 for a precise closed form of the measures and Table 5.3 for the full quantitative results, indicating that the DiffInfinite model is not prone to data copying across all resolutions and variations considered here [8]. The A range between 0.86 and 0.98, signifying a high rate of authenticity. While other papers unfortunately do not report such detailed memorization statistics for their models, the results by [4] suggest that a score $\gg 0.8$ is not trivial to achieve. None of the models under consideration in [4] (VAE, DCGAN, WGAN-GP, ADS-GAN) achieve more than 0.82 in A on simpler data (MNIST). This interpretation is strengthened by the results of a $C_T \gg 0$ which indicates that the model might even be underfitting and is not in a data copying regime. Qualitative results on the nearest neighbour search between training and synthetic data in Figure 5.1 further corroborate these quantitative results.

5.6 Discussion

DiffInfinite offers a novel sampling method to generate large images in digital pathology. Due to the high-level mask generation followed by the low-level image generation, synthetic images contain long-range correlations while maintaining high-quality details. Since the model trains and samples on small patches, it can be efficiently parallelized. We demonstrated that the classifier-free guidance can be extended to a semi-supervised learning method, expanding the labelled data feature space with unlabelled data. The biological plausibility of the synthetic images was assessed in a survey by 10 domain experts. Despite their training, most participants found it challenging to differentiate between real and synthetic data, reporting an average low confidence in

[8] We use `https://github.com/marcojira/fls` from [93] to calculate both scores.

their decisions. We found that samples from DiffInfinite can help in certain downstream machine learning tasks, on both in- as well as out-of-distribution datasets. Finally, authenticity metrics validate DiffInfinite's capacity to generate novel data points with little similarity to the training data which is beneficial for the privacy-preserving use of generative models in medicine.

Chapter 6.

Conclusions

Scientific Machine Learning (SciML) represents a novel methodology to address challenges in science and engineering, aiming to develop advanced machine learning techniques that operate effectively with incomplete information about the system. The strength of SciML lies in its ability to leverage domain-specific knowledge, which often includes well-established physical laws and principles, to guide and enhance the learning process. This approach not only improves the accuracy of predictions in data-sparse environments but also enhances the interpretability of machine learning models. By doing so, SciML offers a pathway to develop more reliable and trustworthy neural networks in high-stakes domains.

In this thesis, we explored how to integrate physical knowledge in the context of imaging applications. The focus is on developing methods that bridge the gap between physical insights in data acquisition and machine learning paradigms, aiming to improve performance on out-of-distribution data and validate the model's robustness in real-world scenarios. We combined neural networks with well-defined, physics-based models to create hybrid systems, which are more transparent, controllable, and reliable in given scenarios.

For decades, image data processing has been developed to resemble human perception of reality. The image signal process (ISP) is composed of several transformations which irreversibly modify the content of the original information collected by the sensor. Considering all the different algorithms and parameter configurations for each data processing, we have an exponential number of possible output images. While visually they have slight variations, from a machine learning perspective, these perturbations introduce out-of-distribution data, leading to a lack in performance on deployed models. We introduced and defined this problem as *dataset drift*. In Chapter 3, starting from raw data, we combined the image data processing with the downstream model, obtaining a physically faithful framework for model validation and optimization. We showcased three applications of our proposed hybrid model, formulated as *drift synthesis*, *drift forensic* and *drift optimization*.

In the *drift synthesis* experiments, we generate test cases that mimic physical device variability, like camera differences across labs, without collecting new data. The approach offers physically faithful robustness to perturbations compared to standard data augmentation. Our framework validates the model's resilience to perturbations introduced by the processing transformations applied to raw data, in contrast to data augmentation, which superimposes alterations on an already processed image. Through *drift forensics*, we can identify which elements, in the processing pipeline, have the greatest impact on model performance post-deployment. Given the data forward model, we leveraged adversarial perturbations to detect specific transformations or parameter sets that could lead to model failures. Lastly, *drift optimization* connects the processing model with the downstream model. The gradient backpropagates from the output, passing through the hybrid model, all the way to the raw data. This experiment fine-tunes the data processing to suit the specific task at hand, demonstrating that the ISP designed for human visual perception does not align with the optimal configuration for a neural network.

While ISP transformations do address a wide range of possible perturbations on the output data, achieving a truly comprehensive understanding and control over every possible perturbation necessitates extending our analysis all the way back to the initial light source. In Chapter 4, we extended beyond the conventional ISP framework, designing a data emulation using prior knowledge of the source and target acquisition systems. Starting from raw drone data, we emulated the optical compound, the sensor's noise distribution and the dynamics of a satellite imaging system. Given the emulated data, we established tolerances for the downstream model in-silico, thereby assessing the model's performance prior to any physical prototyping. Utilizing this protocol, we can effectively apply our approach to satellites that are already in orbit. By projecting drone data into the satellite parameters' hyperspace, we can conduct an investigation into the most compatible model architecture for these specific conditions. This method can be extended to medical imaging applications, defining a high-resolution imaging setup as the source, making different hospital acquisition setups resilient to the same machine learning model.

Emulating optical compounds and ISPs enhances model robustness against perturbations, yet it falls short in addressing out-of-distribution challenges posed by images containing novel spatial features. Following a comprehensive examination of every element in the data acquisition process for imaging applications, it's crucial to shift our focus towards generating high-fidelity data that not only exhibits physical plausibility but also maintains this integrity across large scales. In Chapter 5, we introduced DiffInfinite, a method which combines high-level mask generation with subsequent low-level image generation, allowing the synthetic images to encapsulate long-range correlations while preserving high-quality details. A key aspect of this framework involves assessing the biological plausibility of the generated synthetic images. This assessment was carried out through a survey involving domain experts, where most found it challenging to distinguish between real and synthetic data, indicating the high fidelity of the generated images. Moreover,

the authenticity metrics employed in the study confirm the ability of DiffInfinite to generate novel data points that bear little resemblance to the training data. This feature is particularly crucial for the privacy-preserving use of generative models in medicine, as it minimizes the risk of sensitive data exposure.

In conclusion, this thesis explored SciML approaches in imaging applications, focusing on medical and aerospace imagery challenges. The exploration conducted reveals that there is no universally applicable solution. The success of integrating physical knowledge depends on the quality of the information, the complexity of the tasks, and the flexibility of the machine learning architecture used. This work underscores the potential for machine learning to evolve in conjunction with domain-specific knowledge, leading to mutual advancements in both areas, essential for progress in artificial intelligence.

Appendix A.

Processing-based Data Model

A.1 Data models details

The following values were used to initialize $\mathbf{\Phi}_{\text{Proc}}^{\text{para}}$ (both "Frozen" and "Learned") in experiment 3.3.3:

```
class ParametrizedProcessing(nn.Module):
    """Differentiable processing pipeline via torch transformations

    Args:
        camera_parameters (tuple(list), optional): applies given camera parameters in processing
        track_stages (bool, optional): whether or not to retain intermediary steps in processing
        batch_norm_output (bool, optional): adds a BatchNorm layer to the end of the processing
    """

    def __init__(self, camera_parameters=None, track_stages=False, batch_norm_output=True):
        super().__init__()
        self.stages = None
        self.buffer = None
        self.track_stages = track_stages

        if camera_parameters is None:
            camera_parameters = DEFAULT_CAMERA_PARAMS

        black_level, white_balance, colour_matrix = camera_parameters

        self.black_level = nn.Parameter(torch.as_tensor(black_level))
        self.white_balance = nn.Parameter(torch.as_tensor(white_balance).reshape(1, 3))
        self.colour_correction = nn.Parameter(torch.as_tensor(colour_matrix).reshape(3, 3))

        self.gamma_correct = nn.Parameter(torch.Tensor([2.2]))

        self.debayer = Debayer()

        self.sharpening_filter = nn.Conv2d(1, 1, kernel_size=3, padding=1, bias=False)
        self.sharpening_filter.weight.data[0][0] = K_SHARP.clone()

        self.gaussian_blur = nn.Conv2d(1, 1, kernel_size=5, padding=2, padding_mode='reflect',
        self.gaussian_blur.weight.data[0][0] = K_BLUR.clone()

        self.batch_norm = nn.BatchNorm2d(3, affine=False) if batch_norm_output else None

        self.register_buffer('M_RGB_2_YUV', M_RGB_2_YUV.clone())
        self.register_buffer('M_YUV_2_RGB', M_YUV_2_RGB.clone())

        self.additive_layer = None
```

where

```
K_G = torch.Tensor([[0, 1, 0],
                    [1, 4, 1],
                    [0, 1, 0]]) / 4

K_RB = torch.Tensor([[1, 2, 1],
                     [2, 4, 2],
                     [1, 2, 1]]) / 4

M_RGB_2_YUV = torch.Tensor([[0.299, 0.587, 0.114],
                            [-0.14714119, -0.28886916, 0.43601035],
                            [0.61497538, -0.51496512, -0.10001026]])
M_YUV_2_RGB = torch.Tensor([[1.0000000000e+00, -4.1827794561e-09, 1.1398830414e+00],
                            [1.0000000000e+00, -3.9464232326e-01, -5.8062183857e-01],
                            [1.0000000000e+00, 2.0320618153e+00, -1.2232658220e-09]])

K_BLUR = torch.Tensor([[6.9625e-08, 2.8089e-05, 2.0755e-04, 2.8089e-05, 6.9625e-08],
                       [2.8089e-05, 1.1332e-02, 8.3731e-02, 1.1332e-02, 2.8089e-05],
                       [2.0755e-04, 8.3731e-02, 6.1869e-01, 8.3731e-02, 2.0755e-04],
                       [2.8089e-05, 1.1332e-02, 8.3731e-02, 1.1332e-02, 2.8089e-05],
                       [6.9625e-08, 2.8089e-05, 2.0755e-04, 2.8089e-05, 6.9625e-08]])
K_SHARP = torch.Tensor([[0, -1, 0],
                        [-1, 5, -1],
                        [0, -1, 0]])
DEFAULT_CAMERA_PARAMS = (
    [0., 0., 0., 0.],
    [1., 1., 1.],
    [1., 0., 0., 0., 1., 0., 0., 0., 1.],
)
```

Note that the camera parameters are camera, and conversely in our case dataset, dependent and defined in the dataset classes.

A.2 Additional Results

A.2.1 Drift synthesis

	Microscopy-ISP		Microscopy-CC		Drone-ISP		Drone-CC	
Rank	Train pipeline	Avg. score	Train pipeline	Avg. score	Train pipeline	Avg. score	Train pipeline	Avg. score
1	ma,s,me	0.83	bi,u,me	0.63	ma,u,ga	0.68	ma,s,ga	0.60
2	ma,u,me	0.83	me,s,me	0.63	bi,s,ga	0.68	bi,s,ga	0.57
3	ma,u,ga	0.82	bi,u,ga	0.62	bi,s,me	0.67	me,s,ga	0.57
4	bi,s,me	0.81	ma,s,me	0.62	ma,s,me	0.67	ma,s,me	0.55
5	bi,u,me	0.81	me,u,me	0.62	me,u,ga	0.67	me,s,me	0.55
6	me,s,me	0.81	ma,s,ga	0.62	me,u,me	0.67	ma,u,ga	0.55
7	bi,s,ga	0.81	ma,u,me	0.61	ma,u,me	0.66	bi,s,me	0.54
8	me,s,ga	0.80	me,s,ga	0.60	ma,s,ga	0.66	ma,u,me	0.54
9	me,u,me	0.80	bi,s,me	0.59	bi,u,me	0.65	me,u,me	0.53
10	ma,s,ga	0.80	ma,u,ga	0.59	me,s,me	0.65	me,u,ga	0.51
11	bi,u,ga	0.79	bi,s,ga	0.58	me,s,ga	0.64	bi,u,me	0.48
12	me,u,ga	0.79	me,u,ga	0.58	bi,u,ga	0.61	bi,u,ga	0.46

Table A.1: Rankings of task models from Section 3.3.1 trained on different data models (columns 2, 4, 6, 8) according to their average accuracy or IoU (columns 3, 5, 7, 9) across all test pipelines respective corruptions. ISP corresponds to drift synthesis with physically faithful data models, CC corresponds to common corruptions.

	Microscopy-ISP											
Rank	bi,s,me	bi,s,ga	bi,u,me	bi,u,ga	ma,s,me	ma,s,ga	ma,u,me	ma,u,ga	me,s,me	me,s,ga	me,u,me	me,u,ga
1	ma,u,me	ma,u,me	ma,u,ga	ma,u,ga	ma,s,me	ma,u,ga	ma,u,ga	ma,u,ga	ma,u,me	me,s,ga	ma,u,ga	ma,u,ga
2	ma,u,ga	ma,u,ga	bi,s,ga	bi,s,ga	bi,s,me	me,s,ga	ma,s,me	ma,u,me	ma,s,me	ma,u,ga	ma,u,me	ma,u,me
3	bi,s,ga	bi,s,ga	ma,s,me	ma,s,me	bi,u,ga	ma,s,ga	ma,u,me	ma,s,me	bi,s,ga	ma,s,ga	ma,s,me	ma,s,me
4	ma,s,me	ma,s,me	ma,u,me	ma,u,me	ma,u,me	ma,s,me	bi,s,ga	me,u,me	me,s,ga	me,u,ga	me,u,me	me,u,me
5	bi,s,me	bi,u,me	me,u,me	me,u,me	bi,u,me	ma,u,me	me,u,me	ma,s,ga	bi,u,me	me,s,me	bi,s,ga	bi,s,ga
6	bi,u,me	me,u,me	bi,u,me	bi,u,me	ma,u,ga	me,s,me	me,s,ga	bi,s,ga	ma,u,ga	ma,u,me	me,u,ga	me,u,ga
7	me,s,me	bi,s,me	bi,s,me	me,s,me	me,s,me	me,u,me	me,s,me	me,s,ga	me,u,me	ma,s,me	me,s,me	me,s,me
8	me,s,ga	me,s,me	me,s,me	bi,u,ga	bi,s,ga	bi,u,me	ma,s,ga	me,s,me	me,s,me	me,u,me	bi,s,me	bi,s,me
9	me,u,me	me,s,ga	bi,u,ga	bi,s,me	me,s,ga	me,u,ga	bi,u,me	bi,u,me	bi,s,me	bi,s,me	me,s,ga	me,s,ga
10	ma,s,ga	ma,s,ga	ma,s,ga	ma,s,ga	ma,s,ga	bi,s,me	bi,s,me	bi,s,me	ma,s,ga	bi,s,ga	ma,s,ga	ma,s,ga
11	bi,u,ga	me,u,ga	me,u,ga	me,s,ga	me,u,ga	bi,s,ga	me,u,ga	me,u,ga	me,u,ga	bi,u,me	bi,u,me	bi,u,me
12	me,u,ga	bi,u,ga	me,s,ga	me,u,ga	me,u,me	bi,u,ga	bi,u,ga	bi,u,ga	bi,u,ga	bi,u,ga	bi,u,ga	bi,u,ga

Table A.2: Ranking of task models from Section 3.3.1 trained under different train pipelines (rows) for each individual test pipeline (columns 2 - 13).

	Microscopy-CC										
Rank	identity	gauss noise	shot	impulse	speckle	gauss blur	zoom	contrast	brightness	saturate	elastic
1	ma,u,me	ma,u,me	bi,u,me	bi,u,me	ma,s,ga	bi,s,ga	bi,s,ga	bi,s,ga	me,s,me	ma,s,me	bi,s,ga
2	ma,u,ga	ma,s,ga	ma,s,ga	me,u,me	bi,u,me	ma,u,me	ma,u,ga	bi,u,ga	ma,s,me	me,u,me	ma,u,ga
3	bi,s,ga	me,u,me	me,s,me	bi,u,ga	me,s,me	ma,u,ga	ma,s,me	me,u,ga	bi,u,ga	me,s,me	ma,u,me
4	me,s,me	me,s,ga	ma,u,me	me,s,me	me,u,me	bi,u,me	ma,u,me	ma,s,me	ma,s,ga	bi,u,ga	ma,s,me
5	ma,s,me	bi,u,me	me,s,ga	ma,s,me	bi,u,ga	me,u,me	bi,u,me	ma,u,me	bi,s,me	bi,s,ga	me,u,me
6	me,u,me	ma,u,ga	me,u,me	ma,u,me	ma,s,me	ma,s,me	me,s,me	bi,s,me	bi,u,me	bi,u,me	me,s,ga
7	me,s,ga	me,s,me	bi,s,me	ma,u,ga	ma,u,me	me,s,ga	bi,u,ga	bi,u,me	me,s,ga	ma,u,ga	me,s,me
8	bi,u,me	bi,s,me	bi,u,ga	me,s,ga	me,s,ga	ma,s,ga	me,u,ga	me,s,me	ma,u,ga	ma,s,ga	bi,u,ga
9	bi,u,ga	ma,s,me	ma,s,me	me,u,ga	bi,s,me	me,s,me	me,u,me	ma,s,ga	me,u,ga	bi,s,me	bi,u,me
10	ma,s,ga	bi,u,ga	ma,u,ga	ma,s,ga	ma,u,ga	bi,u,ga	me,s,ga	ma,u,ga	bi,s,ga	me,s,ga	ma,s,ga
11	bi,s,me	bi,s,ga	bi,s,ga	bi,s,me	me,u,ga	bi,s,me	ma,s,ga	me,u,me	me,u,me	me,u,ga	me,u,ga
12	me,u,ga	me,u,ga	me,u,ga	bi,s,ga	bi,s,ga	me,u,ga	bi,s,me	me,s,ga	ma,u,me	ma,u,me	bi,s,me

Table A.3: Ranking of task models from Section 3.3.1 trained under different train pipelines (rows) for each individual test corruptions (columns 2 - 12).

	Drone-ISP											
Rank	bi,s,me	bi,s,ga	bi,u,me	bi,u,ga	ma,s,me	ma,s,ga	ma,u,me	ma,u,ga	me,s,me	me,s,ga	me,u,me	me,u,ga
1	bi,s,me	bi,s,ga	bi,u,me	bi,u,me	ma,u,ga	ma,s,ga	ma,u,ga	ma,u,ga	ma,s,me	ma,s,ga	ma,u,ga	ma,u,ga
2	bi,u,me	bi,s,me	bi,s,me	bi,s,me	ma,s,me	me,s,ga	me,u,me	me,u,me	ma,s,ga	me,s,ga	me,u,me	me,u,ga
3	ma,u,ga	ma,u,ga	bi,u,ga	bi,u,ga	bi,s,ga	ma,s,me	ma,u,me	me,u,me	ma,u,ga	ma,s,me	ma,s,me	me,u,me
4	bi,s,ga	ma,s,me	ma,u,ga	ma,u,ga	me,u,ga	me,s,me	bi,s,me	bi,s,me	bi,s,ga	me,s,me	me,u,ga	ma,s,me
5	me,u,me	me,u,ga	me,u,me	me,u,me	ma,s,ga	bi,s,ga	ma,s,me	ma,s,me	me,u,ga	bi,s,ga	ma,u,me	ma,u,me
6	bi,u,ga	ma,s,ga	bi,s,ga	bi,s,ga	ma,u,me	ma,u,ga	bi,s,ga	bi,s,ga	me,s,me	ma,u,ga	bi,s,me	bi,s,me
7	ma,s,me	ma,u,me	ma,u,me	ma,u,me	me,u,me	me,u,ga	me,u,ga	me,u,ga	me,s,ga	me,u,ga	bi,u,me	bi,s,ga
8	me,u,ga	me,s,ga	ma,s,me	ma,s,me	me,s,me	me,u,me	bi,u,me	bi,u,me	me,u,me	me,u,me	bi,s,ga	bi,u,me
9	ma,u,me	me,u,me	me,u,ga	me,u,ga	bi,s,me	ma,u,me	bi,u,ga	ma,s,ga	ma,u,me	ma,u,me	me,s,me	ma,s,ga
10	me,s,me	me,s,me	me,s,me	me,s,me	me,s,ga	bi,s,me	ma,s,ga	me,s,me	bi,s,me	bi,s,me	bi,u,ga	me,s,me
11	ma,s,ga	bi,u,me	me,s,ga	ma,s,ga	bi,u,me	bi,u,me	me,s,me	bi,u,ga	bi,u,me	bi,u,me	ma,s,ga	bi,u,ga
12	me,s,ga	bi,u,ga	ma,s,ga	me,s,ga	bi,u,ga	bi,u,ga	me,s,ga	me,s,ga	bi,u,ga	bi,u,ga	me,s,ga	me,s,ga

Table A.4: Ranking of task models from 3.3.1 trained under different train pipelines (rows) for each individual test pipeline (columns 2 - 13).

	Drone-CC										
Rank	identity	gauss noise	shot	impulse	speckle	gauss blur	zoom	contrast	brightness	saturate	elastic
1	ma,s,ga	ma,s,ga	ma,s,ga	ma,s,ga	ma,s,ga	ma,s,ga	bi,s,me	bi,s,ga	bi,s,ga	ma,s,ga	ma,s,ga
2	bi,s,ga	me,s,ga	me,s,ga	me,s,ga	me,s,ga	bi,s,ga	ma,s,ga	ma,s,ga	ma,s,ga	ma,s,me	ma,u,ga
3	me,s,ga	bi,s,ga	bi,s,ga	me,s,me	bi,s,ga	ma,s,me	bi,s,ga	me,s,me	ma,s,me	ma,u,ga	ma,s,me
4	ma,s,me	me,s,me	ma,s,me	bi,s,ga	ma,s,me	ma,u,ga	me,s,ga	ma,s,me	me,s,me	me,u,ga	bi,s,ga
5	ma,u,ga	ma,u,ga	me,s,me	ma,u,ga	me,s,me	bi,u,me	ma,u,me	bi,s,me	ma,u,me	me,s,ga	bi,s,me
6	bi,s,me	ma,u,me	ma,u,ga	ma,u,me	ma,u,ga	bi,s,me	me,s,me	ma,u,me	ma,u,ga	bi,s,ga	bi,u,me
7	me,u,ga	me,u,me	ma,u,me	me,u,me	bi,s,me	me,s,ga	ma,s,me	ma,u,ga	me,u,me	bi,s,me	me,s,ga
8	bi,u,me	ma,s,me	bi,s,me	ma,s,me	ma,u,me	ma,u,me	bi,u,me	me,s,ga	bi,s,me	me,s,me	me,u,me
9	ma,u,me	bi,s,me	me,u,me	bi,s,me	me,u,me	me,u,me	me,u,me	bi,u,me	me,u,ga	me,u,me	me,u,ga
10	me,u,me	me,u,ga	me,u,ga	me,u,ga	me,u,ga	me,s,me	bi,u,ga	bi,u,ga	me,s,ga	bi,u,me	me,s,me
11	me,s,me	bi,u,me	bi,u,me	bi,u,me	bi,u,me	me,u,ga	ma,u,ga	me,u,ga	bi,u,me	ma,u,me	ma,u,me
12	bi,u,ga	bi,u,ga	bi,u,ga	bi,u,ga	bi,u,ga	bi,u,ga	me,u,ga	me,u,me	bi,u,ga	bi,u,ga	bi,u,ga

Table A.5: Ranking of task models from Section 3.3.1 trained under different train pipelines (rows) for each individual test corruptions (columns 2 - 12).

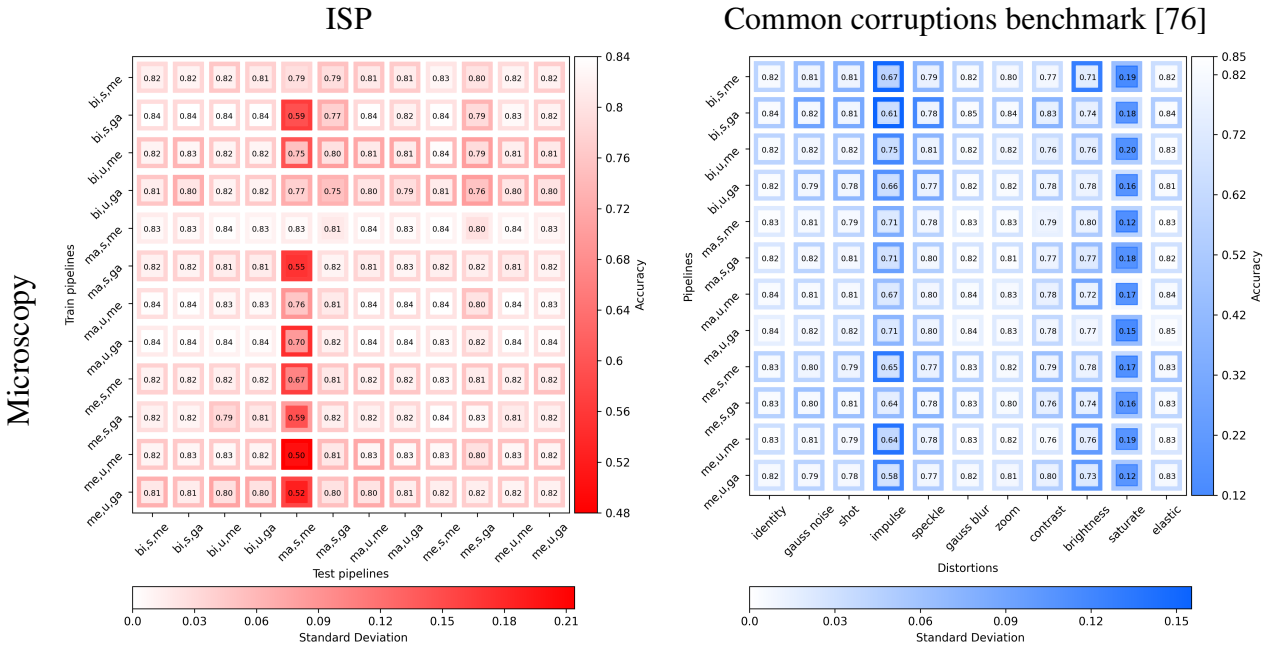

Figure A.1: Experiment from Section 3.3.1 with weak severity (level 1) for the Common corruptions benchmark.

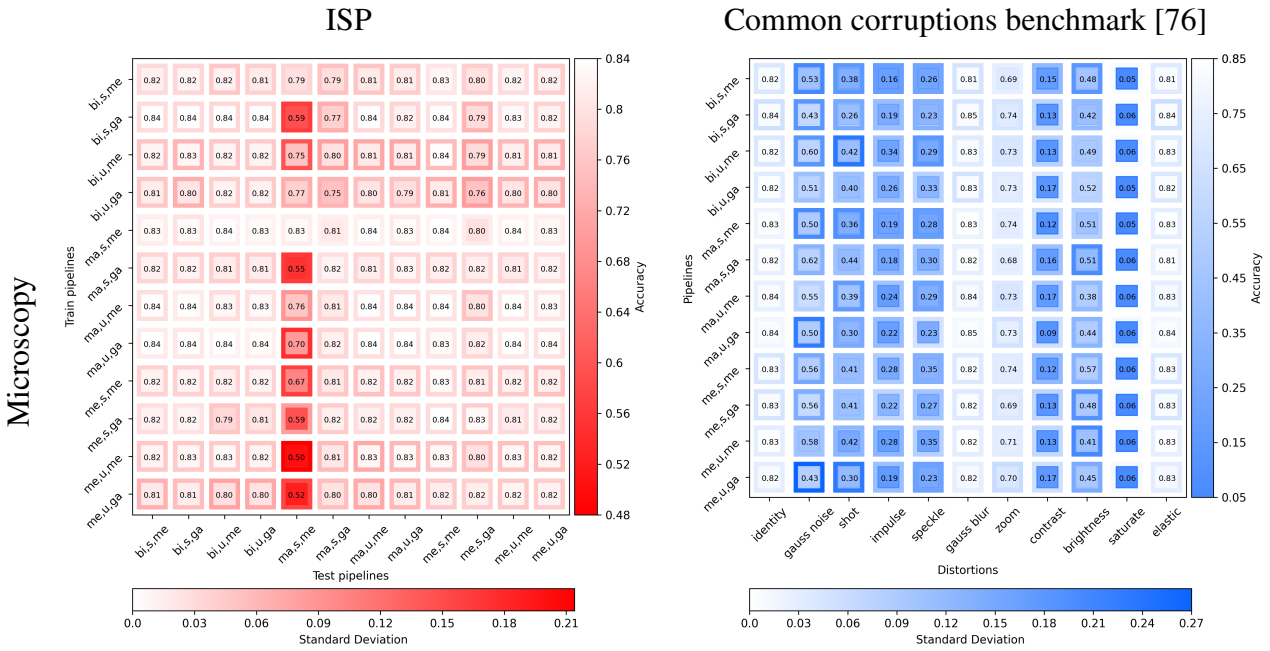

Figure A.2: Experiment from Section 3.3.1 with strong severity (level 5) for the Common corruptions benchmark.

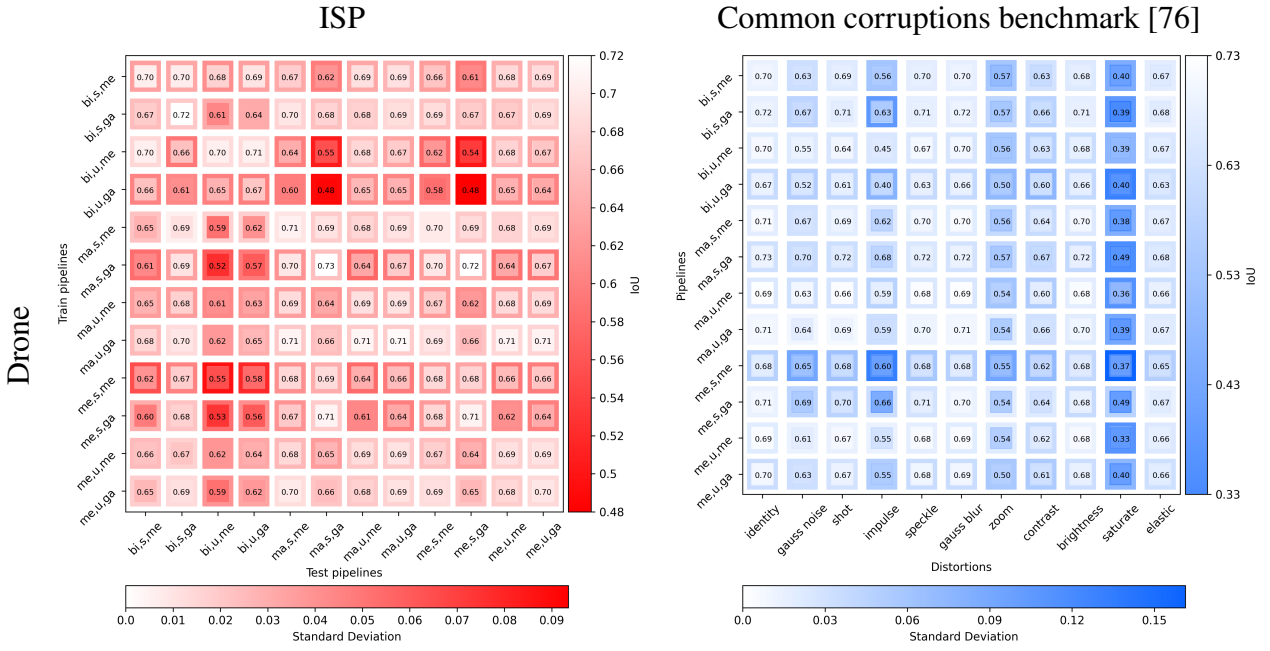

Figure A.3: Experiment from Section 3.3.1 with weak severity (level 1) for the Common corruptions benchmark.

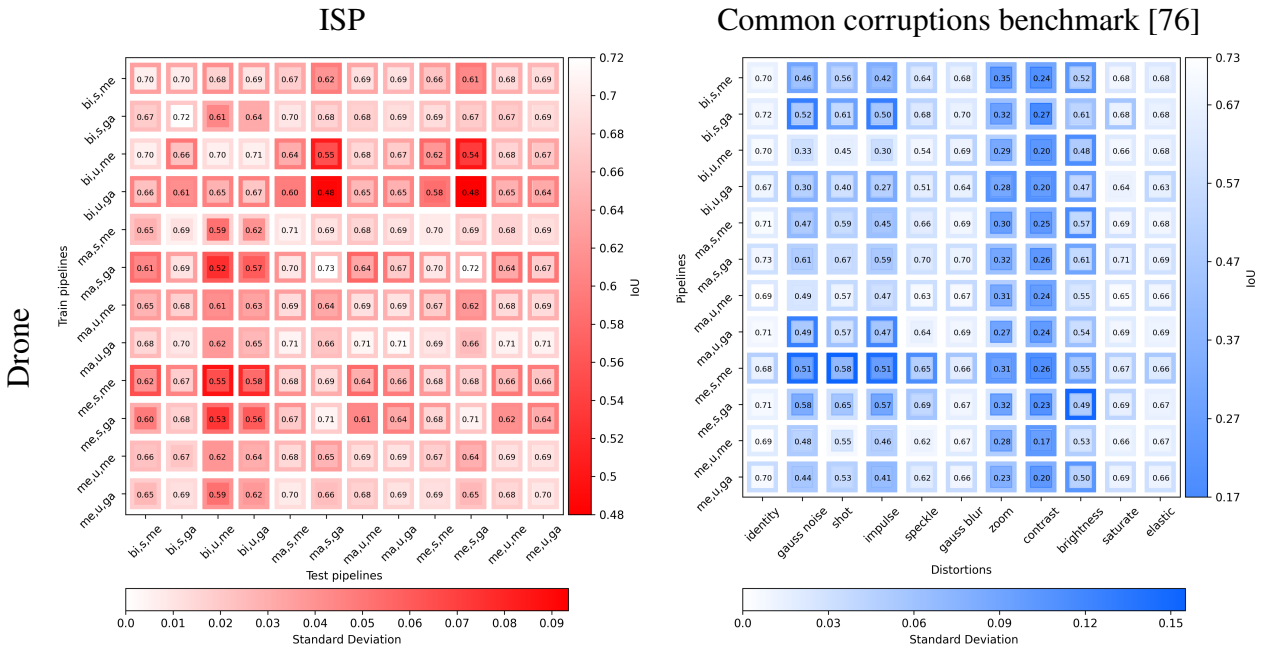

Figure A.4: Experiment from Section 3.3.1 with strong severity (level 5) for the Common corruptions benchmark.

Bibliography

[1] Midjourney. `https://www.midjourney.com`.

[2] Abdelrahman Abdelhamed, Stephen Lin, and Michael S. Brown. A high-quality denoising dataset for smartphone cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[3] Ravi Aggarwal, Viknesh Sounderajah, Guy Martin, Daniel SW Ting, Alan Karthikesalingam, Dominic King, Hutan Ashrafian, and Ara Darzi. Diagnostic accuracy of deep learning in medical imaging: a systematic review and meta-analysis. *NPJ digital medicine*, 4(1):65, 2021.

[4] Ahmed Alaa, Boris Van Breugel, Evgeny S Saveliev, and Mihaela van der Schaar. How faithful is your synthetic data? sample-level metrics for evaluating and auditing generative models. In *International Conference on Machine Learning*, pages 290–306. PMLR, 2022.

[5] B Albertina, M Watson, C Holback, R Jarosz, S Kirk, Y Lee, and J Lemmerman. Radiology data from the cancer genome atlas lung adenocarcinoma [tcga-luad] collection. *The Cancer Imaging Archive*, 2016.

[6] Anastasios N Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2021.

[7] Guilherme Aresta, Teresa Araújo, Scotty Kwok, Sai Saketh Chennamsetty, Mohammed Safwan, Varghese Alex, Bahram Marami, Marcel Prastawa, Monica Chan, Michael Donovan, Gerardo Fernandez, Jack Zeineh, Matthias Kohl, Christoph Walz, Florian Ludwig, Stefan Braunewell, Maximilian Baust, Quoc Dang Vu, Minh Nguyen Nhat To, Eal Kim, Jin Tae Kwak, Sameh Galal, Veronica Sanchez-Freire, Nadia Brancati, Maria Frucci, Daniel Riccio, Yaqi Wang, Lingling Sun, Kaiqiang Ma, Jiannan Fang, Ismael Kone, Lahsen Boulmane, Aurélio Campilho, Catarina Eloy, António Polónia, and Paulo Aguiar. Bach: Grand challenge on breast cancer histology images. *Medical Image Analysis*, 56:

122–139, 2019. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2019.05.010. URL https://www.sciencedirect.com/science/article/pii/S1361841518307941.

[8] Vinay Ayyappan, Alex Chang, Chi Zhang, Santosh Kumar Paidi, Rosalie Bordett, Tiffany Liang, Ishan Barman, and Rishikesh Pandey. Identification and staging of b-cell acute lymphoblastic leukemia using quantitative phase imaging and machine learning. *ACS Sensors*, 5(10):3281–3289, 2020. doi: 10.1021/acssensors.0c01811. URL https://doi.org/10.1021/acssensors.0c01811. PMID: 33092347.

[9] Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *Journal of Machine Learning Research*, 20:1–25, 2019.

[10] B. Bain. Diagnosis from the blood smear. *The New England journal of medicine*, 353 5: 498–507, 2005.

[11] Shane Barratt and Rishi Sharma. A note on the inception score. *arXiv preprint arXiv:1801.01973*, 2018.

[12] Bryce E Bayer. Color imaging array, July 20 1976. US Patent 3,971,065.

[13] Emma Beede, Elizabeth Baylor, Fred Hersch, Anna Iurchenko, Lauren Wilcox, Paisan Ruamviboonsuk, and Laura M. Vardoulakis. A human-centered evaluation of a deep learning system deployed in clinics for the detection of diabetic retinopathy. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, page 1–12, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450367080. URL https://doi.org/10.1145/3313831.3376718.

[14] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 456–473, 2018.

[15] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

[16] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424, 2000.

[17] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5050–5060, 2019.

[18] Julian Bitterwolf, Alexander Meinke, and Matthias Hein. Certifiably adversarially robust detection of out-of-distribution data. *Advances in Neural Information Processing Systems*, 33:16085–16095, 2020.

[19] Hamidreza Bolhasani, Elham Amjadi, Maryam Tabatabaeian, and Somayyeh Jafarali Jassbi. A histopathological image dataset for grading breast invasive ductal carcinomas. *Informatics in Medicine Unlocked*, 19:100341, 2020. ISSN 2352-9148. doi: https://doi.org/10.1016/j.imu.2020.100341. URL https://www.sciencedirect.com/science/article/pii/S2352914820300757.

[20] Sam Bond-Taylor and Chris G. Willcocks. ∞-diff: Infinite resolution diffusion with subsampled mollified states, 2023.

[21] Daniel L. Bongiorno, Mitch Bryson, Donald G. Dansereau, and Stefan B. Williams. Spectral characterization of COTS RGB cameras using a linear variable edge filter. page 86600N, Burlingame, California, USA, January 2013. doi: 10.1117/12.2001460. URL http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.2001460.

[22] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T Barron. Unprocessing images for learned raw denoising. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[23] Tatiana A Bubba, Gitta Kutyniok, Matti Lassas, Maximilian März, Wojciech Samek, Samuli Siltanen, and Vignesh Srinivasan. Learning the invisible: a hybrid deep learning-shearlet framework for limited angle computed tomography. *Inverse Problems*, 35(6): 064002, 2019.

[24] Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. *arXiv preprint arXiv:2301.13188*, 2023.

[25] Pierre Chambon, Christian Bluethgen, Curtis P Langlotz, and Akshay Chaudhari. Adapting pretrained vision-language foundational models to medical imaging domains. *arXiv preprint arXiv:2210.04133*, 2022.

[26] Pierre J. Chambon, Christian Bluethgen, Jean-Benoit Delbrouck, Rogier van der Sluijs, Malgorzata Polacin, Juan Manuel Zambrano Chaves, Tanishq Mathew Abraham, Shivanshu Purohit, Curtis P. Langlotz, and Akshay Chaudhari. Roentgen: Vision-language foundation model for chest x-ray generation. *CoRR*, abs/2211.12737, 2022. URL https://doi.org/10.48550/arXiv.2211.12737.

[27] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3291–3300, 2018.

[28] Mayee Chen, Karan Goel, Nimit S Sohoni, Fait Poms, Kayvon Fatahalian, and Christopher Ré. Mandoline: Model evaluation under distribution shift. In *International Conference on Machine Learning*, pages 1617–1629. PMLR, 2021.

[29] Richard Chen, Ming Lu, Tiffany Chen, Drew Williamson, and Faisal Mahmood. Synthetic data in machine learning for medicine and healthcare. *Nature Biomedical Engineering*, 5: 1–5, 06 2021.

[30] Richard J Chen, Chengkuan Chen, Yicong Li, Tiffany Y Chen, Andrew D Trister, Rahul G Krishnan, and Faisal Mahmood. Scaling vision transformers to gigapixel images via hierarchical self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16144–16155, 2022.

[31] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[32] Phillip Chlap, Min Hang, Nym Vandenberg, Jason Dowling, Lois Holloway, and Annette Haworth. A review of medical image data augmentation techniques for deep learning applications. *Journal of Medical Imaging and Radiation Oncology*, 65, 06 2021. doi: 10.1111/1754-9485.13261.

[33] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pages 1310–1320. PMLR, 2019.

[34] Joseph Paul Cohen, Margaux Luck, and Sina Honari. Distribution matching losses can hallucinate features in medical image translation. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, pages 529–536. Springer International Publishing, 09 2018. ISBN 978-3-030-00928-1. doi: 10.1007/978-3-030-00928-1_60.

[35] Ryan Conrad and Kedar Narayan. Cem500k, a large-scale heterogeneous unlabeled cellular electron microscopy image dataset for deep learning. *eLife*, 10:e65894, apr 2021. ISSN 2050-084X. doi: 10.7554/eLife.65894. URL `https://doi.org/10.7554/eLife.65894`.

[36] Patrick Cousot. Abstract interpretation. *ACM Computing Surveys (CSUR)*, 28(2):324–328, 1996.

[37] Miles Cranmer, Sam Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. Lagrangian neural networks. *arXiv preprint arXiv:2003.04630*, 2020.

[38] Duc-Tien Dang-Nguyen, Cecilia Pasquini, Valentina Conotter, and Giulia Boato. Raise: A raw images dataset for digital image forensics. In *Proceedings of the 6th ACM Multimedia Systems Conference*, MMSys '15, page 219–224, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450333511. doi: 10.1145/2713168.2713194. URL https://doi.org/10.1145/2713168.2713194.

[39] Dipankar Dasgupta. *Artificial immune systems and their applications*. Springer Science & Business Media, 2012.

[40] Sumanth Dathathri, Krishnamurthy Dvijotham, Alexey Kurakin, Aditi Raghunathan, Jonathan Uesato, Rudy R Bunel, Shreya Shankar, Jacob Steinhardt, Ian Goodfellow, Percy S Liang, et al. Enabling certification of verification-agnostic networks via memory-efficient semidefinite programming. *Advances in Neural Information Processing Systems*, 33: 5318–5331, 2020.

[41] Mathieu Dejean-Servières, Karol Desnos, Kamel Abdelouahab, Wassim Hamidouche, Luce Morin, and Maxime Pelcat. Study of the impact of standard image compression techniques on performance of image classification with a convolutional neural network. Research Report hal-01725126, INSA Rennes; Univ Rennes; IETR; Institut Pascal, 2017.

[42] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[43] Samuel Dodge and Lina Karam. A study and comparison of human and deep learning recognition performance under visual distortions. In *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–7, 2017. doi: 10.1109/ICCCN.2017.8038465.

[44] James M Dolezal, Rachelle Wolk, Hanna M Hieromnimon, Frederick M Howard, Andrew Srisuwananukorn, Dmitry Karpeyev, Siddhi Ramesh, Sara Kochanny, Jung Woo Kwon, Meghana Agni, et al. Deep learning generates synthetic cancer histology for explainability and education. *NPJ Precision Oncology*, 7(1):49, 2023.

[45] Xuefeng Du, Zhaoning Wang, Mu Cai, and Yixuan Li. Vos: Learning what you don't know by virtual outlier synthesis. In *International Conference on Learning Representations*, 2021.

[46] Alexei A Efros and Thomas K Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1033–1038. IEEE, 1999.

[47] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *nature*, 542(7639):115–118, 2017.

[48] Virginia Fernandez, Walter Hugo Lopez Pinaya, Pedro Borges, Petru-Daniel Tudosiu, Mark S Graham, Tom Vercauteren, and M Jorge Cardoso. Can segmentation models be trained with fully synthetically generated data? In *Simulation and Synthesis in Medical Imaging: 7th International Workshop, SASHIMI 2022, Held in Conjunction with MICCAI 2022, Singapore, September 18, 2022, Proceedings*, pages 79–90. Springer, 2022.

[49] A Forsey and S Gungor. Demosaicing images from colour cameras for digital image correlation. *Optics and lasers in engineering*, 86:20–28, 2016.

[50] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

[51] Thomas J Fuchs and Joachim M Buhmann. Computational pathology: challenges and promises for tissue analysis. *Computerized Medical Imaging and Graphics*, 35(7-8): 515–530, 2011.

[52] KM Fuhad, Jannat Ferdousey Tuba, Md Sarker, Rabiul Ali, Sifat Momen, Nabeel Mohammed, and Tanzilur Rahman. Deep learning based automatic malaria parasite detection from blood smear and its smartphone based application. *Diagnostics*, 10(5):329, 2020.

[53] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR. URL `https://proceedings.mlr.press/v48/gal16.html`.

[54] MD Garris and RA Wilkinson. Handwritten segmented characters database. technical report special database 3. Technical report, National Institute of Standards and Technology, 1992.

[55] Michael Garris. Design, collection, and analysis of handwriting sample image databases. (31), 1994-08-10 1994. URL `https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=906483`.

[56] Jochen Gast and Stefan Roth. Lightweight probabilistic deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3369–3378, 2018.

[57] Timon Gehr, Matthew Mirman, Dana Drachsler-Cohen, Petar Tsankov, Swarat Chaudhuri, and Martin Vechev. Ai2: Safety and robustness certification of neural networks with abstract interpretation. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2018.

[58] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

[59] Martin Genzel, Jan Macdonald, and Maximilian Marz. Solving inverse problems with deep neural networks - robustness included. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2022. doi: 10.1109/TPAMI.2022.3148324.

[60] Michaël Gharbi, Gaurav Chaurasia, Sylvain Paris, and Frédo Durand. Deep joint demosaicking and denoising. *ACM Transactions on Graphics (ToG)*, 35(6):1–12, 2016.

[61] Judy Wawira Gichoya, Imon Banerjee, Ananth Reddy Bhimireddy, John L Burns, Leo Anthony Celi, Li-Ching Chen, Ramon Correa, Natalie Dullerud, Marzyeh Ghassemi, Shih-Cheng Huang, Po-Chih Kuo, Matthew P Lungren, Lyle J Palmer, Brandon J Price, Saptarshi Purkayastha, Ayis T Pyrros, Lauren Oakden-Rayner, Chima Okechukwu, Laleh Seyyed-Kalantari, Hari Trivedi, Ryan Wang, Zachary Zaiman, and Haoran Zhang. Ai recognition of patient race in medical imaging: a modelling study. *The Lancet Digital Health*, 2022. ISSN 2589-7500. doi: https://doi.org/10.1016/S2589-7500(22)00063-2. URL `https://www.sciencedirect.com/science/article/pii/S2589750022000632`.

[62] Karan Goel, Albert Gu, Yixuan Li, and Christopher Re. Model patching: Closing the subgroup performance gap with data augmentation. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=9YlaeLfuhJF`.

[63] Shafi Goldwasser, Adam Tauman Kalai, Yael Kalai, and Omar Montasser. Beyond perturbations: Learning guarantees with arbitrary adversarial test examples. *Advances in Neural Information Processing Systems*, 33:15859–15870, 2020.

[64] Shafi Goldwasser, Guy N Rothblum, Jonathan Shafer, and Amir Yehudayoff. Interactive proofs for verifying machine learning. In *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

[65] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[66] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[67] Sven Gowal, Krishnamurthy Dj Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. Scalable verified training for provably robust image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4842–4851, 2019.

[68] Bhawna Goyal, Ayush Dogra, Sunil Agrawal, BS Sohi, and Apoorav Sharma. Image denoising review: From classical to state-of-the-art approaches. *Information Fusion*, 55: 220–244, 2020. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2019.09.003. URL https://www.sciencedirect.com/science/article/pii/S1566253519301861.

[69] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020.

[70] IMDRF SaMD Working Group et al. Software as a medical device (samd): Application of quality management system, 2018.

[71] Miriam Hägele, Philipp Seegerer, Sebastian Lapuschkin, Michael Bockmayr, Wojciech Samek, Frederick Klauschen, Klaus-Robert Müller, and Alexander Binder. Resolving challenges in deep learning-based analyses of histopathological images using explanation methods. *Scientific Reports*, 10(1):1–12, 2020.

[72] HAMAMATSU. *ORCA-Flash4.0 V3 Digital CMOS camera C13440-20CU - Technical note*. HAMAMATSU. URL https://www.hamamatsu.com/eu/en/product/cameras/cmos-cameras/C13440-20CU.html#element-id-95e67d91-3547-3319-a6c7-fd29c03e0089.

[73] Samuel W. Hasinoff, Dillon Sharlet, Ryan Geiss, Andrew Adams, Jonathan T. Barron, Florian Kainz, Jiawen Chen, and Marc Levoy. Burst photography for high dynamic range and low-light imaging on mobile cameras. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 35(6), 2016.

[74] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

[75] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

[76] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

[77] Dan Hendrycks, Norman Mu, Ekin D. Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty, 2020.

[78] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[79] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[80] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020.

[81] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *The Journal of Machine Learning Research*, 23(1):2249–2281, 2022.

[82] Petr Holub, Heimo Müller, Tomáš Bíl, Luca Pireddu, Markus Plass, Fabian Prasser, Irene Schlünder, Kurt Zatloukal, Rudolf Nenutil, and Tomáš Brázdil. Privacy risks of whole-slide image sharing in digital pathology. *Nature Communications*, 14(1):2577, 2023.

[83] Le Hou, Ayush Agarwal, Dimitris Samaras, Tahsin M Kurc, Rajarsi R Gupta, and Joel H Saltz. Unsupervised histopathology image synthesis. *arXiv preprint arXiv:1712.05021*, 2017.

[84] Frederick M Howard, James Dolezal, Sara Kochanny, Jefree Schulte, Heather Chen, Lara Heij, Dezheng Huo, Rita Nanda, Olufunmilayo I Olopade, Jakob N Kather, et al. The impact of site-specific digital histology signatures on deep learning model accuracy and bias. *Nature communications*, 12(1):1–13, 2021.

[85] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=nZeVKeeFYf9`.

[86] Peter J Huber. Robust statistics. In *International encyclopedia of statistical science*, pages 1248–1251. Springer, 2011.

[87] Robert William Gainer Hunt and Michael R Pointer. *Measuring colour*. John Wiley & Sons, 2011.

[88] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/e2c420d928d4bf8ce0ff2ec19b371514-Paper.pdf.

[89] Matthew Jagielski, Om Thakkar, Florian Tramer, Daphne Ippolito, Katherine Lee, Nicholas Carlini, Eric Wallace, Shuang Song, Abhradeep Thakurta, Nicolas Papernot, et al. Measuring forgetting of memorized training examples. *arXiv preprint arXiv:2207.00099*, 2022.

[90] Ronnachai Jaroensri, Camille Biscarrat, Miika Aittala, and Frédo Durand. Generating training data for denoising real rgb images via camera pipeline simulation. *arXiv*, 1904.08825, 2019.

[91] Jongheon Jeong and Jinwoo Shin. Consistency regularization for certified robustness of smoothed classifiers. *Advances in Neural Information Processing Systems*, 33:10558–10570, 2020.

[92] Xiyue Jia, Yining Cao, David O'Connor, Jin Zhu, Daniel CW Tsang, Bin Zou, and Deyi Hou. Mapping soil pollution by using drone image recognition and machine learning at an arsenic-contaminated agricultural field. *Environmental Pollution*, 270:116281, 2021.

[93] Marco Jiralerspong, Avishek Joey Bose, and Gauthier Gidel. Feature likelihood score: Evaluating generalization of generative models using samples. *arXiv preprint arXiv:2302.04440*, 2023.

[94] Yong-Yeon Jo, Young Sang Choi, Hyun Woo Park, Jae Hyeok Lee, Hyojung Jung, Hyo-Eun Kim, Kyounglan Ko, Chan Wha Lee, Hyo Soung Cha, and Yul Hwangbo. Impact of image compression on deep learning-based mammogram classification. *Scientific Reports*, 11(1): 1–9, 2021.

[95] Andrej Karpathy. Tesla ai day 2021. URL https://youtu.be/j0z4FweCy4M.

[96] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[97] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35:26565–26577, 2022.

[98] Jakob Nikolas Kather, Niels Halama, and Alexander Marx. 100,000 histological images of human colorectal cancer and healthy tissue, April 2018. URL `https://doi.org/10.5281/zenodo.1214456`.

[99] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, 1412.6980, 2015.

[100] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[101] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, October 2023.

[102] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran Haque, Sara M Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. Wilds: A benchmark of in-the-wild distribution shifts. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5637–5664. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/koh21a.html`.

[103] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran Haque, Sara M Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. Wilds: A benchmark of in-the-wild distribution shifts. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5637–5664. PMLR, 2021. URL `https://proceedings.mlr.press/v139/koh21a.html`.

[104] Daisuke Komura and Shumpei Ishikawa. Machine learning methods for histopathological image analysis. *Computational and Structural Biotechnology Journal*, 16:34–42, 2018. ISSN 2001-0370. doi: https://doi.org/10.1016/j.csbj.2018.01.001. URL `https://www.sciencedirect.com/science/article/pii/S2001037017300867`.

[105] Georg Krempl, Vera Hofer, Geoffrey Webb, and Eyke Hüllermeier. Beyond Adaptation: Understanding Distributional Changes (Dagstuhl Seminar 20372). *Dagstuhl Reports*, 10 (4):1–36, 2021. ISSN 2192-5283. doi: 10.4230/DagRep.10.4.1. URL `https://drops.dagstuhl.de/opus/volltexte/2021/13735`.

[106] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[107] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, may 2017. ISSN 0001-0782. doi: 10.1145/3065386. URL `https://doi.org/10.1145/3065386`.

[108] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2void-learning denoising from single noisy images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2129–2137, 2019.

[109] Marek Kulbacki, Jakub Segen, Wojciech Knieć, Ryszard Klempous, Konrad Kluwak, Jan Nikodem, Julita Kulbacka, and Andrea Serester. Survey of drones for agriculture automation from planting to harvest. In *2018 IEEE 22nd International Conference on Intelligent Engineering Systems (INES)*, pages 000353–000358. IEEE, 2018.

[110] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[111] Tuomas Kynkäänniemi, Tero Karras, Miika Aittala, Timo Aila, and Jaakko Lehtinen. The role of imagenet classes in fréchet inception distance. In *Proc. ICLR*, 2023.

[112] Ruggero Donida Labati, Vincenzo Piuri, and Fabio Scotti. All-idb: The acute lymphoblastic leukemia image database for image processing. In *2011 18th IEEE International Conference on Image Processing*, pages 2045–2048, 2011. doi: 10.1109/ICIP.2011.6115881.

[113] Avisek Lahiri, Arnav Kumar Jain, Sanskar Agrawal, Pabitra Mitra, and Prabir Kumar Biswas. Prior guided gan based semantic inpainting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13696–13705, 2020.

[114] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

[115] Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature Communications*, 10(1):1–8, 2019.

[116] Alexander Lavin, David Krakauer, Hector Zenil, Justin Gottschlich, Tim Mattson, Johann Brehmer, Anima Anandkumar, Sanjay Choudry, Kamil Rocki, Atılım Güneş Baydin, et al. Simulation intelligence: Towards a new generation of scientific methods. *arXiv preprint arXiv:2112.03235*, 2021.

[117] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[118] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, pages 7167–7177, 2018.

[119] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2Noise: Learning image restoration without clean data. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2965–2974. PMLR, 10–15 Jul 2018. URL `https://proceedings.mlr.press/v80/lehtinen18a.html`.

[120] Adrian B Levine, Jason Peng, David Farnell, Mitchell Nursey, Yiping Wang, Julia R Naso, Hezhen Ren, Hossein Farahani, Colin Chen, Derek Chiu, Aline Talhouk, Brandon Sheffield, Maziar Riazy, Philip P Ip, Carlos Parra-Herran, Anne Mills, Naveena Singh, Basile Tessier-Cloutier, Taylor Salisbury, Jonathan Lee, Tim Salcudean, Steven JM Jones, David G Huntsman, C Blake Gilks, Stephen Yip, and Ali Bashashati. Synthesis of diagnostic quality cancer pathology images by generative adversarial networks. *The Journal of Pathology*, 252(2):178–188, 2020.

[121] Linyi Li, Xiangyu Qi, Tao Xie, and Bo Li. Sok: Certified robustness for deep neural networks. *arXiv preprint arXiv:2009.04131*, 2020.

[122] Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Luke Zettlemoyer, Omer Levy, Jason Weston, and Mike Lewis. Self-alignment with instruction backtranslation. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=1oijHJBRsT`.

[123] Xin Li, Bahadir Gunturk, and Lei Zhang. Image demosaicing: A systematic survey. In *Visual Communications and Image Processing 2008*, volume 6822, page 68221J. International Society for Optics and Photonics, 2008.

[124] Lin Liang, Ce Liu, Ying-Qing Xu, Baining Guo, and Heung-Yeung Shum. Real-time texture synthesis by patch-based sampling. *ACM Transactions on Graphics (ToG)*, 20(3): 127–150, 2001.

[125] Weixin Liang and James Zou. Metashift: A dataset of datasets for evaluating contextual distribution shifts and training conflicts. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=MTex8qKavoS.

[126] Liang Liao, Jing Xiao, Zheng Wang, Chia-Wen Lin, and Shin'ichi Satoh. Guidance and evaluation: Semantic-aware image inpainting for mixed scenes. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16*, pages 683–700. Springer, 2020.

[127] Library of Congress. Camera Raw Formats (Group Description). https://www.loc.gov/preservation/digital/formats/fdd/fdd000241.shtml, December 2016. URL https://www.loc.gov/preservation/digital/formats/fdd/fdd000241.shtml. Accessed: 2020-11-03.

[128] Chieh Hubert Lin, Yen-Chi Cheng, Hsin-Ying Lee, Sergey Tulyakov, and Ming-Hsuan Yang. InfinityGAN: Towards infinite-pixel image synthesis. In *International Conference on Learning Representations*, 2022.

[129] Jeremiah Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax Weiss, and Balaji Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in Neural Information Processing Systems*, 33: 7498–7512, 2020.

[130] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[131] S Longanbach, MK Miers, EM Keohane, LJ Smith, and JM Walenga. Rodak's hematology: Clinical principles and applications. 2016.

[132] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471, 2022.

[133] Ye Lyu, George Vosselman, Gui-Song Xia, Alper Yilmaz, and Michael Ying Yang. Uavid: A semantic segmentation dataset for uav imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 165:108–119, 2020. ISSN 0924-2716. doi: https://doi.org/10.1016/j.isprsjprs.2020.05.009. URL https://www.sciencedirect.com/science/article/pii/S0924271620301295.

[134] Jun Ma, Yuting He, Feifei Li, Lin Han, Chenyu You, and Bo Wang. Segment anything in medical images. *Nature Communications*, 15(1), January 2024. ISSN

2041-1723. doi: 10.1038/s41467-024-44824-z. URL http://dx.doi.org/10.1038/s41467-024-44824-z.

[135] Hartmut Maennel. Uncertainty estimates and out-of-distribution detection with sine networks. 2019.

[136] Andreas Maier, Harald Köstler, Marco Heisig, Patrick Krauss, and Seung Hee Yang. Known operator learning and hybrid machine learning in medical imaging — a review of the past, the present, and the future. *arXiv*, 2108.04543, 2021.

[137] Andreas Maier, Harald Köstler, Marco Heisig, Patrick Krauss, and Seung Hee Yang. Known operator learning and hybrid machine learning in medical imaging—a review of the past, the present, and the future. *Progress in Biomedical Engineering*, 2022.

[138] Andreas K Maier, Christopher Syben, Bernhard Stimpel, Tobias Würfl, Mathis Hoffmann, Frank Schebesch, Weilin Fu, Leonid Mill, Lasse Kling, and Silke Christiansen. Learning with known operators reduces maximum error bounds. *Nature machine intelligence*, 1(8): 373–380, 2019.

[139] Maitiniyazi Maimaitijiang, Vasit Sagan, Paheding Sidike, Sean Hartling, Flavio Esposito, and Felix B. Fritschi. Soybean yield prediction from uav using multimodal data fusion and deep learning. *Remote Sensing of Environment*, 237:111599, 2020. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2019.111599. URL https://www.sciencedirect.com/science/article/pii/S0034425719306194.

[140] MT Makler, CJ Palmer, and AL Ager. A review of practical techniques for the diagnosis of malaria. *Annals of Tropical Medicine and Parasitology*, 92(4):419–433, 1998.

[141] C Matek, S Schwarz, C Marr, and K Spiekermann. A single-cell morphological dataset of leukocytes from aml patients and non-malignant controls (aml-cytomorphology_lmu). *The Cancer Imaging Archive (TCIA)*, 2019.

[142] Christian Matek and Carsten Marr. Robustness evaluation of a convolutional neural network for the classification of single cells in acute myeloid leukemia. In *ICLR 2021, RobustML workshop*, 2020.

[143] Christian Matek, Simone Schwarz, Karsten Spiekermann, and Carsten Marr. Human-level recognition of blast cells in acute myeloid leukaemia with convolutional neural networks. *Nature Machine Intelligence*, 1(11):538–544, 2019.

[144] Casey Meehan, Kamalika Chaudhuri, and Sanjoy Dasgupta. A non-parametric test to detect data-copying in generative models. *CoRR*, abs/2004.05675, 2020.

[145] Puria Azadi Moghadam, Sanne Van Dalen, Karina C. Martin, Jochen Lennerz, Stephen Yip, Hossein Farahani, and Ali Bashashati. A morphology focused diffusion probabilistic model for synthesis of histopathology images. *CoRR*, abs/2209.13167, 2022. URL `https://doi.org/10.48550/arXiv.2209.13167`.

[146] Krikamol Muandet. Impossibility of collective intelligence, 2022. URL `https://arxiv.org/abs/2206.02786`.

[147] Zachary Nado, Neil Band, Mark Collier, Josip Djolonga, Michael Dusenberry, Sebastian Farquhar, Angelos Filos, Marton Havasi, Rodolphe Jenatton, Ghassen Jerfel, Jeremiah Liu, Zelda Mariet, Jeremy Nixon, Shreyas Padhy, Jie Ren, Tim Rudner, Yeming Wen, Florian Wenzel, Kevin Murphy, D. Sculley, Balaji Lakshminarayanan, Jasper Snoek, Yarin Gal, and Dustin Tran. Uncertainty Baselines: Benchmarks for uncertainty & robustness in deep learning. *arXiv preprint arXiv:2106.04015*, 2021.

[148] Rose Nakasi, Ernest Mwebaze, Aminah Zawedde, Jeremy Tusubira, Benjamin Akera, and Gilbert Maiga. A new approach for microscopic diagnosis of malaria parasites in thick blood smears using pre-trained deep learning models. *SN Applied Sciences*, 2(7):1–7, 2020.

[149] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? *arXiv preprint arXiv:1810.09136*, 2018.

[150] Seonghyeon Nam, Abhijith Punnappurath, Marcus A. Brubaker, and Michael S. Brown. Learning srgb-to-raw-rgb de-rendering with content-aware metadata. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17704–17713, June 2022.

[151] Rang Nguyen, Dilip K Prasad, and Michael S Brown. Raw-to-raw: Mapping between image sensor color responses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3398–3405, 2014.

[152] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

[153] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of

Proceedings of Machine Learning Research, pages 16784–16804. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/nichol22a.html.

[154] Yaniv Nikankin, Niv Haim, and Michal Irani. SinFusion: Training diffusion models on a single image or video. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 26199–26214. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/nikankin23a.html.

[155] David A Nix and Andreas S Weigend. Estimating the mean and variance of the target probability distribution. In *Proceedings of 1994 ieee international conference on neural networks (ICNN'94)*, volume 1, pages 55–60. IEEE, 1994.

[156] USDOT NSTC. Ensuring american leadership in automated vehicle technologies: Automated vehicles 4.0. *Las Vegas. Recuperado el*, 25:2020–02, 2020.

[157] Luis Oala, Jana Fehr, Luca Gilli, Pradeep Balachandran, Alixandro Werneck Leite, Saul Calderon-Ramirez, Danny Xie Li, Gabriel Nobis, Erick Alejandro Muñoz Alvarado, Giovanna Jaramillo-Gutierrez, Christian Matek, Arun Shroff, Ferath Kherif, Bruno Sanguinetti, and Thomas Wiegand. Ml4h auditing: From paper to practice. In Emily Alsentzer, Matthew B. A. McDermott, Fabian Falck, Suproteem K. Sarkar, Subhrajit Roy, and Stephanie L. Hyland, editors, *Proceedings of the Machine Learning for Health NeurIPS Workshop*, volume 136 of *Proceedings of Machine Learning Research*, pages 280–317. PMLR, 11 Dec 2020. URL http://proceedings.mlr.press/v136/oala20a.html.

[158] Luis Oala, Jana Fehr, Luca Gilli, Pradeep Balachandran, Alixandro Werneck Leite, Saul Calderon-Ramirez, Danny Xie Li, Gabriel Nobis, Erick Alejandro Muñoz Alvarado, Giovanna Jaramillo-Gutierrez, et al. Ml4h auditing: From paper to practice. In *Machine learning for health*, pages 280–317. PMLR, 2020.

[159] Luis Oala, Cosmas Heiß, Jan Macdonald, Maximilian März, Gitta Kutyniok, and Wojciech Samek. Detecting failure modes in image reconstructions with interval neural network uncertainty. *International Journal of Computer Assisted Radiology and Surgery*, 16(12): 2089–2097, 2021.

[160] Luis Oala, Marco Aversa, Gabriel Nobis, Kurt Willis, Yoan Neuenschwander, Michèle Buck, Christian Matek, Jerome Extermann, Enrico Pomarico, Wojciech Samek, Roderick Murray-Smith, Christoph Clausen, and Bruno Sanguinetti. Data models for dataset drift controls in machine learning with optical images. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

[161] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems*, pages 3235–3246, 2018.

[162] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30, 2017.

[163] Antonio Parziale, Monica Agrawal, Shengpu Tang, Kristen Severson, Luis Oala, Adarsh Subbaswamy, Sayantan Kumar, Elora Schoerverth, Stefan Hegselmann, Helen Zhou, Ghada Zamzmi, Purity Mugambi, Elena Sizikova, Girmaw Abebe Tadesse, Yuyin Zhou, Taylor Killian, Haoran Zhang, Fahad Kamran, Andrea Hobby, Mars Huang, Ahmed Alaa, Harvineet Singh, Irene Y. Chen, and Shalmali Joshi. Machine learning for health (ml4h) 2022. In Antonio Parziale, Monica Agrawal, Shalmali Joshi, Irene Y. Chen, Shengpu Tang, Luis Oala, and Adarsh Subbaswamy, editors, *Proceedings of the 2nd Machine Learning for Health symposium*, volume 193 of *Proceedings of Machine Learning Research*, pages 1–11. PMLR, 28 Nov 2022. URL `https://proceedings.mlr.press/v193/parziale22a.html`.

[164] PerkinElmer. *TotalChrom Workstation User's Guide - Volume I*. PerkinElmer. URL `https://www.perkinelmer.com/CMSResources/Images/44-74577MAN_TotalChromWorkstationVolume1.pdf`.

[165] Buu Phan, Fahim Mannan, and Felix Heide. Adversarial imaging pipelines. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16051–16061, 2021.

[166] Nicolas Pielawski and Carolina Wählby. Introducing hann windows for reducing edge-effects in patch-based image segmentation. *PloS one*, 15(3):e0229839, 2020.

[167] Emma Pierson, David M Cutler, Jure Leskovec, Sendhil Mullainathan, and Ziad Obermeyer. An algorithmic approach to reducing unexplained pain disparities in underserved populations. *Nature Medicine*, 27(1):136–140, 2021.

[168] Enrico Pomarico, Cédric Schmidt, Florian Chays, David Nguyen, Arielle Planchette, Audrey Tissot, Adrien Roux, Laura Batti, Christoph Clausen, Theo Lasser, et al. Statistical distortion of supervised learning predictions in optical microscopy induced by image compression. *Scientific reports*, 12(1):1–10, 2022.

[169] Mahdieh Poostchi, Kamolrat Silamut, Richard J. Maude, Stefan Jaeger, and George Thoma. Image analysis and machine learning for detecting malaria. *Translational Research*, 194:36–55, 2018. ISSN 1931-5244. doi: https://doi.org/10.1016/j.trsl.2017.12.004. URL `https://www.sciencedirect.com/science/article/pii/S193152441730333X`. In-Depth Review: Diagnostic Medical Imaging.

[170] Matt Poyser, Amir Atapour-Abarghouei, and Toby P Breckon. On the impact of lossy image and video compression on the performance of deep convolutional neural network architectures. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 2830–2837. IEEE, 2021.

[171] Ahmad B Qasim, Ivan Ezhov, Suprosanna Shit, Oliver Schoppe, Johannes C Paetzold, Anjany Sekuboyina, Florian Kofler, Jana Lipkova, Hongwei Li, and Bjoern Menze. Red-gan: Attacking class imbalance via conditioned generation. yet another medical imaging perspective. In *Medical Imaging with Deep Learning*, pages 655–668. PMLR, 2020.

[172] Adalberto Claudio Quiros, Roderick Murray-Smith, and Ke Yuan. PathologyGAN: learning deep representations of cancer tissue. *Journal of Machine Learning for Biomedical Imaging*, 4:1–48, 2021.

[173] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[174] Jonathan Ragan-Kelley, Connelly Barnes, Andrew Adams, Sylvain Paris, Frédo Durand, and Saman Amarasinghe. Halide: a language and compiler for optimizing parallelism, locality, and recomputation in image processing pipelines. *Acm Sigplan Notices*, 48(6): 519–530, 2013.

[175] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

[176] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2): 3, 2022.

[177] S. Ratnasingam. Deep camera: A fully convolutional neural network for image signal processing. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3868–3878, Los Alamitos, CA, USA, oct 2019. IEEE Computer Society. doi: 10.1109/ICCVW.2019.00480. URL `https://doi.ieeecomputersociety.org/10.1109/ICCVW.2019.00480`.

[178] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

[179] E. Riba, D. Mishkin, D. Ponsa, E. Rublee, and G. Bradski. Kornia: an open source differentiable computer vision library for pytorch. In *Winter Conference on Applications of Computer Vision*, 2020. URL https://arxiv.org/pdf/1910.02190.pdf.

[180] Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger R Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N Galtier, Bennett A Landman, Klaus Maier-Hein, et al. The future of digital health with federated learning. *NPJ digital medicine*, 3(1):119, 2020.

[181] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[182] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[183] Matteo Ronchetti. Torchradon: Fast differentiable routines for computed tomography. *arXiv*, 2009.14788, 2020.

[184] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[185] Andy Rowlands. *Physics of digital photography*. IOP Publishing, 2017.

[186] Gert Rudolph and Uwe Voelzke. Three sensor types drive autonomous vehicles, 2017. URL https://www.fierceelectronics.com/components/three-sensor-types-drive-autonomous-vehicles.

[187] Lukas Ruff, Jacob R Kauffmann, Robert A Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Thomas G Dietterich, and Klaus-Robert Müller. A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, 2021.

[188] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023.

[189] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. volume 115, page 211–252, USA, dec 2015. Kluwer Academic Publishers. doi: 10.1007/s11263-015-0816-y. URL https://doi.org/10.1007/s11263-015-0816-y.

[190] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vision*, 115(3):211–252, dec 2015. ISSN 0920-5691. doi: 10.1007/s11263-015-0816-y. URL https://doi.org/10.1007/s11263-015-0816-y.

[191] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–10, 2022.

[192] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

[193] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4713–4726, 2022.

[194] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[195] Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. *Advances in Neural Information Processing Systems*, 32, 2019.

[196] Hadi Salman, Greg Yang, Huan Zhang, Cho-Jui Hsieh, and Pengchuan Zhang. A convex relaxation barrier to tight robustness verification of neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[197] Nithya Sambasivan, Shivani Kapania, Hannah Highfill, Diana Akrong, Praveen Paritosh, and Lora M Aroyo. "everyone wants to do the model work, not the data work": Data cascades in high-stakes ai. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380966. URL https://doi.org/10.1145/3411764.3445518.

[198] Nithya Sambasivan, Shivani Kapania, Hannah Highfill, Diana Akrong, Praveen Paritosh, and Lora M Aroyo. "everyone wants to do the model work, not the data work": Data cascades in high-stakes ai. In *proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–15, 2021.

[199] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Robust and communication-efficient federated learning from non-iid data. *IEEE transactions on neural networks and learning systems*, 31(9):3400–3413, 2019.

[200] Felix Sattler, Klaus-Robert Müller, Thomas Wiegand, and Wojciech Samek. On the byzantine robustness of clustered federated learning. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8861–8865. IEEE, 2020.

[201] Florian Schiffers, Zekuan Yu, Steve Arguin, Andreas Maier, and Qiushi Ren. Synthetic fundus fluorescein angiography using deep neural networks. In Andreas Maier, Thomas M. Deserno, Heinz Handels, Klaus Hermann Maier-Hein, Christoph Palm, and Thomas Tolxdorff, editors, *Bildverarbeitung für die Medizin 2018*, pages 234–238, Berlin, Heidelberg, 2018. Springer Berlin Heidelberg. ISBN 978-3-662-56537-7.

[202] Felix Schill. pyraw. `https://github.com/fschill/pyraw`, 2015.

[203] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. *Advances in Neural Information Processing Systems*, 33:11539–11551, 2020.

[204] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5b: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL `https://openreview.net/forum?id=M3Y74vmsMcY`.

[205] Christopher G Schwarz, Walter K Kremers, Terry M Therneau, Richard R Sharp, Jeffrey L Gunter, Prashanthi Vemuri, Arvin Arani, Anthony J Spychalla, Kejal Kantarci, David S Knopman, et al. Identification of anonymous mri research participants with face-recognition software. *New England Journal of Medicine*, 381(17):1684–1686, 2019.

[206] David Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Francois Crespo, and Dan Dennison. Hidden technical debt in machine learning systems. *Advances in neural information processing systems*, 28, 2015.

[207] Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3), 2008.

[208] Zhan Shi, Xu Zhou, Xipeng Qiu, and Xiaodan Zhu. Improving image captioning with better use of captions. *arXiv preprint arXiv:2006.11807*, 2020.

[209] Aman Shrivastava and P. Thomas Fletcher. Nasdm: Nuclei-aware semantic histopathology image generation using diffusion models. *CoRR*, abs/2303.11477, 2023. URL `https://doi.org/10.48550/arXiv.2303.11477`.

[210] Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. Model dementia: Generated data makes models forget. *arXiv e-prints*, pages arXiv–2305, 2023.

[211] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

[212] Nimit Sohoni, Jared Dunnmon, Geoffrey Angus, Albert Gu, and Christopher Ré. No subclass left behind: Fine-grained robustness in coarse-grained classification problems. *Advances in Neural Information Processing Systems*, 33:19339–19352, 2020.

[213] Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6048–6058, 2023.

[214] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2020.

[215] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020.

[216] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2020.

[217] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=PxTIG12RRHS`.

[218] Skyler Speakman, Sriram Somanchi, Edward McFowland III, and Daniel B Neill. Penalized fast subset scanning. *Journal of Computational and Graphical Statistics*, 25(2):382–404, 2016.

[219] Maximilian Springenberg, Annika Frommholz, Markus Wenzel, Eva Weicken, Jackie Ma, and Nils Strodthoff. From modern cnns to vision transformers: Assessing the performance, robustness, and classification strategies of deep learning models in histopathology. *Medical Image Analysis*, 87:102809, 2023. ISSN 1361-8415.

[220] Adarsh Subbaswamy, Roy Adams, and Suchi Saria. Evaluating model robustness and stability to dataset shift. In Arindam Banerjee and Kenji Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 2611–2619. PMLR, 13–15 Apr 2021. URL https://proceedings.mlr.press/v130/subbaswamy21a.html.

[221] Adarsh Subbaswamy, Roy Adams, and Suchi Saria. Evaluating model robustness and stability to dataset shift. In *International Conference on Artificial Intelligence and Statistics*, pages 2611–2619. PMLR, 2021.

[222] Shenghuan Sun, Gregory Goldgof, Atul Butte, and Ahmed Alaa. Aligning synthetic medical images with clinical knowledge using human feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=qlnlamFQEa.

[223] Susan M. Swetter. Artificial intelligence may improve melanoma detection. *Dermatology Times*, 41(9):36, 2020. URL https://cdn.sanity.io/files/0vv8moc6/dermatologytimes/4ba31530532b36aaeb80506db61bb5691d841d06.pdf.

[224] Christopher Syben, Markus Michen, Bernhard Stimpel, Stephan Seitz, Stefan Ploner, and Andreas K Maier. Pyro-nn: Python reconstruction operators in neural networks. *Medical physics*, 46(11):5110–5115, 2019.

[225] Nai-Sheng Syu, Yu-Sheng Chen, and Yung-Yu Chuang. Learning deep convolutional networks for demosaicing. *arXiv*, 1802.03769, 2018.

[226] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[227] Shinsuke Tani, Yasuhiro Fukunaga, Saori Shimizu, Munenori Fukunishi, Kensuke Ishii, and Kosei Tamiya. Color Standardization Method and System for Whole Slide Imaging Based on Spectral Sensing. *Analytical Cellular Pathology*, 35(2):107–115, 2012. ISSN 2210-7177, 2210-7185. doi: 10.1155/2012/154735. URL http://www.hindawi.com/journals/acp/2012/154735/.

[228] Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification.

In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. URL https://arxiv.org/abs/2007.00644.

[229] David Tellez, Geert Litjens, Péter Bándi, Wouter Bulten, John-Melle Bokhorst, Francesco Ciompi, and Jeroen van der Laak. Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology. *Medical Image Analysis*, 58:101544, 2019. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2019.101544. URL https://www.sciencedirect.com/science/article/pii/S1361841519300799.

[230] T Terwilliger and MJBCJ Abdul-Hay. Acute lymphoblastic leukemia: a comprehensive review and 2017 update. *Blood cancer journal*, 7(6):e577–e577, 2017.

[231] Chunwei Tian, Lunke Fei, Wenxian Zheng, Yong Xu, Wangmeng Zuo, and Chia-Wen Lin. Deep learning on image denoising: An overview. *Neural Networks*, 131:251–275, 2020. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2020.07.025. URL https://www.sciencedirect.com/science/article/pii/S0893608020302665.

[232] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *Advances in Neural Information Processing Systems*, 33:6827–6839, 2020.

[233] Daniel Shu Wei Ting, Louis R Pasquale, Lily Peng, John Peter Campbell, Aaron Y Lee, Rajiv Raman, Gavin Siew Wei Tan, Leopold Schmetterer, Pearse A Keane, and Tien Yin Wong. Artificial intelligence and deep learning in ophthalmology. *British Journal of Ophthalmology*, 103(2):167–175, 2019. ISSN 0007-1161. doi: 10.1136/bjophthalmol-2018-313173. URL https://bjo.bmj.com/content/103/2/167.

[234] AAMI TIR57. Principles for medical device security—risk management. *Arlington, VA: Association for the Advancement of Medical Instrumentation*, 2016.

[235] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[236] Yan Tong, Wei Lu, Yue Yu, and Yin Shen. Application of machine learning in ophthalmic imaging modalities. *Eye and Vision*, 7(1):1–15, 2020.

[237] Peter Toth, Danilo Jimenez Rezende, Andrew Jaegle, Sébastien Racanière, Aleksandar Botev, and Irina Higgins. Hamiltonian generative networks. *arXiv preprint arXiv:1909.13789*, 2019.

[238] Ethan Tseng, Ali Mosleh, Fahim Mannan, Karl St-Arnaud, Avinash Sharma, Yifan Peng, Alexander Braun, Derek Nowrouzezahrai, Jean-François Lalonde, and Felix Heide. Differentiable compound optics and processing pipeline optimization for end-to-end camera

design. *ACM Trans. Graph.*, 40(2), June 2021. ISSN 0730-0301. doi: 10.1145/3446791. URL https://doi.org/10.1145/3446791.

[239] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9446–9454, 2018.

[240] Hamed Valizadegan, Miguel J. S. Martinho, Laurent S. Wilkens, Jon M. Jenkins, Jeffrey C. Smith, Douglas A. Caldwell, Joseph D. Twicken, Pedro C. L. Gerum, Nikash Walia, Kaylie Hausknecht, Noa Y. Lubin, Stephen T. Bryson, and Nikunj C. Oza. ExoMiner: A highly accurate and explainable deep learning classifier that validates 301 new exoplanets. *The Astrophysical Journal*, 926(2):120, feb 2022. doi: 10.3847/1538-4357/ac4399. URL https://doi.org/10.3847/1538-4357/ac4399.

[241] Joost Van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. In *International conference on machine learning*, pages 9690–9700. PMLR, 2020.

[242] Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant cnns for digital pathology. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part II 11*, pages 210–218. Springer, 2018.

[243] GJJ Verhoeven. It's all about the format–unleashing the power of raw aerial photography. *International Journal of Remote Sensing*, 31(8):2009–2042, 2010.

[244] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

[245] Nikhil Vyas, Sham Kakade, and Boaz Barak. Provable copyright protection for generative models. *arXiv preprint arXiv:2302.10870*, 2023.

[246] Hao Wang, Zakhary Kaplan, Di Niu, and Baochun Li. Optimizing federated learning on non-iid data with reinforcement learning. In *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, pages 1698–1707, 2020. doi: 10.1109/INFOCOM41043.2020.9155494.

[247] Qiwei Wang, Shusheng Bi, Minglei Sun, Yuliang Wang, Di Wang, and Shaobao Yang. Deep learning approach to peripheral leukocyte recognition. *PloS one*, 14(6):e0218808, 2019.

[248] Xinlong Wang, Xiaosong Zhang, Yue Cao, Wen Wang, Chunhua Shen, and Tiejun Huang. Seggpt: Segmenting everything in context. *arXiv preprint arXiv:2304.03284*, 2023.

[249] Li-Yi Wei and Marc Levoy. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 479–488, 2000.

[250] CL Wilson and MD Garris. Handprinted character database. technical report special database 1. Technical report, National Institute of Standards and Technology, 1990.

[251] MacIej Wojtkowski, Tomasz Bajraszewski, Iwona GorczyÅ„ska, Piotr Targowski, Andrzej Kowalczyk, Wojciech Wasilewski, and CzesÅ‚aw Radzewicz. Ophthalmic imaging by spectral optical coherence tomography. *American Journal of Ophthalmology*, 138(3): 412–419, 2004. ISSN 0002-9394. doi: https://doi.org/10.1016/j.ajo.2004.04.049. URL https://www.sciencedirect.com/science/article/pii/S0002939404004635.

[252] Logan G Wright, Tatsuhiro Onodera, Martin M Stein, Tianyu Wang, Darren T Schachter, Zoey Hu, and Peter L McMahon. Deep physical neural networks trained with backpropagation. *Nature*, 601(7894):549–555, 2022.

[253] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33:6256–6268, 2020.

[254] Kaidi Xu, Zhouxing Shi, Huan Zhang, Yihan Wang, Kai-Wei Chang, Minlie Huang, Bhavya Kailkhura, Xue Lin, and Cho-Jui Hsieh. Automatic perturbation analysis for scalable certified robustness and beyond. *Advances in Neural Information Processing Systems*, 33:1129–1141, 2020.

[255] Zongben Xu and Jian Sun. Image inpainting by patch propagation using patch sparsity. *IEEE transactions on image processing*, 19(5):1153–1165, 2010.

[256] Yuan Xue, Jiarong Ye, Qianying Zhou, L Rodney Long, Sameer Antani, Zhiyun Xue, Carl Cornwell, Richard Zaino, Keith C Cheng, and Xiaolei Huang. Selective synthetic augmentation with histogan for improved histopathology image classification. *Medical image analysis*, 67:101816, 2021.

[257] Zhuolin Yang, Linyi Li, Xiaojun Xu, Bhavya Kailkhura, and Bo Li. On the certified robustness for ensemble models and beyond, 2021. URL https://openreview.net/forum?id=IUYthV32lbK.

[258] De Jong Yeong, Gustavo Velasco-Hernandez, John Barry, and Joseph Walsh. Sensor and sensor fusion technology in autonomous vehicles: A review. *Sensors*, 21(6), 2021. ISSN 1424-8220. doi: 10.3390/s21062140. URL https://www.mdpi.com/1424-8220/21/6/2140.

[259] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514, 2018.

[260] Farhad Ghazvinian Zanjani, Svitlana Zinger, Bastian Piepers, Saeed Mahmoudpour, Peter Schelkens, and Peter H. N. de With. Impact of JPEG 2000 compression on deep convolutional neural networks for metastatic cancer detection in histopathological images. *Journal of Medical Imaging*, 6(2):1 – 9, 2019. doi: 10.1117/1.JMI.6.2.027501. URL `https://doi.org/10.1117/1.JMI.6.2.027501`.

[261] ZEISS. *Exporting Images and Movies in ZEN Blue*. ZEISS. URL `https://www.zeiss.com/content/dam/Microscopy/us/download/pdf/zen-software-education-center/exporting-images-and-movies-in-zen-blue.pdf`.

[262] Runtian Zhai, Chen Dan, Di He, Huan Zhang, Boqing Gong, Pradeep Ravikumar, Cho-Jui Hsieh, and Liwei Wang. Macer: Attack-free and scalable robust training via maximizing certified radius. In *International Conference on Learning Representations*, 2019.

[263] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3836–3847, October 2023.

[264] Qinsheng Zhang, Jiaming Song, Xun Huang, Yongxin Chen, and Ming yu Liu. Diffcollage: Parallel generation of large content with diffusion models. In *CVPR*, 2023.

[265] Shengyu Zhao, Jonathan Cui, Yilun Sheng, Yue Dong, Xiao Liang, Eric I Chang, and Yan Xu. Large scale image completion via co-modulated generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2021.

[266] Bo Zhu, Jeremiah Z Liu, Stephen F Cauley, Bruce R Rosen, and Matthew S Rosen. Image reconstruction by domain-transform manifold learning. *Nature*, 555(7697):487, 2018.

[267] P. Zhu, L. Wen, D. Du, X. Bian, H. Fan, Q. Hu, and H. Ling. Detection and tracking meet drones challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (01): 1–1, October 2021. ISSN 1939-3539. doi: 10.1109/TPAMI.2021.3119563.