TAILORED DEEP LEARNING FOR ENHANCED PERFORMANCE IN RESOURCE-CONSTRAINED SETTINGS

Aman Shrivastava

A Dissertation submitted to the Graduate Faculty
of the University of Virginia in Candidacy for the Degree of
Doctor of Philosophy

Department of Computer Science

University of Virginia

Sep 2024

Tom Fletcher, Co-Advisor
Vicente Ordóñez, Co-Advisor
Yangfeng Ji, Member / Chair
Donald E. Brown, Member
Miaomiao Zhang, Member
Sana Syed, Member

Acknowledgments

First and foremost, I express my deepest gratitude to my advisors, Prof. Tom Fletcher and Prof. Vicente Ordóñez, for their unwavering support, insightful guidance, and encouragement throughout this journey. Their expertise and mentorship have been invaluable, and I am truly blessed to have had the opportunity to learn and grow under your supervision. I could not have asked for better advisors in my doctoral journey. I am profoundly grateful to my family for their unconditional love, patience, and understanding. To my parents, your belief in me and my judgment has been a great source of strength and motivation. Thank you for being my constant support system and for reminding me of the importance of perseverance and balance. To my sister, Sanjana, your constant encouragement and belief in me have meant the world. Thank you for always being there in stressful times and for inspiring me with your own strength of resolve. Having you here during this journey has made an immeasurable impact and made the process infinitely more fun. To my partner, Prerana, your companionship, laughter, and encouragement have lightened the load and made this experience truly memorable. I am especially thankful for your reminders to work as well as celebrate small victories along the way. To Machana, thanks for adding all the humor and facilitating regular getaways which were an essential breaks to keep me going. This thesis is a culmination of the support and kindness I have received from so many. To each one of you, I am eternally grateful.

Contents

1

Introduction

In recent years, the field of multi-disciplinary artificial intelligence has seen tremendous progress driven by computer vision and natural language processing advancements. Recent work in these fields has enabled computers to better understand and interpret visual information from images, and accurately comprehend and generate human language. Extending these capabilities to data- and resource-constrained settings presents unique challenges but can have transformative change. These innovations have also significantly transformed the healthcare landscape by offering unprecedented opportunities to enhance patient care, improve diagnostic accuracy, and optimize treatment strategies. In this thesis, I present my contributions to tackle major challenges in tailored application of state-of-the-art machine learning techniques to low data and resource settings.

One key challenge is adapting general-use, real-world models into smaller, domain-specific models. This is achieved via knowledge distillation. Foundation models are large-scale artificial intelligence models that have been trained on vast amounts of general data. They possess a broad understanding of language and images but are not tailored to specific tasks. Distilling information from these models means extracting and refining the valuable knowledge they contain to create smaller, more specialized models suited for specific applications. In my work, we address this challenge by introducing a flexible knowledge distillation method that can be optimized in severe

resource-constrained settings and can work with a varied range of model architectures.

Another challenge in multi-disciplinary application of computer vision is a lack to annotations. However, textual annotations are relatively easy to obtain and are often abundant. Leveraging these readily available text annotations reduces the need for extensive manual labeling of images, which is both time-consuming and costly. By using this existing textual data, vision models can efficiently learn to associate visual features with semantic concepts, improving their performance and generalization. This process requires what's known as vision-language alignment, where the model learns to associate visual features from images with relevant textual descriptions using large paired image-text datasets. In this work, we design alignment models to learn effectively from small annotated datasets. We present a vision-language alignment objective that is designed to be trained with significantly reduced amounts of data and using smaller batch sizes while providing comparable or superior performance on standard benchmarks against other methods trained in similar settings.

In medical image analysis, machine learning models often suffer from bias and reduced performance due to variations in data from different sample sources. In histopathology, these differences manifest as stain variations which are discernible variations in color of the slides that arise from inconsistencies in laboratory procedures, differences in staining reagents, and variations in imaging equipment. These variations can negatively impact the performance of automated image analysis algorithms, such as those used for nuclei detection, tissue segmentation, or disease classification. Such inconsistencies can impair the ability of models to generalize to new, differently stained images. We tackle this problem by introducing a novel adversarial method that executes many-to-one domain stain normalization. The training objective is designed to make sure that the structure of the image is preserved during translation. Our

method demonstrates impressive performance in preserving the structural integrity of images while transferring the stain distributions when tested on duodenum biopsies.

Model training and diagnostics for medical imaging applications suffers from pervasive lack of data due to ethical and privacy concerns. Specifically in the field of pathology, histopathological analysis relies on hematoxylin and eosin (H&E) stained biopsies for microscopic inspection to identify diseases, including cancers, with diagnosis heavily dependent on the pathologist's training and exposure to various disease subtypes. This presents challenges, especially with rare variants, which are harder to identify visually. Recently, deep learning methods have been developed to support diagnosis, particularly through segmentation models that identify nuclei types. Generative models can generate histopathology images with specific characteristics, addressing the imbalance in datasets and reducing bias in model training. These models hold potential to improve diagnosis by aiding both deep learning systems and human pathologists, and synthetic datasets can help overcome privacy concerns in medical data sharing. Conditional generation of annotated data adds value by alleviating the high costs of labeling medical images. This synthetic data can also be used to train machine learning models without compromising patient privacy, thereby overcoming the limitations posed by scarce or sensitive data.

Overall, this thesis makes contributions towards extending state-of-the-art machine learning techniques to low data and resource settings on multiple fronts. Our proposed knowledge distillation techniques facilitate seamless knowledge transfer between neural networks, enabling effective model compression and transfer learning from foundational vision and language models. We present methodology for leveraging textual captions for vision model pretraining in resource and data-constrained environments. We also present a stain-normalization methodology specifically addressing the prob-

lem of variation in visual appearance of digital slides due to differences in staining mechanisms across sites. Further, our examination of conditional diffusion models for generative modeling of histopathological tissue slides holds substantial promise for the field of pathology. By synthesizing tissue patches conditioned on nuclei masks, this approach presents a pioneering solution for enhancing the accuracy and efficiency of histopathological analysis by addressing the pervasive lack of annotated data in medical imaging analysis. As such, my thesis statement is:

Tailored deep learning methods enable consistent and critical progress toward enhancing reasoning capabilities, particularly in data- and resource-constrained settings like healthcare. Improvements in knowledge distillation using information maximization enables cheaper optimization and domain-transfer for large vision models. Information-efficient contrastive learning for aligning images with textual data leads to better performance on small datasets. Structure-preserving generative adversarial networks help minimize visual variations in medical images. Conditional diffusion can be used to develop end-to-end models for synthesizing inherently-annotated histology tissue samples with pixel-perfect nuclei localization.

1.1 Contributions

This thesis makes the following novel contributions:

1. Knowledge distillation
    (a) We developed three flexible mutual information maximization objectives for knowledge distillation.
    (b) Our method is effective across a wide range of model pairs and enables learning transferable representations.
2. Contrastive image & text alignment
    (a) Our work allows training multi-modal alignment models in data and resource constrained settings.
    (b) The method achieves state-of-the-art performance in downstream tasks like retrieval, unsupervised object localization, and zero-shot learning.
3. Stain normalization for H&E images
    (a) We designed a structure-preserving cycle consistent architecture for unpaired image to image translation to normalize color distributions.
    (b) The method achieves unprecedented performance in normalizing stain distributions in histology images.
4. Generative modeling for medical imaging
    (a) Our end-to-end method synthesizes unlimited annotated realistic histology tissue samples with pixel-perfect nuclei localization.
    (b) It demonstrates competitive metrics quantitatively, and our expert qualitative evaluations suggest that synthetic patches are comparable to the real set.

2

Estimating and Maximizing Mutual Information for Knowledge Distillation

In this chapter, we discuss Mutual Information Maximization Knowledge Distillation (MIMKD). Our method uses a contrastive objective to simultaneously estimate and maximize a lower bound on the mutual information of local and global feature representations between a teacher and a student network. We demonstrate through extensive experiments that this can be used to improve the performance of low capacity models by transferring knowledge from more performant but computationally expensive models. This can be used to produce better models that can be run on devices with low computational resources. Our method is flexible, we can distill knowledge from teachers with arbitrary network architectures to arbitrary student networks. Our empirical results show that MIMKD outperforms competing approaches across a wide range of student-teacher pairs with different capacities, with different architectures, and when student networks are with extremely low capacity.

2.1 Introduction

Recent machine learning literature has seen a lot of progress driven by deep neural networks. Many such models that achieve state-of-the-art performance on different benchmarks require large amounts of computation and memory capacities (Zehao Huang and N. Wang 2018). To this end, Knowledge Distillation (KD) has been used to transfer knowledge from a stronger teacher network to a smaller and less computationally expensive student network (Buciluǎ, Caruana, and Niculescu-Mizil 2006; G. Hinton, Vinyals, and Dean 2015). We look at knowledge distillation from an information-theoretic perspective and propose Mutual Information Maximization Knowledge Distillation (MIMKD). Multiple approaches have been proposed to estimate the mutual information between high-dimensional continuous variables (Belghazi et al. 2018; Hjelm et al. 2018). Belghazi et al Belghazi et al. (2018) propose a KL-divergence based formulation of mutual information. We observe that this approach can be extended to maximize the mutual information in a contrastive setup. Contrastive methods have had an outsized impact in other problems such as self-supervised learning (T. Chen et al. 2020; He, Fan, et al. 2020), however they rely on sampling a rather large number of paired inputs to optimize their objective functions. We find that by using a Jensen-Shannon divergence (JSD) based formulation we obtain a more stable objective to optimize where the performance is invariant to the number of negative samples while being monotonically related to the true mutual information as also shown in Hjelm et al Hjelm et al. (2018). Recently proposed Contrastive Representation Distillation (CRD) framework (Tian, Krishnan, and Isola 2019) uses a Noise Contrastive Estimation (NCE) objective (Oord, Y. Li, and Vinyals 2018; Gutmann and Hyvärinen 2010) to transfer structured relational knowledge from the teacher to the student. However, a caveat of this approach is that it ignores in-

termediate distillation for feature level information and requires a large number of negative samples requiring large batches (T. Chen et al. 2020) or memory banks (He, Fan, et al. 2020; Z. Wu et al. 2018). We extend this work by using a JSD-based contrastive objective that is insensitive to the number of negative samples. This enables us to impose additional region-consistent local and feature-level constraints with just one negative sample.

We propose three mutual information maximization objectives between the teacher and student networks: (1) *Global information maximization*, which aims to maximize the shared information between the final output representations. This pushes the student network to generate feature vectors that are as rich as the ones generated by the teacher. (2) *Local information maximization*, which pushes the student network to recognize complex patterns from each region of the image that are ultimately useful for classification. This is achieved by maximizing the mutual information between region-specific vectors extracted from an intermediate representation of the student network and the final representation of the teacher network. Finally, (3) *Feature Information Maximization*, which is designed to structurally improve the granular feature-extraction capability of the student by maximizing the mutual information between region-consistent local vectors extracted from intermediate representations of the networks.

Our experimental results demonstrate that these objectives are effective across a wide range of student-teacher pairs and carry out extensive ablation studies of the effect of each proposed objective.

2.2 Background

In this section, we discuss previous efforts in improving knowledge distillation, and in estimating mutual information which are the key areas of contribution of our work.

2.2.1 Knowledge distillation.

The concept of knowledge distillation (KD) was introduced in the works of Buciluǎ et al. (Buciluǎ, Caruana, and Niculescu-Mizil 2006) and later formalized for deep neural networks by Hinton et al. (G. Hinton, Vinyals, and Dean 2015). In knowledge distillation, the goal is to train smaller models that can mimic the performance of larger models. Hinton et al. (G. Hinton, Vinyals, and Dean 2015) proposed a knowledge distillation method in which the student network is trained using soft labels extracted from teacher networks.

Attention transfer (Zagoruyko and Komodakis 2016a) introduced the idea of transferring intermediate attention maps from the teacher to the student network. Fitnets (Romero et al. 2014) also presented the idea of adding more supervision by matching the intermediate representation using regressors. Yim et al. (Yim et al. 2017) formulated the distillation problem using the flow of solution procedure (FSP), which is computed as the gram matrix of features across layers. Sau et al. (Sau and Balasubramanian 2016) proposed to include a noise-based regularizer while training the student with the teacher. Specifically, they perform perturbation in the logits of the teacher as a regularization approach. In Correlation Congruence for Knowledge Distillation (CCKD) (B. Peng et al. 2019), the authors present a framework which transfers not only instance-level information but also the correlation between instances. In CCKD, a Taylor series expansion-based kernel method is proposed

to better capture the correlation between instances. Tung et al. (Tung and Mori 2019) propose a loss that is based on the observation that semantically similar inputs produce similar activation patterns in trained networks. Variational Information Distillation (VID) (Ahn et al. 2019) uses a variational lower-bound for the mutual information between the teacher and the student representations by approximating an intractable conditional distribution using a pre-defined variational distribution.

More closely related to our work are methods that cast knowledge distillation as a mutual information maximization problem. Contrastive representation distillation (CRD) (Tian, Krishnan, and Isola 2019) used a contrastive objective similar to Oord et al. (Oord, Y. Li, and Vinyals 2018) to maximize a lower-bound on mutual information between final representations. The objective used by CRD is a strong lower-bound on the mutual information but requires a significant number of negative samples during training, consequently, requiring large batch-sizes or memory buffers. These practical constraints become even more limiting if mutual information needs to be minimized at the feature-level to enforce regional-supervision during student training. Our work proposes an alternative that bypasses the needed for such large batch-sizes and thus enables to optimize for mutual information through three separate objectives.

2.2.2 Mutual Information Estimation.

Mutual information is a fundamental quantity that measures the relationship between random variables but it is notoriously difficult to measure (Paninski 2003). An exact estimate is only tractable for discrete variables or a small set of problems where the probability distributions are know. However, both the mentioned scenarios are un-

likely for real-world visual datasets. Recently, Mutual Information Neural Estimation (MINE) (Belghazi et al. 2018) demonstrated a strong method for estimation of mutual information between high-dimensional continuous random variables using neural networks and gradient descent. MINE (Belghazi et al. 2018) proposed a general-purpose parametric neural estimator of mutual information based on dual representations of the KL-divergence (Ruderman et al. 2012). Following from MINE (Belghazi et al. 2018), Deep InfoMax (Hjelm et al. 2018) proposed a mutual information based objective for unsupervised representation learning. Deep InfoMax (Hjelm et al. 2018) contends that it is unnecessary to use the exact KL-divergence based formulation of mutual information and demonstrated the use of an alternative formulation based on the Jensen-Shannon divergence (JSD). The authors showed that the JSD based estimator is stable, and does not require a large number of negative samples. In addition, Deep InfoMax (Hjelm et al. 2018) also demonstrated the value of including global and local structure-based mutual information objectives for representation learning. We leverage this line of work in our method to propose a framework for knowledge distillation that leverages both local and global features without significantly adding memory overheads during training.

2.3 Method

In this section, we describe our general framework for model compression or knowledge distillation in a teacher student setup. Consider a stronger teacher network $f_t : X \rightarrow Y$ with trained parameters ϕ and a student network, operating on the same domain, $f_s : X \rightarrow Y$ with parameters θ. Let x be the sample drawn from the data distribution $p(x)$ and $f_t(x)$ & $f_s(x)$ denote the representations extracted from the

pre-classification layer, while $f_t^{cls}(x)$ & $f_s^{cls}(x)$ denote the predicted class-probability distributions from the teacher and the student networks respectively. Now consider a set $\mathcal{R} = \{(f_t^{(k)}(x), f_s^{(k)}(x))\}_{k=1}^{K}$ that contains K pairs of intermediate representations extracted from the networks such that each pair in set $\mathcal{R}$ contains same-sized intermediate representations extracted from the networks, where $m_k \times m_k$ is the size corresponding to the k-th pair in the set. Each location in these 2-dimensional intermediate representations corresponds to a specific region in the input image. Note that we do not include the final representations $f_t(x)$ and $f_s(x)$ in the set $\mathcal{R}$.

Our method focuses on maximizing the mutual information, (1) between final image representations $f_t(x)$ and $f_s(x)$ (global information maximization), (2) between the global image representation from the teacher network $f_t(x)$ and the last intermediate representation from the student network $f_s^{(K)}(x)$ (local information maximization), and (3) between the pairs in set $\mathcal{R}$ (feature information maximization). Figure 2.1 shows an overview of our method.

2.3.1 Mutual Information Maximization

In order to estimate and maximize mutual information between random variables X and Z, we train a neural network to distinguish samples generated from the joint distribution, $P(X, Z)$ and the product of marginals $P(X)P(Z)$. In MINE (Belghazi et al. 2018), the authors use the Donsker-Varadhan (DV) (Donsker and Varadhan 1983) representation of the KL-divergence as the lower bound on the mutual information. Recently, another bound on mutual information, formulated as infoNCE (Oord, Y. Li, and Vinyals 2018) based on Noise-Contrastive Estimation (Gutmann and Hyvärinen 2010), has seen wide adoption in representation learning due to its low variance and

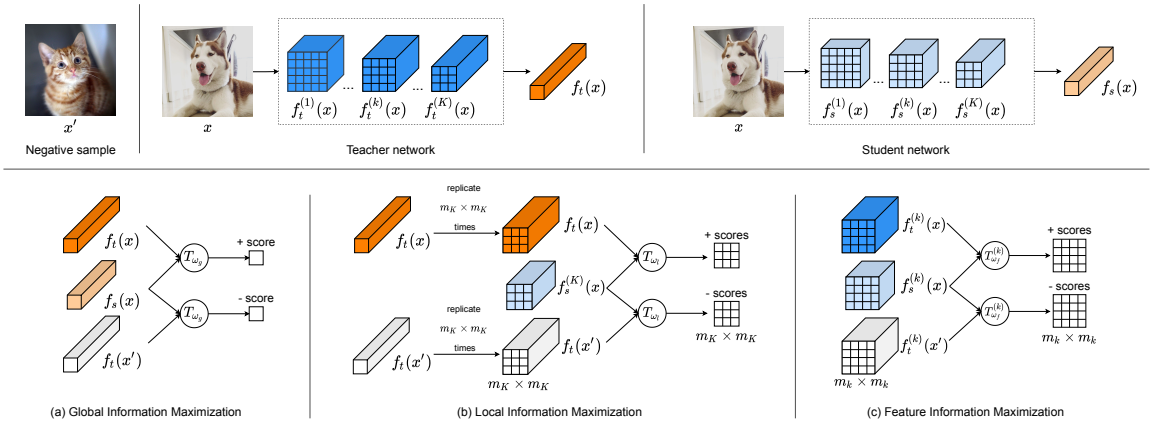


Figure 2.1: Overall schematic of our proposed method for mutual information maximization based knowledge distillation (MIMKD). ***Top:*** Representations generated by teacher and student networks for image x and a negative sample x'. Note that our method uses only one negative sample. ***Bottom:*** **(a)** Positive and negative pairs of final feature vectors are passed into the discriminator function to get scores. **(b)** Teacher's final representation is replicated to match student's last intermediate representation. **(c)** For each group of same-sized intermediate feature maps in set $\mathcal{R}$, positive and negative pairs are passed into a distinct discriminator function to get scores. The positive and negative scores obtained are then used with equation (2) to estimate and maximize a lower-bound on mutual information.

accurate estimate of MI. It is defined as follows;

$$\hat{I}_{\omega}^{InfoNCE}(X;Z) = \mathbb{E}_{P(X,Z)}\left[T_\omega - \mathbb{E}_{P(X)P(Z)}\left[\log \sum T_\omega\right]\right], \tag{2.1}$$

where $T_\omega : \mathcal{X} \times \mathcal{Z} \to \mathbb{R}$ is the discriminator neural network with parameters ω. However, as demonstrated in (Hjelm et al. 2018), both DV and infoNCE require a large number of negative samples during training. Recent works tackle this problem by using a memory-buffer that keeps representations from previous samples in memory to be accessed during training. As implemented in CRD (Tian, Krishnan, and Isola 2019), this can be done if mutual information is maximized only between the final representations of the networks as the dimensions of the representations to be kept

in memory is limited. In this work, we extend this infoNCE based MI maximization framework to include feature and local level information maximization. As a result, we require negative samples for each location in the multiple K intermediate feature maps as well as for the final representations. This becomes unfeasible for most large state-of-the-art architectures. To this end, in our approach we adopt Jensen-Shannon divergence based mutual information estimation, similar to the formulations in (Nowozin, Cseke, and Tomioka 2016) and (Brakel and Bengio 2017). The MI estimate from this JSD-based bound on MI, due to its formulation, is insensitive to the number of negative samples.

$$I(X;Z) \geq \hat{I}^{JSD}_{\omega}(X;Z) = \mathbb{E}_{P(X,Z)}\left[-\log(1+e^{-T_\omega})\right] - \mathbb{E}_{P(X)P(Z)}\left[\log(1+e^{T_\omega})\right]. \quad (2.2)$$

Overall, we optimize the parameters θ of the student network f_s and parameters ω of the critic network T_ω by simultaneously estimating and maximizing mutual information between the representations of the frozen teacher network and the student network.

2.3.2 Global information maximization

Our global objective aims to maximize the mutual information between the richer final representation of the frozen teacher network $f_t(x)$ and the final representation of the student network $f_s(x)$ to encourage the student to learn richer representations. This objective uses a discriminator function T_{ω_g}, where ω_g are the trainable parameters. We use the infoNCE bound for global MI maximization as it is computationally feasible to maintain a memory bank of negative samples due to the lower dimensionality of the

final representations from the networks. We optimize the parameters of the student and the discriminator function simultaneously as:

$$(\hat{\omega_g}, \hat{\theta}) = \underset{\omega_g, \theta}{\operatorname{argmax}} \hat{I}^{\text{infoNCE}}_{\omega_g}(f_t(x), f_s(x)). \tag{2.3}$$

2.3.3 Local information maximization

In this objective we maximize the mutual information between a richer final representation of the teacher network and representations of local regions extracted by the student network. This objective draws from the assertion that the final teacher representations contains valuable information required for downstream classification. Hence, this objective encourages the student network to extract information from local image regions that is ultimately useful for classification.

We enforce this objective between $f_t(x)$ and the last intermediate representation from the student network in the set $\mathcal{R}$. Therefore for $k = K$, $f_s^{(K)}(x)$ is a $m_K \times m_K$ feature map where each location roughly corresponds to an $H/m_K \times W/m_K$ patch in the input image where H, W are the height and width of the image. The representation of each such patch $\{f_s^{(K)}(x)\}_{i,j}$ is then paired with $f_t(x)$, where $i, j \in [1, m_K]$ denotes the specific location in the feature map. The pairs are then used with the mutual information estimator to optimize the parameters as follows:

$$(\hat{\omega_l}, \hat{\theta}) = \underset{\omega_l, \theta}{\operatorname{argmax}} \frac{1}{m_K^2} \sum_{i=1}^{m_K} \sum_{j=1}^{m_K} \hat{I}^{\text{JSD}}_{\omega_l}(f_t(x), \{f_s^{(K)}(x)\}_{i,j}) \tag{2.4}$$

where a discriminator neural network T_{ω_l} with parameters ω_l is used.

2.3.4 Feature Information maximization

This objective aims to maximize the mutual information between region-consistent intermediate representations from the networks. In neural networks, the complexity of captured visual patterns increases towards the later layers (Zeiler and Fergus 2014). Intuitively, to mimic the representational power of the teacher, the student network needs to learn these complex patterns hierarchically. In order to motivate such hierarchical learning, mutual information is maximized between intermediate features at different depths in the networks. This enables the student to learn to identify complex patterns in a bottom-up fashion and systematically learn to generate richer features. Note that within each pair of intermediate feature maps in set $\mathcal{R}$, mutual information is maximized between vectors corresponding to the same location in the image. This information maximization pushes the student network to extract features from each region of the image that share maximum information with the features extracted by the teacher network from the same region. For a pair $(f_t^{(k)}(x), f_s^{(k)}(x)) \in \mathcal{R}$, information is maximized between pairs of region-consistent vectors $\{f_t^{(k)}(x)\}_{i,j}$ and $\{f_s^{(k)}(x)\}_{i,j}$ for each $i, j \in [1, m_k]$ as follows:

$$(\hat{\omega_f}, \hat{\theta}) = \underset{\omega_f, \theta}{\operatorname{argmax}} \frac{1}{K} \frac{1}{m_k^2} \sum_{k=1}^{K} \sum_{i=1}^{m_k} \sum_{j=1}^{m_k} \hat{I}_{\omega_f}^{\mathrm{JSD}}(\{f_t^{(k)}(x)\}_{i,j}, \{f_s^{(k)}(x)\}_{i,j}) \tag{2.5}$$

where a discriminator neural network T_{ω_f} with parameters ω_f is used.

2.3.5 Classification objective

Here the cross-entropy loss is minimized between the output of the classification function $f_s^{cls}(x)$ and the target label y as follows:

$$(\hat{\theta}) = \underset{\theta}{\operatorname{argmin}}\ \mathcal{L}_{CE}(y, f_s^{cls}(x)), \tag{2.6}$$

where $\mathcal{L}_{CE}$ denotes the cross-entropy function.

Our overall objective is a weighted-summation of all the above individual objectives with weights α (cross-entropy loss), λ_g (global MI maximization), λ_l (local MI maximization), and λ_f (feature MI maximization)

2.3.6 Mutual Information Discriminators

The parameterized mutual information discriminator functions (T_{ω_g}, T_{ω_l}, and T_{ω_f}) can be modeled as neural networks. In our experiments, we use two distinct discriminator architectures inspired from the functions presented in Deep InfoMax (Hjelm et al. 2018). For global information maximization, we use the standard project and dot architecture. The representations from both the teacher and the student are first projected using an appropriate projection architecture with a linear shortcut. The dot-product of these projections is then computed to get the score. Positive and negative pairs of representations are passed through the discriminator to get respective scores to be passed into equation 2.2 to get the estimates on the lower bound of the mutual information. Whereas, for local and feature information maximization we use a convolution based architecture as it is cheaper for higher dimensional inputs.

Specifically, for local information maximization, we replicate the final representation from the teacher $f_t(x)$ to match the $m_K \times m_K$ size of the student's last intermediate feature map ($f_s^{(K)}(x)$). The resulting replicated tensor is then concatenated with $f_s^{(K)}(x)$ to get $[f_t(x), f_s^{(K)}(x)]$ which serves as the input for the critic function (ref. table on right). Similarly, consider feature mutual information maximization, for each

pair in the set $\mathcal{R}$ we use a distinct discriminator $T_{\omega_f}^{(k)}$. For a given k, each pair of intermediate feature representations in the set $\mathcal{R}$ are concatenated together to get $[f_t^{(k)}(x), f_s^{(k)}(x)]$. Which is then passed through two convolutional (1×1 kernels and 512 filters) where each layer is followed by a ReLU non-linearity. The output obtained is then further passed into a convolutional layer (1×1 kernels and 1 filter) to give $m_k \times m_k$ scores. Further details are provided in supplementary.

2.3.7 Implementation Details

We adopted the generally established approach for training CNNs on the CIFAR-100 dataset. We use SGD with momentum 0.9, weight decay 5×10^{-4}, and an initial learning rate of 0.05 for a total of 240 epochs with batch-size 64. The learning rate is decayed by 0.1 at the 150th, 180th and the 210th epoch. We used random horizontal flips and random crop for augmenting the dataset during training. For ImageNet, we use the standard PyTorch training scheme for ResNets (He, X. Zhang, et al. 2016). Code implementation will be made public on publication.

2.4 Experiments

In this section, we demonstrate the efficacy of our framework using various ablative and quantitative analyses. We first establish the value of each of our mutual information maximization formulations by performing an extensive ablative study (sec. 2.4.1). Further, we demonstrate the prowess of our distillation framework based on model compression performance in the following setups: (1) Under similar student-teacher network architectures (sec. 2.4.2), (2) under dissimilar architectures (sec. 2.4.3), (3)

under a setting with custom designed shallow student networks (ref. appendix for results), (4) in a larger scale setting on Imagenet (ref. appendix for results), and (5) in terms of transfer learning performance (sec. 2.4.4) as a measure of the transferability of distilled representations. Our model compression experiments are performed on the CIFAR-100 dataset which contains colored natural images of size 32×32. It has 50K training images with 500 images in each of 100 classes and a total of 10K test images. In our experiments, we use standard CNN architectures of varied capacities, such as ResNet (He, X. Zhang, et al. 2016), Wide ResNet (WRN) (Zagoruyko and Komodakis 2016b), MobileNet (Sandler et al. 2018), ShuffleNet (X. Zhang et al. 2018), and VGG (Simonyan and Zisserman 2014). We compare our method with other knowledge distillation methods, such as (1) Knowledge Distillation (KD) (G. Hinton, Vinyals, and Dean 2015), (2) FitNets (Romero et al. 2014), (3) Attention Transfer (AT) (Zagoruyko and Komodakis 2016a), (4) Variational Information Distillation (VID) (Ahn et al. 2019), and (5) Contrastive Representation Distillation (CRD) (Tian, Krishnan, and Isola 2019). We used the following values for hyper-parameters based on a held out set: $\alpha = 1$, $\lambda_g = 1$, $\lambda_l = 0.75$, $\lambda_f = 1$ for all our experiments. The infoNCE bound in CRD as well as our global MI is set to use 4096 negatives. The hyper-parameter choice for other approaches can be found in supplementary. Additionally, in order to demonstrate the scalability of our method, we compare our distillation performance on the ImageNet (Deng et al. 2009) dataset against AT (Zagoruyko and Komodakis 2016a), and KD (G. Hinton, Vinyals, and Dean 2015). ImageNet is a large-scale dataset with 1.2 million training images across 1K classes and a total of 50K validation images.

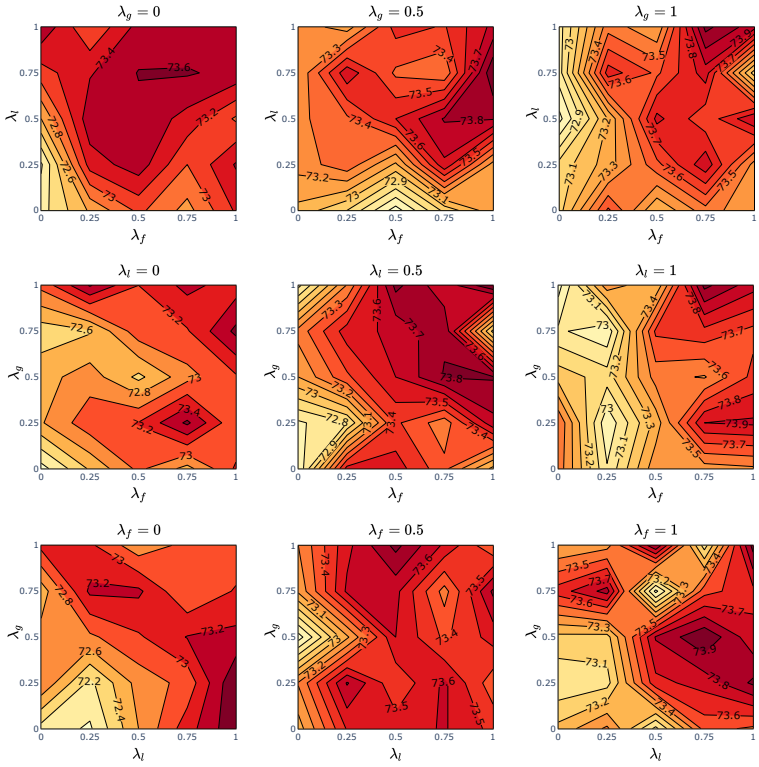


Figure 2.2: Results from the ablation studies on CIFAR100 dataset using a student ResNet-8x4 (baseline acc. 72.44%) with teacher ResNet-32x4 (baseline acc. 79.24%). Contour lines represent the final test accuracy of the student. The study was performed by varying the values of λ_f, λ_g, λ_l from 0 to 1 with increments of 0.25 while α was kept constant at 1. In each plot, the accuracy landscape is shown with λ_g set to a constant value.

2.4.1 Ablation Study

We perform an extensive ablation study to demonstrate the value of each component of our mutual information maximization objective. Ablative study experiments are performed with ResNet-32x4 as the teacher network and ResNet-8x4 as the student network where the baseline accuracy of the teacher is 79.24% and that of the student network is 72.44%. The values of the hyper-parameters λ_g, λ_l and λ_f — that control the weight of the global, local and feature mutual information maximization

objectives respectively – were varied between 0 and 1 with an increment of 0.25 while the weight for the cross-entropy loss, α was set to 1. Note that for this study, we use the JSD-based bound for all MI maximization formulations including for global MI which is not the case for our final competitive models presented further. The contour plots in Figure 2.2 shows the test accuracy landscape with respect to a pair of hyper-parameters when the third hyper-parameter is set to distinct values. For instance, we observe that for any value of λ_g, better performance is achieved towards higher values of both λ_f and λ_l. Similar trends can be observed in all the accuracy landscape plots. Overall, this demonstrates the value of maximizing region-consistent local and feature-level mutual information between representations in addition to just global information maximization. Please refer to the appendix for additional accuracy landscape plots.

2.4.2 Similar CNN Architectures

We perform knowledge distillation from a teacher network to a student network of the same family (e.g. ResNets of different capacities). Table 2.1 presents our results, showing that our method outperforms others in most setups and always obtains gains with respect to student networks. Notice that CRD (Tian, Krishnan, and Isola 2019) is able to slightly surpass the performance of our method in one setup while being close in most cases. We find this encouraging as CRD (Tian, Krishnan, and Isola 2019) uses a similar mutual information maximization based formulation in their distillation objective with a tighter lower-bound. Therefore, if we only use the global objective in our method, CRD (Tian, Krishnan, and Isola 2019) should outperform our method due to its tighter bound. Despite compromising the lower bound on mutual information, MIMKD takes advantage of using region-consistent local and

Student Net.	**WRN-16-1**	**WRN-16-2**	**ResNet-8**	**ResNet-20**	**ResNet-20**	**ResNet-8x4**	**VGG-8**
Teacher Net.	**WRN-40-2**	**WRN-40-2**	**ResNet-110**	**ResNet-110**	**ResNet-56**	**ResNet-32x4**	**VGG-19**
Student Acc.	67.01	72.80	59.63	69.10	69.10	72.44	69.67
Teacher Acc.	75.31 **+8.30**	75.31 **+2.51**	73.82 **+14.19**	73.82 **+4.72**	72.31 **+3.21**	79.24 **+3.80**	74.63 **+4.96**
FitNets	68.35 **+1.34**	73.11 +0.31	60.36 +0.73	69.12 +0.02	69.28 +0.18	73.80 **+1.36**	71.32 **+1.65**
AT	68.49 **+1.48**	73.37 +0.57	60.24 +0.61	70.36 **+1.26**	70.18 **+1.08**	73.20 +0.76	71.71 **+2.04**
VID	68.95 **+1.94**	73.89 **+1.09**	60.44 +0.81	70.32 **+1.22**	70.52 **+1.42**	73.19 +0.75	71.52 **+1.85**
KD	68.24 **+1.23**	73.91 **+1.11**	61.01 **+1.38**	70.32 **+1.22**	70.59 **+1.40**	73.21 +0.77	72.29 **+2.62**
CRD	69.21 **+2.20**	74.17 **+1.37**	60.82 **+1.19**	**71.45** **+2.35**	71.12 **+2.02**	75.21 **+2.77**	73.10 **+3.43**
MIMKD (ours)	**70.20** **+3.19**	**75.16** **+2.36**	**61.81** **+2.18**	71.43 **+2.33**	**71.31** **+2.21**	**75.83** **+3.39**	**73.27** **+3.60**

Table 2.1: Observed test accuracy (in %) of student networks trained with teacher networks of higher capacity but similar architecture on the CIFAR100 dataset using MIMKD and other competing methods. MIMKD shows consistent increases in accuracy for all model pairs and the largest gains overall.

feature-level mutual information maximization.

2.4.3 Dissimilar CNN Architectures

Here, we perform knowledge distillation from a teacher network to a student network with a significantly different architecture. This tests the flexibility methods to adapt to distinct data-abstraction flows of dissimilar neural network architectures. Table 2.2 demonstrates that our method (MIMKD) outperforms other distillation methods in most teacher-student combinations increasing the accuracy of a ShuffleNetV2 by 4.7% while distilling from a much different ResNet-50 model. This demonstrates that our method is able to accommodate significant architectural differences in teacher-

Student Net.	**WRN-16-1**	**WRN-16-2**	**VGG-8**	**ShuffleNetV1**	**ShuffleNetV2**	**MobileNetV2**
Teacher Net.	**ResNet-110**	**ResNet-32x4**	**ResNet-32x4**	**VGG-13**	**ResNet-50**	**VGG-13**
Student Acc.	67.01	72.80	69.67	70.51	69.85	61.11
Teacher Acc.	73.82$_{+6.81}$	79.24$_{+6.44}$	79.24$_{+9.57}$	74.62$_{+4.11}$	79.23$_{+9.38}$	74.62$_{+13.51}$
FitNets	67.99$_{+0.98}$	73.79$_{+0.99}$	70.28$_{+0.61}$	72.29$_{+1.78}$	71.80$_{+1.95}$	61.42$_{+0.31}$
AT	66.42$_{-0.59}$	72.19$_{-0.61}$	71.77$_{+2.10}$	71.19$_{+0.68}$	70.78$_{+0.93}$	61.96$_{+0.85}$
VID	67.47$_{+0.46}$	73.38$_{+0.58}$	71.52$_{+1.85}$	72.22$_{+1.71}$	72.84$_{+2.99}$	63.01$_{+1.90}$
KD	68.86$_{+1.85}$	74.63$_{+1.83}$	73.46$_{+3.79}$	72.26$_{+1.75}$	72.91$_{+3.06}$	64.47$_{+3.36}$
CRD	69.71$_{+2.70}$	75.61$_{+2.81}$	73.73$_{+4.06}$	72.86$_{+2.35}$	73.65$_{+3.80}$	**66.34**$_{+5.23}$
MIMKD (ours)	**69.88**$_{+2.87}$	**76.24**$_{+3.44}$	**74.09**$_{+4.42}$	**73.88**$_{+3.37}$	**74.55**$_{+4.70}$	65.89$_{+4.78}$

Table 2.2: Observed test accuracy (in %) of student networks trained with teacher networks of higher capacity and different architecture on the CIFAR100 dataset using our method MIMKD and other distillation frameworks.

student pairs and does not impose structural constraints on intermediate layers that hinder training. While other methods that work on intermediate feature maps like AT (Zagoruyko and Komodakis 2016a) and FitNets (Romero et al. 2014) do not show much improvement from base student accuracy.

2.4.4 Transferring representations

Finally, we compare the transferability of features learned with knowledge distillation from MIMKD, on two other datasets: STL-10 and TinyImagenet. A WRN-16-2 network is trained with and without distillation from a pre-trained WRN-40-2 teacher on the CIFAR100 dataset. The student is then used as a frozen feature extractor (pre-classification layer) for images in the STL-10 and the TinyImageNet dataset. A linear

	STL-10	**TinyImageNet**
Base Accuracy (no distillation)	69.5	33.8
Knowledge Distillation (KD)	70.6	33.9
Attention Transfer (AT)	70.8	34.4
Contrastive Repr. Distill (CRD)	71.4	35.6
MIMKD (this work)	**71.8**	**36.2**

Table 2.3: Observed test-set accuracy (in %) of the student network on STL-10 and TinyImagenet datasets using our method (MIMKD) and other distillation frameworks.

classifier is trained on these extracted features to perform classification on the test sets of these datasets. The classification accuracy on the unseen datasets is interpreted as the transferability of representations. Results are presented in Table 2.3 and show that MIMKD learns more transferrable representations.

2.5 Discussion

We presented a framework (MIMKD) motivated by an information-theoretic perspective on knowledge distillation. Utilizing an information-efficient lower bound on mutual information, we proposed three information maximization formulations and demonstrated the value of region-consistent local and feature-level information maximization on distillation. We enable intermediate distillation using a JSD based lower-bound on MI which we optimize using only one negative sample.

Acknowledgments This work was supported by NSF Awards IIS-2221943 and IIS-2201710, and through gift funding from a Facebook Research Award: Towards On-Device AI.

3

CLIP-Lite: Information Efficient Visual Representation Learning with Language Supervision

We propose CLIP-Lite, an information efficient method for visual representation learning by feature alignment with textual annotations. Compared to the previously proposed CLIP model, CLIP-Lite requires only one negative image-text sample pair for every positive image-text sample during the optimization of its contrastive learning objective. We accomplish this by taking advantage of an information efficient lower-bound to maximize the mutual information between the two input modalities. This allows CLIP-Lite to be trained with significantly reduced amounts of data and batch sizes while obtaining better performance than CLIP at the same scale. We evaluate CLIP-Lite by pretraining on the COCO-Captions dataset and testing transfer learning to other datasets. CLIP-Lite obtains a +14.0% mAP absolute gain in performance on Pascal VOC classification, and a +22.1% top-1 accuracy gain on ImageNet, while being comparable or superior to other, more complex, text-supervised models. CLIP-Lite is also superior to CLIP on image and text retrieval, zero-shot classification, and visual grounding. Implementation: https://github.com/4m4n5/CLIP-Lite

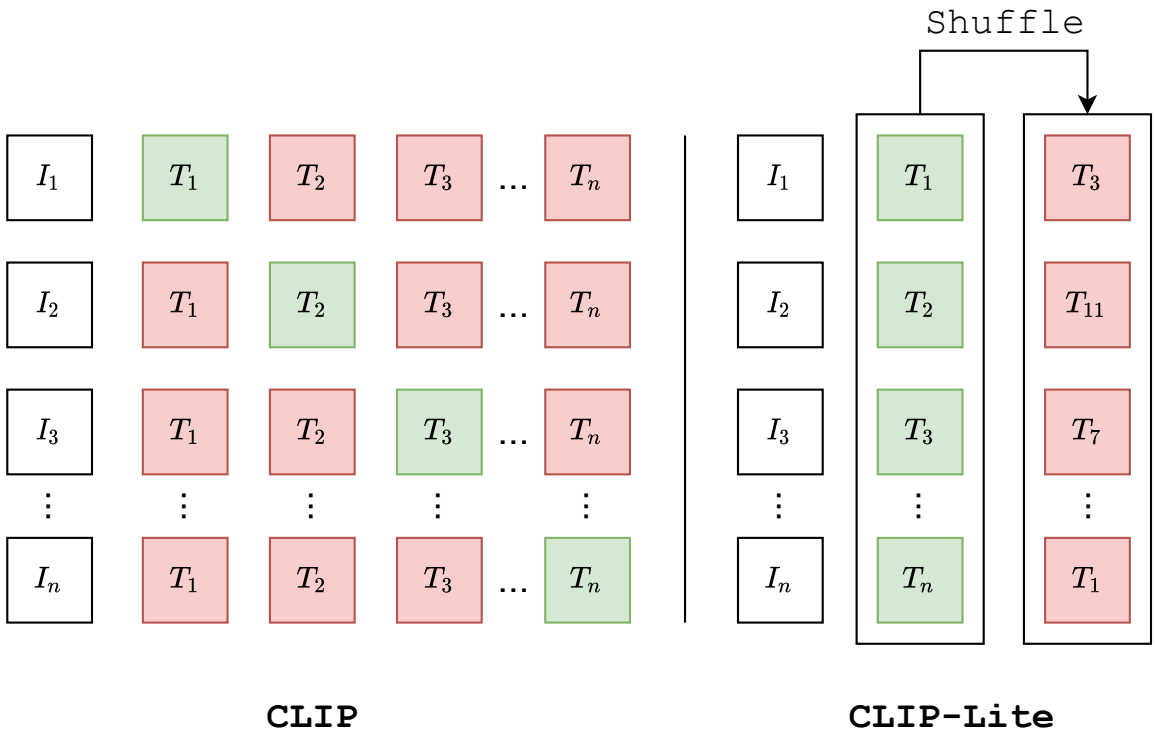


Figure 3.1: Given a batch of n image-caption pairs $\{(I_i, T_i)\}$, CLIP requires a large number of negative pairs $\{(I_i, T_j) \mid i \neq j\}$ due to the need to pair every image in the batch with captions from other images. Whereas, CLIP-Lite can learn representations using a single negative pair (in red) for every positive pair (in green).

3.1 Introduction

Pretraining image classification networks on the Imagenet dataset has led to visual representations that transfer to other tasks (Girshick et al. 2014; Long, Shelhamer, and Darrell 2015; Vinyals et al. 2015; Antol et al. 2015; Y. Zhu et al. 2016). However, such classification based pretraining requires a large amount of human-annotated data which is hard to obtain at scale. In contrast, captioned image data is an information-dense source of supervision that is relatively cheap to collect and plentiful on the internet. Therefore, recent methods have used joint vision-language pretraining to learn representations from image-caption pairs (Desai and Johnson 2021; Sariyildiz,

Perez, and Larlus 2020). However, methods such as VirTex (Desai and Johnson 2021) which train on complex language modeling tasks such as masked language modeling, token classification, and captioning fail to align features in a common latent space.

Recently, CLIP (Radford et al. 2021), a vision-language pretraining model, was developed using contrastive learning between the two modalities on an Internet-sized dataset of 400 million image-caption pairs. Contrastive learning methods work by pulling closer the representations of independent views of the same datum *i.e.* a positive or matching image-caption pair and pushing apart the representations of independent views of different data *i.e.* negative or non-matching image-caption pairs. However, contrastive learning in vision-language pretraining still has some limitations as it seems to be most effective only with large scale data, and it requires a large number of negative image-caption pairs during training. Our work aims to address and explore these two limitations by proposing CLIP-Lite, an information efficient variation of CLIP that is useful even in smaller data regimes, does not rely in as many negative sample pairs during training, and provides comparable or superior performance on standard benchmarks against other methods trained at the same scale. Our work is motivated by the observation that multiple contrastive objectives maximize a lower-bound on the mutual information between two or more views of the same datum (M. Wu et al. 2020).

3.2 Background

In this section, we discuss previous efforts in improving pretraining, and in vision language alignment which are the key areas of contribution of our work. Our work is related to several strands of research on visual pretraining without full-supervision.

Vision-Language Pretraining: Research on learning visual representations by using textual labels or annotations has a long history. In (Quattoni, Collins, and Darrell 2007), the authors learn data-efficient image representations using manifold learning in the weight space of classifiers trained to predict tokens in image captions. Following this work, (Joulin et al. 2016) used convolutional neural networks to predict words in image captions to learn image representations.

This approach was later extended in (Lei Ba, Swersky, Fidler, et al. 2015) where the model learns to predict phrase n-grams, which demonstrated impressive zero-shot performance on downstream classification tasks. Recently, VirTex (Desai and Johnson 2021) used proxy language modeling tasks, such as image-captioning to train a visual encoder and a transformer based language decoder which generates captions. ICMLM (Sariyildiz, Perez, and Larlus 2020) demonstrated a similar masked language modeling approach but relied on pretrained textual encoders for generating textual features. In (Stroud et al. 2020), video representations are learned using paired textual metadata, however the method does not extend to visual pretraining for images. In general, these methods distill the rich semantic information from a caption into the visual representation by learning to predict each token in the caption given the corresponding image. More recent work, such as CLIP (Radford et al. 2021), has shown that a simpler contrastive objective for aligning image and caption pairs is also able to learn a powerful visual representation. Our work extends CLIP using a more information-efficient approach.

Contrastive Representation Learning and Mutual Information Estimation: As demonstrated in (M. Wu et al. 2020), we observe that contrastive frameworks learn by maximizing the mutual information (MI) between different views of a given data point. For images, this is achieved by maximizing the MI between different augmenta-

tions of the data as in SimCLR (T. Chen et al. 2020; Bachman, Hjelm, and Buchwalter 2019). While for sequential data such as conversational text, consecutive utterances can be considered as different views (Stratos 2018). Similarly, several other contrastive frameworks have been proposed that learn representations in domains such as images (Grill et al. 2020; Caron et al. 2020), text (Mikolov et al. 2013; Stratos 2018), graphs (Veličković et al. 2018), and videos (Jabri, Owens, and Efros 2020). The value of mutual information is extremely challenging to estimate, especially for the high-dimensional continuous representations used in deep learning. To this end, various tractable lower-bounds on mutual information are used for optimization. Recently, MINE (Belghazi et al. 2018) proposed a general-purpose parameterized neural estimator of mutual information. It uses a Donsker-Varadhan (Donsker and Varadhan 1983) representation of KL-divergence as the lower-bound on mutual information. MINE (Belghazi et al. 2018) used a neural network critic to distinguish positive and negative pairs of samples. Another popular bound on mutual information that has seen wide adoption due to its low variance is the InfoNCE (Oord, Y. Li, and Vinyals 2018) bound. In (Hjelm et al. 2018), the infoNCE bound on the mutual information is used for unsupervised representation learning. While it is used by several other methods for self-supervised (T. Chen et al. 2020) representation learning for images. The capacity of the bound is limited by the number of contrastive samples used (McAllester and Stratos 2020). Additionally, InfoNCE can underestimate large amounts of true MI which is generally the case with high-dimensional representations of natural images. To this end, DeepInfoMax (Hjelm et al. 2018) proposed using a lower-bound on mutual information that is based on the Jensen-Shannon Divergence (JSD) instead of the traditional KL-divergence (KLD). Inspired by this, we extend the use of this bound for vision-language pretraining and demonstrate its effectiveness through extensive experimental evaluations.

3.3 Method

In this section, we describe our pretraining framework (Figure 3.2) for visual representation learning. Given a dataset of image-caption pairs, the goal of our pretraining framework is to train an image encoder and a text encoder such that representations learned from the visual and the textual streams share maximum information (Figure 3.2 shows an overview). Consider an image encoder network, f_i with parameters θ_i and a textual encoder, f_t with parameters θ_t. Let (x_i, x_t) be a sampled image-caption pair from the dataset and $f_i(x_i)$ and $f_t(x_t)$ denote the representations extracted from the networks. Based on the information bottleneck principle (Tishby and Zaslavsky 2015), the maximum mutual information (MI) predictive coding framework (Oord, Y. Li, and Vinyals 2018; Hjelm et al. 2018; McAllester and Stratos 2020) aims to learn representations that maximize the MI between inputs and representations. In recent years, several methods (T. Chen et al. 2020; He, Fan, et al. 2020; Bachman, Hjelm, and Buchwalter 2019) have used this principle to maximize MI between representations extracted from multiple views of a shared context. In the case of visual self-supervised learning, this is achieved by creating two independently-augmented copies of the same input and maximizing the MI between the respective features produced by an encoder. This framework can be extended further by considering an image x_i and its caption x_t as distinct views of the same input. This setup is motivated by the observation that image captions contain rich semantic information about images, for instance, presence of objects, location of objects, their relative spatial configurations, etc. Distilling this information into our visual representation is useful for robust representation learning (Radford et al. 2021). To this end, we formulate our objective as follows:

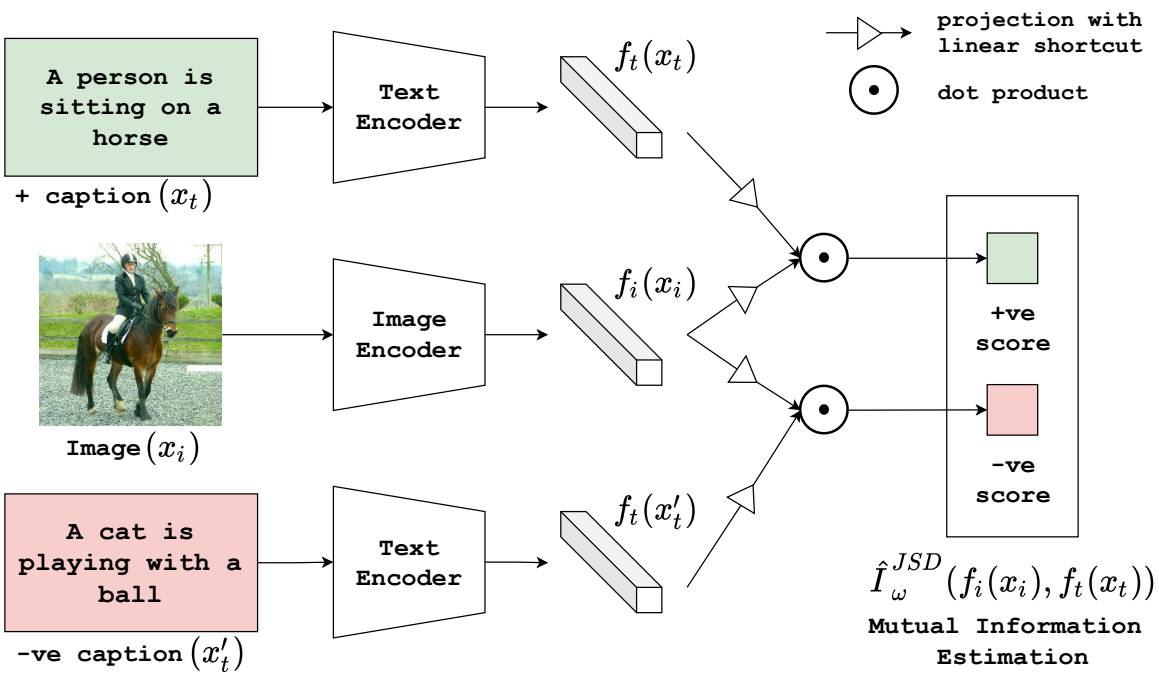


Figure 3.2: **CLIP-Lite:** We extract representations for an image, its positive caption, and one negative caption. Image-caption pairs are then fed into the mutual information discriminator function which outputs a score for each pair. These scores are then used to estimate and maximize mutual information using Jensen-Shannon Divergence (JSD) to optimize the parameters of the encoders and the mutual information discriminator end-to-end. The projection and dot function represents the MI discriminator function T_ω.

$$(\hat{\theta}_i, \hat{\theta}_t) = \text{argmax}_{\theta_i, \theta_t} \, I(f_i(x_i), f_t(x_t)), \tag{3.1}$$

where $I(f_i(x_i), f_t(x_t)) \leq I(x_i; x_t)$; due to the data processing inequality between visual and textual streams.

3.3.1 Mutual Information Maximization

For given random variables y and z, their mutual information is defined as a Kullback-Leibler (KL) divergence between their joint distribution $p(y, z)$ and the product of

their marginal distributions, $p(y)p(z)$ as,

$$I(y;z) = D_{\mathrm{KL}}(p(y,z) \,||\, p(y)p(z)). \tag{3.2}$$

However, mutual information is notoriously hard to estimate for high-dimensional continuous variables, especially when the distributions $p(y,z)$, $p(x)$, or $p(z)$ are not explicitly known. As a result, recent approaches use various tractable lower bounds on the mutual information which are differentiable and hence can be maximized with gradient-descent based optimization. For contrastive learning, a commonly used bound is infoNCE (Oord, Y. Li, and Vinyals 2018) based on Noise-Contrastive Estimation (Gutmann and Hyvärinen 2010). This bound is relatively more stable and has been shown to work in a wide variety of tasks (T. Chen et al. 2020; Bachman, Hjelm, and Buchwalter 2019; X. Chen, Fan, et al. 2020) including CLIP (Radford et al. 2021) which, similar to our method, aims to learn visual representations from textual annotations. The infoNCE bound has seen wider adoption as it demonstrates lower variance compared to the Donsker-Varadhan bound (Donsker and Varadhan 1983). However, both of these bounds require a large number of negative samples and as a result, recent methods either train with extremely large batch-sizes (Radford et al. 2021; T. Chen et al. 2020); or an additional memory-bank of negative samples (X. Chen, Fan, et al. 2020; Tian, Sun, et al. 2020).

Unlike these works, we estimate mutual information using a Jensen-Shannon Divergence (JSD) bound, similar to formulations used for generative modeling (Nowozin, Cseke, and Tomioka 2016); and source separation (Brakel and Bengio 2017). This bound on mutual information is derived by replacing the KL-divergence in equation 3.2 with the Jensen-Shannon divergence (ref. appendix for further discussion). Interestingly, the lower bound derived as such is stable, differentiable, monotonically

related to the mutual information $I(y;z)$, and most importantly, not dependent on the number of negative samples. Hence we have, $I(Y;Z) \geq \hat{I}^{JSD}_{\omega}(Y;Z)$ where,

$$\begin{aligned}\hat{I}^{JSD}_{\omega}(Y;Z) :=& \mathbb{E}_{P(Y,Z)}[-\log(1+e^{-T_\omega})] \\ &- \mathbb{E}_{P(Y)P(Z)}[\log(1+e^{T_\omega})],\end{aligned} \tag{3.3}$$

and $T_\omega : \mathcal{Y} \times \mathcal{Z} \to \mathbb{R}$ is a discriminator neural network with trainable parameters ω which are jointly optimized to distinguish between a paired-sample from a joint distribution (positive image-caption pair) and one pair from the product of marginals (negative image-caption pair). Therefore we are able to optimize our overall objective with just one negative sample as follows:

$$(\hat{\omega}, \hat{\theta}_i, \hat{\theta}_t) = \underset{\omega,\theta_i,\theta_t}{\operatorname{argmax}} \hat{I}^{JSD}_{\omega}(f_i(x_i), f_t(x_t)), \tag{3.4}$$

where the visual encoder is a convolution neural network, and features are extracted from the pre-classification layer of the network. The textual encoder is parameterized by a neural network that takes the caption as a string of textual-tokens and generates a one-dimensional representation.

3.4 Experiments

In this section, we describe the experiments that demonstrate the value of using textual captions for learning visual representations using CLIP-Lite. In our experiments, the CLIP-Lite architecture consists of a ResNet-50 image encoder and the BERT-base textual encoder and is trained on the COCO Captions (X. Chen, Fang, et al. 2015)

```
# image_encoder - CNN (eg. ResNet50)
# text_encoder - Transformer (eg. BERT)
# mi_discriminator - Project, Normalize and Dot
# I[n, h, w, c] - Batch of images
# T[n, l] - Batch of texts

# Extract image and text features
image_feats = image_encoder(I)
text_feats = text_encoder(T)

# Shuffle text features to get negative samples
text_feats_neg = shuffle(text_feats)

# Compute alignment scores using project, normalize and dot
positive_scores = mi_discriminator(image_feats, text_feats)
negative_scores = mi_discriminator(image_feats, text_feats_neg)

# MI Estimation / Loss function
loss = softplus(-1.0 * positive_score) + softplus(negative_score)
```

Figure 3.3: **CLIP-Lite:** Pytorch style pseudo-code for our pretraining framework.

dataset. We evaluate the robustness of our visual encoder through the following downstream tasks which use the visual encoder (1) as a frozen feature extractor, or (2) as source of weight initialization for finetuning (ref. appendix). In addition, we also demonstrate the data efficiency of our method by evaluating performance on fractional datasets.

3.4.1 Architecture and Training Details

In all experiments, we use a standard ResNet-50 (He, X. Zhang, et al. 2016) that takes in a 224×224 image and generates 2048-dimensional features at the pre-logit layer. For textual encoding, we use a transformer (Vaswani et al. 2017) model initialized using $\text{BERT}_{\text{base}}$ (Devlin et al. 2018) and use the output `[CLS]` token as the text representation. We use the COCO Captions dataset (X. Chen, Fang, et al. 2015) which

has 118K images with five captions per image. During training time we apply (1) random cropping, (2) color jittering, (3) random horizontal flips while interchanging the words 'left' and 'right' in the caption, and (4) normalization using the ImageNet image mean. We use SGD with momentum 0.9 (Sutskever et al. 2013; B. T. Polyak 1964) and weight decay 10^{-4} wrapped in LookAhead (M. R. Zhang et al. 2019) with $\alpha = 0.5$, and 5 steps. We perform distributed training across 8 GPUs with batch normalization (Ioffe and Szegedy 2015a) per GPU with an overall batch size of 1024 images for 250K iterations. We use linear learning rate warmup (Goyal et al. 2019) for the first 10K iterations followed by cosine decay (Loshchilov and Hutter 2016) to zero. Additionally, we train CLIP (Radford et al. 2021) on the COCO-dataset using an open-source implementation[1] with the originally recommended (Radford et al. 2021) training schedule that suit smaller datasets, reasonable batch-sizes, and compute resources. Specifically, we train using the Adam Optimizer (Kingma and Ba 2014a) with decoupled weight decay regularization (Loshchilov and Hutter 2016) for all weights except gains or biases. We train with a batch-size of 1024 and warm-up to an initial learning rate of 10^{-4} in 10K steps and decay to zero with the cosine schedule. We found that the performance slightly improves with longer training therefore we train for 250K iterations, similar to ours. All other training details and hyperparameters were kept the same as the original work (Radford et al. 2021). Please note that our ResNet-50 based CLIP-COCO model outperforms (+1.2% Zero-shot Acc. on CIFAR10) publicly available weights[2], refer to appendix for further details on CLIP-COCO training.

[1] https://github.com/mlfoundations/open_clip

[2] https://github.com/revantteotia/clip-training/blob/main/zero_shot_eval_output/coco_trained_clip_observations.md

3.4.2 Mutual Information Discriminator

As described in main paper, our JSD-based lower-bound on mutual information relies on a discriminator function, $T_\omega : \mathcal{Y} \times \mathcal{Z} \to \mathbb{R}$, which distinguishes between samples extracted from the joint distribution, $P(Y, Z)$ i.e. a positive image-caption pair and the product of marginals, $P(Y)P(Z)$ i.e. a negative image-caption pair. This discriminator function can be modelled as an arbitrary neural network with parameters ω that can be jointly optimized with the encoders during training (Belghazi et al. 2018). In this work, we use a projection and alignment based architecture similar to the one presented in Deep InfoMax (Hjelm et al. 2018).

Given a pair of one-dimensional representations, both vectors are first projected using a projection module with two linear layers separated by a ReLU and a linear shortcut. A dot-product of these projections is computed to get alignment scores. The projection function maps these representations to an aligned cross-modal latent space. Separate projection functions are used for image and text representations. Positive and negative pairs of image-text representations are passed through the discriminator to get respective scores which are then used to estimate and maximize mutual information using our objective. This architecture, in addition to being simple and computationally inexpensive, also offers alignment of the representations into a common cross-modal latent space which uses cosine similarity as the distance metric.

3.4.3 Transfer Learning with Frozen Backbone

In these experiments, we train linear models on frozen visual backbones pretrained using CLIP-Lite and compare with pretraining methods on PASCAL VOC (Everingham et al. 2010) and ImageNet-1k (Russakovsky et al. 2015) classification problems.

Table 3.1: **Frozen Backbone Results:** On Pascal VOC07 and Imagenet-1k classification, CLIP-Lite outperforms baseline CLIP when evaluated using linear classifiers trained on top of frozen backbone networks pretrained on the COCO Dataset. CLIP-Lite's performance is competitive with more complex vision-language models. CLIP-Lite also performs better than supervised and self-supervised models trained on COCO images, without captions (ref. supplemental materials for additional results).

Method	**# images**	**Annotations**	**VOC07**	**IN-1k**
COCO-Sup.	118K	labels	86.2	46.4
MoCo-COCO	118K	self-sup.	67.5	46.5
ICMLM	118K	captions	87.5	47.9
VirTex	118K	captions	**88.7**	53.8
CLIP-COCO	118K	captions	74.2	33.2
CLIP-Lite	118K	captions	88.2	**55.3**

PASCAL VOC linear classification: For this experiment, our setup is identical to VirTex (Desai and Johnson 2021). We train on VOC07 trainval split (9K images, 20 classes) and report mAP on the test split. For classification, we train per-class SVMs on 2048-dimensional global average pooled features extracted from the last layer of our trained visual encoder. For each class, we train SVMs for cost values $C \in \{0.01, 0.1, 1, 10\}$ and select best C by 3-fold cross-validation.

Imagenet-1k linear classification: For this experiment, our setup is identical to VirTex (Desai and Johnson 2021). We train on the ILSVRC 2012 train split and report top-1 accuracy on val split. We train a linear classifier (fully connected layer + softmax) on 2048-dimensional global average pooled features extracted from the last layer of the visual backbone. For training, we use a batch-size of 256 for 100 epochs. We use SGD with momentum 0.9 and weight decay 0. The learning rate schedule is decayed by 0.1 after 60 & 80 epochs with an initial LR of 30.

Results: We compare CLIP-Lite to supervised, self- and textually-supervised models

Table 3.2: **Data Efficiency:** CLIP-Lite is more data efficient than CLIP, as shown in this experiment where we pretrain on $\{25, 50, 75, 100\}\%$ of the COCO Captions dataset and evaluate the models on VOC and ImageNet classification tasks with a frozen backbone. CLIP-Lite trained with just 25% of COCO already surpasses CLIP trained on the whole dataset.

	# images	**VOC07**	**IN-1k**
CLIP COCO-100%	118K	74.2	33.2
CLIP-Lite COCO-25%	29.5K	$77.7_{+3.5}$	$45.1_{+11.9}$
CLIP-Lite COCO-50%	59K	$84.4_{+10.2}$	$51.3_{+18.1}$
CLIP-Lite COCO-75%	88.5K	$86.8_{+12.6}$	$53.2_{+20.0}$
CLIP-Lite COCO-100%	118K	$88.2_{+14.0}$	$55.3_{+22.1}$

in Table 3.1. CLIP-Lite significantly outperforms baseline CLIP when trained with the same amount of data on both tasks. When compared to other image-caption pretraining methods, CLIP-Lite performs competitively with VirTex (Desai and Johnson 2021) on VOC2007 and outperforms both VirTex (Desai and Johnson 2021) and ICMLM (Sariyildiz, Perez, and Larlus 2020), which are trained on relatively complex language modeling tasks, on Imagenet classification. In addition, different from them, our method also generates a shared latent space that encodes both image and text modalities and enables cheap computation of cross-modal alignment, which enables downstream tasks such as zero-shot retrieval, and zero-shot transfer. It also allows us to find subspaces associated with abstract concepts that are better expressed with language than with visual examples, which allows for applications in bias mitigation through the synthesis of gender-neutral image representations. CLIP-Lite also outperforms a fully-supervised model trained with COCO image labels, showing that it learns a better visual representation from information-dense captions as compared to training with labels alone. Additional results in the supplement show that CLIP-Lite is comparable or better than image-only SSL learning models trained on ImageNet, even though it is trained on much fewer images, albeit with textual supervision.

Data Efficiency: Due to our information-efficient approach for mutual information maximization, CLIP-Lite should be able to learn effective feature representations without requiring as much pretraining data as CLIP. To evaluate this claim, we train ResNet-50 backbones with our pretraining setup on multiple fractional subsets of the COCO Captions dataset and measure their downstream performance on both VOC and ImageNet classification tasks. As demonstrated in Table 3.2, CLIP-Lite outperforms the original CLIP training objective on VOC with 20% and on Imagenet with just 10% of the data, while obtaining a substantial improvement when both are trained with 100% data. Additionally, when compared with Virtex, CLIP-Lite performs competitively on VOC while being consistently better on Imagenet-1k.

3.4.4 Transfer Learning with Backbone Finetuning

Next, we evaluate the performance of of our visual backbone when the entire network is finetuned for the downstream task. For this purpose, we perform fine-grained classification on the iNaturalist 2018 (Van Horn et al. 2018) dataset, which contains images from $8,142$ fine-grained categories, with a long-tailed distribution. We train with the 'train2018' split and evaluate in the 'val2018' split. We finetune pretrained ResNet-50 models with a linear layer, using SGD with momentum 0.9 and weight decay 10^{-4} for 100 epochs. Initial learning rate is set to 0.025, which is reduced by $10\times$ at epochs 70 and 90. We use a batch size of 256 distributed across 8 GPUs.

Results: We summarize our results in Table 3.3. CLIP-Lite is competitive with supervised and self-supervised learning models trained with images alone even those trained with 5-10x more images. Its performance matches closely a model trained with full-supervision on 50% of the ImageNet (Krizhevsky, Sutskever, and G. E. Hinton

Table 3.3: **Backbone Finetuning Results:** CLIP-Lite outperforms CLIP-COCO on iNaturalist, and performs comparably to VirTex. (IN-Sup. = ImageNet-supervised.)

Method	**# images**	**Annotations**	**iNat 18**
Random Init	-	-	61.4
IN-sup	1.28M	labels	65.2
IN-sup-50%	640K	labels	63.2
IN-sup-10%	128K	labels	60.2
MoCo-COCO	118K	self-sup.	60.5
MoCo-IN	1.28M	self-sup.	63.2
VirTex	118K	captions	63.4
CLIP-COCO	118K	captions	61.8
CLIP-Lite	118K	captions	63.1

Table 3.4: **Text Retrieval Results:** CLIP-Lite substantially outperforms CLIP-COCO and the baseline Visual N-grams (A. Li et al. 2017) approach. CLIP-Lite is superior when evaluated on the COCO test split, which is similar to the CLIP-Lite training set and on Flickr30K, generalizing to unseen images and text zero-shot.

	Flickr30k			**MSCOCO**		
Method	*R@1*	*R@5*	*R@10*	*R@1*	*R@5*	*R@10*
Visual N-Grams	15.4	35.7	45.1	8.7	23.1	33.3
CLIP-COCO	19.9	41.9	54.9	18.9	42.9	54.6
CLIP-Lite	**28.8**	**55.8**	**67.4**	**26.0**	**54.6**	**68.0**

2012) dataset, equal to 5.4× the number of images as our pretraining dataset. Finally, CLIP-Lite obtains a 1.3% improvement over CLIP-COCO, while being competitive with VirTex.

3.4.5 Image-Text and Text-Image Retrieval

Our method is expected to produce effective representations for the task of image-text retrieval as it is trained by aligning text and image representations. We evaluate

Table 3.5: **Image Retrieval Results:** CLIP-Lite substantially outperforms CLIP-COCO and the baseline Visual N-grams (A. Li et al. 2017) approach. CLIP-Lite is superior when evaluated on the COCO test split, which is similar to the CLIP-Lite training set and on Flickr30K, generalizing to unseen images and text zero-shot.

	Flickr30k			**MSCOCO**		
Method	*R@1*	*R@5*	*R@10*	*R@1*	*R@5*	*R@10*
Visual N-Grams	8.8	21.2	29.9	5.0	14.5	21.9
CLIP-COCO	13.9	33.0	43.8	13.9	33.5	44.2
CLIP-Lite	**23.1**	**51.1**	**62.9**	**20.2**	**48.1**	**62.2**

the image-text retrieval capabilities of CLIP-Lite on the validation set of COCO and the test split of Flickr30k (Young et al. 2014) datasets, following CLIP. We perform zero-shot image-text and text-image retrieval by ranking image-text pairs by their alignment score, which is the dot product of the normalized representations in the shared latent space. This ability to perform zero-shot retrieval is a salient feature of our and CLIP-like methods over works that rely on language modeling tasks.

Results: Table 3.4, 3.5 shows that CLIP-Lite substantially outperforms CLIP-COCO on all metrics for both text and image retrieval. The performance improvement is large both when evaluated on the COCO validation set, which is similar to the the COCO-Captions training split used for CLIP-Lite training; and when testing zero-shot on unseen text vocabulary and object categories of Flickr30K. Taken together, these results show that CLIP-Lite learns a superior representation for retrieval tasks as compared to CLIP, when trained on same amounts of data.

3.4.6 Zero-Shot Transfer

We use the cross-modal alignment capability of CLIP-Lite to perform zero-shot classification on unseen datasets CIFAR-10, CIFAR100 (Krizhevsky, G. Hinton, et al.

Table 3.6: **Zero Shot Transfer:** CLIP-Lite obtains satisfactory zero-shot transfer to unseen datasets.

	CLIP-COCO		**CLIP-Lite**	
Dataset	*Top1*	*Top5*	*Top1*	*Top5*
CIFAR10	16.3	68.9	**33.0**	**82.7**
CIFAR100	2.9	12.4	**6.8**	**33.1**
ImageNet-V2	4.4	11.1	**9.9**	**21.4**
ImageNet-A	1.7	7.3	**3.8**	**14.9**

2009), ImageNetV2 (Recht et al. 2019), and ImageNet-A (Hendrycks et al. 2021). Our model generates a shared latent space where we can readily compute the alignment between (image, text) pairs as the cosine of their representations. Therefore, we use the names of the classes to generate a textual description of each class label (class prompt). In this experiment, we use templates such as, "a photo of a {class name}" to generate such class prompts, following CLIP (Radford et al. 2021). Please refer to the appendix for comparison between different templates for generating the prompts. For a given image, we compute its alignment with each of the class prompts which are then normalized into a probability distribution via a softmax.

Results: Our results for the zero-shot transfer task on unseen datasets are compiled in table 3.6. Given the zero-shot nature of the task, CLIP-Lite obtains satisfactory performance on the complex ImageNet evaluations while clearly outperforming CLIP trained with the same amount of data in all settings.

3.4.7 Evaluating Visual Grounding

Next, we evaluate the capability of CLIP-Lite to localize a region in the image that corresponds to a given textual description. We compute the dot-product of the visual

and textual embedding and compute its gradients with respect to the last convolutional layer of ResNet. We global average pool these gradients and perform a weighted sum with the last convolutional activations and clip the negative values to obtain Grad-CAM (Selvaraju et al. 2017). We then use the areas highlighted by Grad-CAM to approximate a predicted bounding box. We evaluate this experiment on the RefCOCO+ (Yu et al. 2016) dataset. We note that the images in the RefCOCO+ dataset are extracted from the training set of the COCO (X. Chen, Fang, et al. 2015) dataset which our model uses for pretraining. Therefore, we view this evaluation as an explorative study to establish that our model is focusing on the relevant areas of the image while computing the alignment score with the caption. RefCOCO+ results can be seen in the table to the right. CLIP-Lite significantly outperforms CLIP on all settings.

Method	Val-acc	TestA-acc	TestB-acc
CLIP-COCO	29.1	28.5	28.5
CLIP-Lite (ours)	**36.1**	**41.4**	**32.0**

Qualitative results in Figure 3.4 demonstrate that even though the network has not been trained with any localization supervision, it is surprisingly good at localizing phrases in the image. For instance, in Figure 3.4 bottom left, for the phrase "blue", the network attends to all blue regions in the player's outfit. Interestingly, it is also able to localize abstract concepts as "blurry player".

Figure 3.4: **Visual Grounding on RefCOCO+:** CLIP-Lite is able to localize textual descriptions to relevant areas in the image, shown here through Grad-CAM visualization using the alignment score with the mentioned textual description. *Top left:* CLIP-Lite is able to localize the action phrases such as “bending over”. This demonstrates the value of learning from semantically rich textual captions.

3.4.8 Editing Concepts from Image Representations

One salient feature of CLIP-like methods, which other methods such as VirTex (Desai and Johnson 2021) and ICMLM (Sariyildiz, Perez, and Larlus 2020) lack, is that they are able to generate a shared latent space that encodes both image and text modalities. This enables us to find representations and subspaces associated with abstract concepts that are better expressed with language than with visual examples. Using this property, we demonstrate a methodology to remove concepts from visual representations. For instance, it is non trivial and even problematic to collect visual examples that capture the concept of gender, while it is relatively straightforward to express this concept in a sentence using language. Therefore, we can identify the gender subspace in our shared embedding space using text and use it to remove variance along this direction to smooth out the concept of gender from image representations. We motivate this experiment in the growing body of literature regarding

Figure 3.5: **Demonstrating Neutral Representations:** Qualitative demonstration of our concept editing method. For each text prompt, most aligned images are retrieved from male and female buckets of the gendered COCO subset before (top row) and after (bottom row) gender smoothing. Once representations are gender-neutralized the gendered references in the query become irrelevant and the image is only retrieved based on its remaining contents. Alignment score decreases from left to right for each set of queried images. Boundary color denotes perceived image gender; red for female, blue for male.

bias mitigation, where the objective is to build invariant representations with respect to sensitive or protected attributes (T. Wang et al. 2019; Zeyu Wang et al. 2020). In comparison to our work other methods require retraining the models to obtain invariant bias representations through adversarial learning (T. Wang et al. 2019) or effectively combining domain independent classifiers (Zeyu Wang et al. 2020).

Identifying the Concept Subspace: The first step of our approach is to isolate the direction in the embedding space that captures maximum gender variance. For this purpose, we follow a strategy similar to Bolukbasi et al. (Bolukbasi et al. 2016) that deals with debiasing word representations. For characterizing features for male and female genders, we use word pairs (*man, woman*), (*son, daughter*) that indicate opposite genders. Now, consider a dataset $\mathcal{D} = \{(w_m, w_f)\}_{i=1}^{m}$ where each entry (w_m, w_f) is a tuple of opposite gendered words. Intuitively, each tuple should contain words that have the same meaning if not for the target attribute. To make the set $\mathcal{D}$ more robust, we used the sentence contextualization strategy presented in Liang

Table 3.7: **Concept Editing Results:** We compute the mean alignment scores for the top 10 images queried using prompts that either contain male or female gendered tokens. The images are queried using gendered and neutralized representations. We observe that after gender-deletion the alignment score for images with men and women converge to similar values.

	Images with Men			**Images with Women**		
	gendered	neutral	delta	gendered	neutral	delta
Male queries	0.085	0.069	+0.016	0.057	0.067	-0.010
Female queries	0.042	0.068	-0.026	0.089	0.062	+0.027

et al. (Liang et al. 2020). In this step, the predefined sets of gendered tokens in the set, $\mathcal{D}$, are used to generate paired sentences which have the same meaning except for the gender attribute. We perform this contextualization by using simple sentence templates such as "I am a [word]" where [word] can be replaced with the word pairs in our dataset $\mathcal{D}$ to give, for instance, (*"I am a boy.", "I am a girl."*). Hence, we obtain a contextualized bias attribute dataset $\mathcal{S} = \{(s_m, s_f)\}_{i=1}^{n}$ where each entry is a tuple of semantically similar sentences with opposite genders. We extract the sentence representations for all entries in the set $\mathcal{S}$ by passing them through our pretrained text encoder and then projecting them to the shared latent space using the projector trained with our mutual information discriminator T_ω. We define sets $\mathcal{R}_m$ and $\mathcal{R}_f$ that contain sentence representations of the male and the female category, for example, $\mathcal{R}_m = \{F_t(s_m)\}_{i=1}^{n}$ where $F_t(.)$ is the sequential combination of our pretrained text-encoder and text-projection functions. Now we estimate the gender subspace $V = \{v_1, ..., v_k\}$ using the Principal Component Analysis corresponding mean shifted representation from both sets as described in (Liang et al. 2020).

Removing Concept from Image Representations: After estimating the gender subspace in our shared cross-modal latent space, we extend the hard debias algorithm (Bolukbasi et al. 2016) to edit visual representations. This is achieved by first

projecting the representation onto the bias subspace, this projection is then subtracted from the original representation to give the de-gendered representation. Given an image, we first encode the image onto our multi-modal shared latent space to get, say, h. Now, consider the identified gender subspace V, we first compute the projection of h onto this gender subspace V to get $h_V = \sum_{j=1}^{k} \langle h, v_j \rangle v_j$. We subtract this projection from the original representation to get a vector, $\hat{h} = h - h_V$ that is orthogonal to the bias subspace and therefore does not encode the target bias.

Analysis: To evaluate concept editing, we use the gendered subset of COCO-Captions (T. Wang et al. 2019; Zhao et al. 2017) for studying bias. The gender labels for images in the COCO dataset are derived from the captions. We obtain a subset from the COCO dataset with $16,225$ images with men and $6,601$ images with women. We use 10 sentences with male references and 10 sentences with female references from the set $\mathcal{S}$ and use them as prompts for this study. For each gendered prompt, we query the top 10 images independently from the male and the female image sets using both biased and debiased representations to compute alignment with the prompt. The mean alignment scores are then computed for each set given the prompt. Table 3.7 shows that the alignment scores roughly equalize for members of the two groups after removing the variance along the gender direction from the visual representations which indicates the invariance of the visual representations to gendered language tokens.

3.5 Discussion

We introduced CLIP-Lite an image-text pretrained model using contrastive learning that leverages a different objective than the CLIP model that allows for it to be more data efficient. CLIP-Lite's objective is insensitive to the number of nega-

tive samples and hence can be trained with just one negative image-caption pair and shows superior results on lower data regimes while still demonstrating some of the most remarkable capabilities of the original CLIP model such as transferable features, zero-shot capabilities, and a shared latent space. Additionally, we present a concept editing methodology for neutralizing visual representations with respect to a chosen abstract concept. As a followup to the above work, we propose designing a training paradigm for medical image-text data which can be trained cheaply and small amounts of annotated data.

Acknowledgments This work was supported by NSF Awards IIS-2221943 and IIS-2201710, and through gift funding from a Salesforce AI Research Grant.

4

SAASN: Self-Attentive Adversarial Stain Normalization

Hematoxylin and Eosin (H&E) stained Whole Slide Images (WSIs) are utilized for biopsy visualization-based diagnostic and prognostic assessment of diseases. Variation in the H&E staining process across different lab sites can lead to significant variations in biopsy image appearance. These variations introduce an undesirable bias when the slides are examined by pathologists or used for training deep learning models. To reduce this bias, slides need to be translated to a common domain of stain appearance before analysis. We propose a Self-Attentive Adversarial Stain Normalization (SAASN) approach for the normalization of multiple stain appearances to a common domain. This unsupervised generative adversarial approach includes self-attention mechanism for synthesizing images with finer detail while preserving the structural consistency of the biopsy features during translation. SAASN demonstrates consistent and superior performance compared to other popular stain normalization techniques on H&E stained duodenal biopsy image data. Implementation: https://github.com/4m4n5/saasn-stain-normalization

4.1 Introduction

Histopathology involves staining patient biopsies for microscopic inspection to identify visual evidence of diseases. The most widely used stain in histopathology is the Hematoxylin and Eosin (H&E) stain (A. H. Fischer et al. 2008). Hematoxylin has a deep blue-purple color and stains acidic structures such as nucleic acids (DNA in cell nuclei). While Eosin is red-pink, and stains basic structures such as nonspecific proteins in the cytoplasm and the stromal matrix. Staining is crucial as it enables visualization of the microscopic structural features in the biopsy. The process of staining is followed by glass biopsy slide creation and eventually digitization into Whole Slide Images (WSIs) using digital scanners. These WSIs are further used for histopathology research and electronic transmission of biopsies.

Computer vision is becoming increasingly useful in the field of histology for computed-aided diagnosis and discovering information about histopathological microscopic cellular (Litjens et al. 2017). Tremendous potential has been shown for training deep learning algorithms on these datasets for diagnosis and visual understanding of diseases requiring histopathological assessment. Convolution Neural Networks (CNNs) have been successfully reported for biopsy-based diagnosis of breast cancer and enteropathies among others (Y. Liu et al. 2017; Aman Shrivastava, Kant, et al. 2019; Wei et al. 2019). The performance and fairness of such data-driven methods is dependent on the data used for training. Therefore, it is imperative for the training data to be free of any bias that might skew the models. A common source of such bias is significant stain color variation among images. This is due to the discrepancies in the manufacturing protocol and the raw materials of the staining chemicals (Bejnordi et al. 2014) across different sites where the biopsy slides are prepared. Multiple H&E stain distributions within the CNN input data can lead to biased predictions where

the results are influenced by color differences rather than microscopic cellular features of interest for clinical diagnostic interpretation. Additionally, it causes difficulty for a trained model to make predictions on a biopsy WSI with a new stain appearance that is not represented in the data used to train the model.

To overcome these issues, researchers have developed stain normalization techniques to convert all input images to an equivalent color distribution. Some of the most popular stain normalization techniques depend on a qualitatively chosen target image that represents an ideal color appearance (Macenko et al. 2009; A. M. Khan et al. 2014; Vahadane et al. 2016). The input (source) image is normalized to match the stain profile of the chosen target image. The obvious downside to this approach is that the normalization is highly dependent on the color distribution of a single image. Rather than using just one target image to represent an entire stain distribution, an alternative approach to consider an entire set of images that share the same stain distribution as the target domain has been suggested (Janowczyk, Basavanhally, and Madabhushi 2017; Shaban et al. 2019). A mapping function can then be learned to translate images from a particular source domain to a target domain. This problem can be modelled as an unsupervised image-to-image translation task (M.-Y. Liu, Breuel, and Kautz 2017).

Recently, Generative Adversarial Networks (GANs) have been shown to demonstrate exceptional results in unpaired image translation tasks (Yi et al. 2017; J.-Y. Zhu et al. 2017; T. Kim et al. 2017). However, the challenge posed by the stain normalization task is to ensure the preservation of fine details and microscopic structural properties that are crucial for the correct disease assessment. Additionally, since the biopsy slides can be sourced from multiple sites, the framework needs to be capable of mapping multiple stain distributions to a common target distribution.

In this paper, we propose a novel adversarial approach that can execute *many-to-one* domain stain normalization. A custom loss function, structural cycle-consistency loss, is designed to make sure that the structure of the image is preserved during translation. Self-attention (Parikh et al. 2016) is used to ensure that highly detailed microscopic features can be synthesized in the image. Our approach and other leading stain normalization techniques are compared on duodenum biopsy image data that was used to diagnose Celiac or Environmental Enteropathy disease in children. SAASN demonstrated superior performance in preserving the structural integrity of images while transferring the stain distribution from one domain to the other.

4.2 Background

The earliest methods that attempted stain normalization were primarily simple style transfer techniques. Histogram specification mapped the histogram statistics of the target image with the histogram statistics of the source (Coltuc, Bolon, and Chassery 2006). This approach only works well if the target and source images have similar color distributions. Forcing the normalization of the source image to match the histogram statistics of the target can create artifacts which can alter the structural integrity. As demonstrated by Reinhard et al. (Reinhard et al. 2001), color transfer with histogram specification can also be performed in a decorrelated CIELAB color space which is designed to approximate the human visual system.

For H&E stained histology images, the presence of each stain or the lack thereof at each pixel should represent the most appropriate color space. Considering this, researchers developed stain normalization methods that outperformed the histogram specification technique by leveraging stain separation. These techniques start with

converting an RGB image into Optical Density (O_D) as $O_D = log\frac{I_0}{I}$, where I_0 is the total possible illumination intensity of the image and I is the RGB image. Color Deconvolution (CD) is made easier in the OD space, because the stains now have a linear relationship with the OD values. The CD is typically expressed as $O_D = VS$, where V is the matrix of stain vectors and S is the stain density map. The stain density map can preserve the cell structures of the source image, while the stain vectors are updated to reflect the stain colors of the target image.

In Macenko et al. (Macenko et al. 2009), stain separation is computed using singular value decomposition on the OD tuples. Planes are created from the two largest singular values to represent H&E stains. One useful assumption with this approach is that the color appearance matrix is non-negative, this is because a stain value of zero would refer to the stain not being present at all. The approach by Vahadane et al. (Vahadane et al. 2016) (Vahadane) also includes the non-negative assumption, as well as, a sparsity assumptions, which states that each pixel is characterized by an effective stain that relates to a particular cell structure (nuclei cells, cytoplasm, etc.). Stain separation is generated with Sparse Non-negative Matrix Factorization (SNMF) where the sparsity acts as a constraint to greatly reduce the solution space (Roy et al. 2018). SNMF is calculated using dictionary learning via the SPAMS package.

While Macenko and Vahadane are both unsupervised techniques, supervised approaches to this problem have also been studied. Khan et al. (A. M. Khan et al. 2014) applies a relevance vector machine or a random forest model to classify each pixel as hematoxylin, eosin or background. The authors provide a pre-trained model for cases which is only useful if the color distribution of new source images is close to the color distribution of their training data. Training a new model would require a training set with pixel level annotations for each stain. After the stain separation, the

color of the target image is mapped with a non-linear spline. The non-linear mapping approach can lead to undesirable artifacts and this normalization approach is more computationally costly than the unsupervised approaches.

Recently, techniques for stain normalization have progressed to include deep learning approaches such as autoencoders and GANs (Janowczyk, Basavanhally, and Madabhushi 2017; Shaban et al. 2019). The StainGAN (Shaban et al. 2019) approach applied the CycleGAN framework for *one-to-one* domain stain transfers. In a *one-to-one* stain transfer situation, the cycle-consistency loss is calculated by taking the $L1$ distance between the cycled image and the ground truth. In a *many-to-one* situation, the cycled image will likely have a different color appearance than the original image. Therefore, a new loss function that focuses on image structure and not the color differences is required.

Biopsy WSIs contain repetitive patterns across the image in the form of recurring cell structures, stain gradients, and background alike. During translation, these spatial dependencies can be used to synthesize realistic images with finer details. Self-attention (Parikh et al. 2016) exhibits impressive capability in modelling long-range dependencies in images. SAGAN (H. Zhang et al. 2018) demonstrated the use of self-attention mechanism into convolutional GANs to synthesize images in a class conditional image generation task. We incorporate these advances in SAASN to enable it to efficiently find spatial dependencies in different areas of the image.

4.3 Method

The general objective of the proposed framework is to learn the mapping between stain distributions represented by domains X and Y. Since the aim of the approach

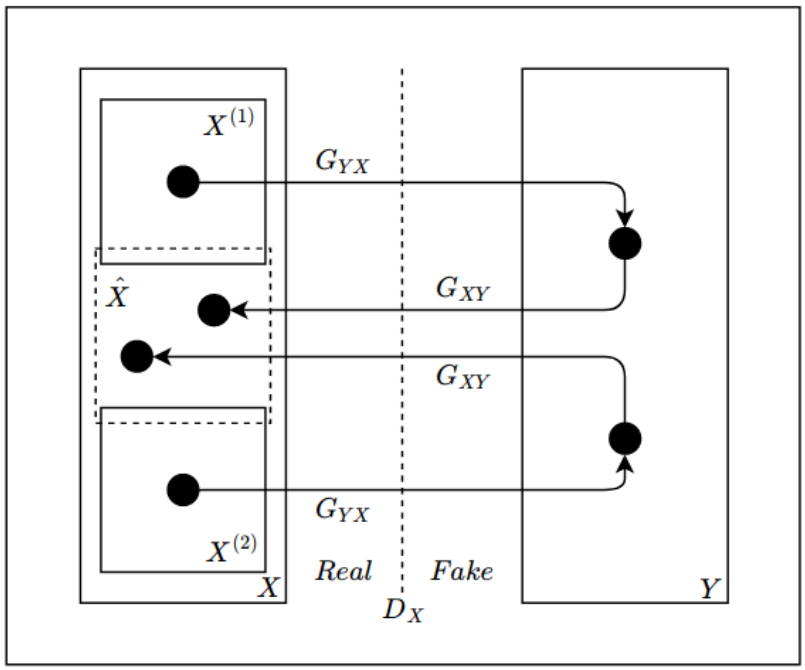


Figure 4.1: Visual example of a *many-to-one* stain transfer network. Two different stains are present as inputs within X: $X^{(1)}$ and $X^{(2)}$. Both of these domains are translated to Y with G_{XY}. To complete the cycle, G_{YX} returns the image back to the X domain, but it can no longer be mapped directly to the input sub-domains $X^{(1)}$ or $X^{(2)}$ from which it originated. Instead, the image is mapped back to $\hat{X}$ which is represents a new domain of stain appearance.

is to normalize stain patterns across the entire dataset, one of these domains can be considered as the target domain (say Y). The task is then to generate images that are indistinguishable the target domain images based on stain differences. The stain normalization task desires translation of images to a singular domain of stain distribution. This allows us to have multiple sub-domains in domain X representing different stain patterns. The overall objective then becomes to learn mapping functions $G_{YX} : X \rightarrow Y$ and $G_{XY} : Y \rightarrow X$ given unpaired training samples $\{x_i^k\}_{i=1}^N$, $x_i^k \in X^{(k)} \in X$, $k \in [1, K]$ where K denotes the number of sub-domains in X and $\{y_j\}_{j=1}^M$, $y_j \in Y$. The distribution of the training dataset is denoted as $x \sim p\,(x \mid k)$ and $y \sim p\,(y)$. Additionally, two discriminator functions D_X and D_Y are used. D_X is

employed to distinguish mapped images $G_{XY}(y_i)$ from x_i while in a similar fashion D_Y is used to distinguish $G_{YX}(x_i)$ from y_i. As illustrated in Figure 4.1, the mapping function G_{XY} will map images from domain Y to a previously undefined sub-domain $\hat{X}$ whose boundary is defined by the optimization function and the training data distributions in domain X. The overall optimization function used to train the designed framework includes a combination of *adversarial loss* (Goodfellow et al. 2014), *cycle consistency loss* (J.-Y. Zhu et al. 2017), *identity loss* (Taigman, A. Polyak, and Wolf 2016), *structural cycle consistency loss* based on the *structural similarity index* (Zhou Wang et al. 2004) and a *discriminator boundary control* factor.

Adversarial loss is used to ensure that the stain distribution of the generated images matches the distribution of the real (ground truth) images in that domain. The objective for the mapping function $G_{YX} : X \rightarrow Y$ and the corresponding discriminator D_Y is defined as:

$$\mathcal{L}_{adv}^{Y} = \mathbb{E}_{y \sim p_{(y)}}[\log D_Y(y)] + \mathbb{E}_{x \sim p_{(x|k)}}[\log(1 - D_Y(G_{YX}(x)))] \tag{4.1}$$

Here G_{YX} tries to generate images that are indistinguishable from images in domain Y and consequently fool the discriminator D_Y, i.e. the generator G_{YX} tries to minimize the given objective function while the discriminator D_Y tries to maximize it. Similarly the objective for the reverse mapping function $G_{XY} : Y \rightarrow X$ is defined. The presence of multiple distinct stain distributions in the domain X can make it challenging for the discriminator D_X to learn the decision boundary surrounding the domain X. This can especially pose a challenge when there is an overlap or proximity in the stain distribution of one of the sub-domains of X and the target domain Y in the high-dimensional space. Therefore, to make sure that the decision boundary learned by D_X does not include sections of the target domain Y, a **discriminator boundary**

control factor is added to the optimization function as follows:

$$\mathcal{L}_{adv}^{X} = \mathbb{E}_{x\sim p_{(x|k)}}[\log D_X(x)] + \mathbb{E}_{y\sim p_{(y)}}[\log(1 - D_X(G_{XY}(y)))] + \mathbb{E}_{y\sim p_{(y)}}[\log(1 - D_X(y))] \tag{4.2}$$

Cycle consistency loss (J.-Y. Zhu et al. 2017) is implemented to reconcile with the unpaired nature of the task. To overcome the lack of a ground truth image for a fake image generated in a particular domain, the image is mapped back to its original domain using the reverse mapping function. The reconstructed image is then compared to the original source image to optimize the mapping function as follows:

$$\mathcal{L}_{cyc} = \mathbb{E}_{x\sim p(x|k)}\left[\|G_{XY}(G_{YX}(x)) - x\|_1\right] + \mathbb{E}_{y\sim p(y)}\left[\|G_{YX}(G_{XY}(y)) - y\|_1\right] \tag{4.3}$$

Structural cycle consistency loss is added to the objective function to alleviate the shortcomings of the cycle consistency loss for *many-to-one* translation. In a *many-to-one* situation, the cycled images are likely to have a distinct color distribution than any of the sub-domains. Therefore minimizing the $L1$ distance between original and the cycled image alone is not an effective way to ensure cycle consistency. We use a color agnostic structural dissimilarity loss based on the Structural Similarity (SSIM) index (Zhou Wang et al. 2004) as follows:

$$\mathcal{L}_{scyc} = \frac{1 - SSIM\left(G_{XY}(G_{YX}(x)), x\right)}{2} + \frac{1 - SSIM\left(G_{YX}(G_{XY}(y)), y\right)}{2} \tag{4.4}$$

Additionally, to ensure that the the mapping learnt by the generator does not result in the loss of biological artifacts, the structural dissimilarity loss is also computed

between the mapped and the original image:

$$\mathcal{L}_{dssim} = \frac{(1 - SSIM\left(G_{YX}\left(x\right), x\right))}{2} + \frac{(1 - SSIM\left(G_{XY}\left(y\right), y\right))}{2} \tag{4.5}$$

where

$$SSIM(a, b) = \frac{(2\mu_a\mu_b + C_1) + (2\sigma_{ab} + C_2)}{(\mu_a^2 + \mu_b^2 + C_1)(\sigma_a^2 + \sigma_b^2 + C_2)} \tag{4.6}$$

where μ, σ are the respective means and standard deviations of the windows (a and b) of the fixed size $N \times N$ that strides over the input image. C_1 and C_2 are stabilizing factors that prevent the denominator from disappearing. These measures are calculated for multiple corresponding windows of gray-scaled input images and aggregated to get the final measure. Gray-scaled inputs are used to focus on structural differences between images and not changes in color.

Identity loss (Taigman, A. Polyak, and Wolf 2016) is utilized to regularize the generator and preserve the overall composition of the image. The generators are rewarded if a near identity mapping is produced when an image from the respective target domain is provided as an input image. In other words, when an image is fed into a generator of its own domain, the generator should produce an image that is nearly identical to the input. This is enforced by minimizing the $L1$ distance of the resulting image with the input image as follows:

$$\mathcal{L}_{id} = \mathbb{E}_{y \sim p(y)}\left[\left\|G_{YX}(y) - y\right\|_1\right] + \mathbb{E}_{x \sim p(x|k)}\left[\left\|G_{XY}(x) - x\right\|_1\right] \tag{4.7}$$

The overall objective function then becomes:

$$\mathcal{L}(G_{YX}, G_{XY}, D_X, D_Y) = \mathcal{L}_{adv}^{Y} + \mathcal{L}_{adv}^{X} + \alpha * \mathcal{L}_{cyc} + \beta * \mathcal{L}_{scyc} + \gamma * \mathcal{L}_{dssim} + \delta * \mathcal{L}_{id} \tag{4.8}$$

where parameters α, β, γ and δ manage the importance of different loss terms. The parameters in the generators and the discriminators are tuned by solving the above objective as:

$$G^*_{YX}, G^*_{XY} = \arg \min_{G_{YX}, G_{XY}} \max_{D_X, D_Y} \mathcal{L}(G_{YX}, G_{XY}, D_X, D_Y) \tag{4.9}$$

In the following sections, we describe the implementation and compare our results with other current state-of-the-art methods of color normalization with both multiple ($K = 2$) and single ($K = 1$) sub-domains in X.

4.4 Experiments

4.4.1 Dataset

For this paper, duodenal biopsy patches were extracted from 465 high resolution WSIs from 150 H&E stained duodenal biopsy slides (where each glass slide could have one or more biopsies). The biopsies were from patients with Celiac Disease (CD) and Environmental Enteropathy (EE). The biopsies were from children who underwent endoscopy procedures at either Site 1 (10 children <2 years with growth faltering, EE diagnosed on endoscopy, n = 34 WSI), Site 2[1] (16 children with severe acute malnutrition, EE diagnosed on endoscopy, n = 19 WSI), or Site 3[1] (63 children <18 years old with CD, n = 236 WSI; and 61 healthy children <5 years old, n = 173 WSI). It was observed that there was a significantly large stain variation between images originating from different sites. While images from Site 1 were different tones of dark blue, images from Site 3 were more pink with images from Site 2 lying somewhere in the middle of this spectrum.

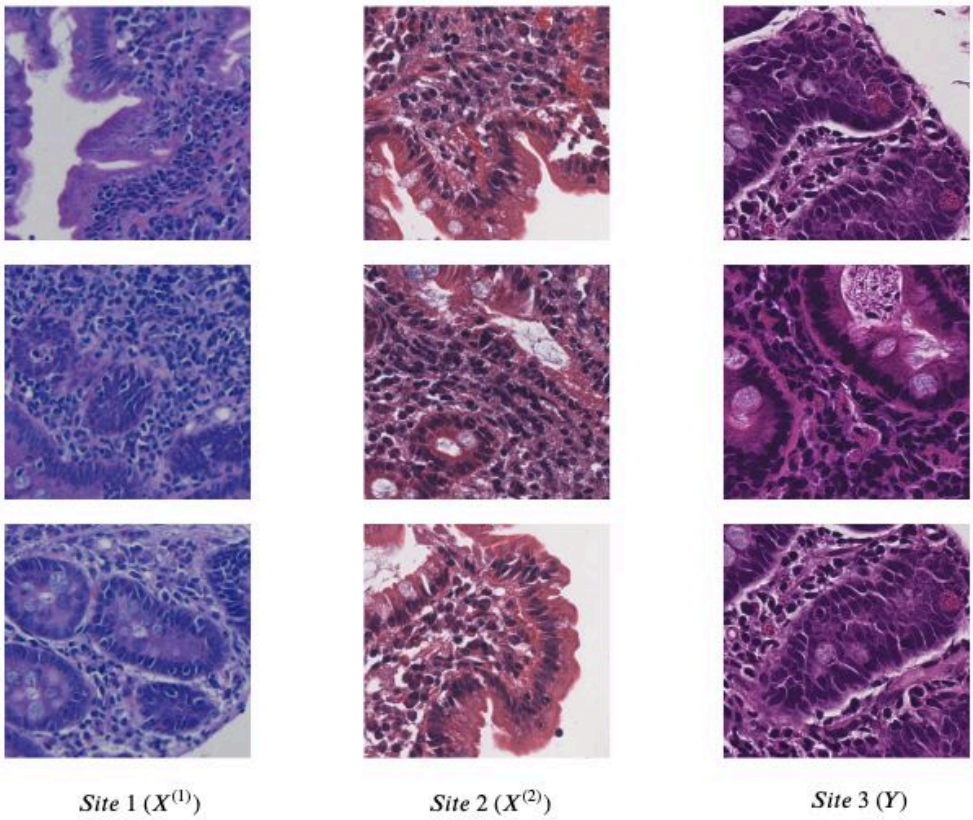


Figure 4.2: H&E stained duodenal biopsy patches created from whole slide images sourced from different locations.

There is always some degree of physical variation between histological sections from different sites. In this study, our approach and other competing methods were performed on 500×500 pixel patches generated from the images, which were further resized to 256×256 pixel to marginally reduce the resolution. In the multi-sub-domain setup, patches from Site 1 (sub-domain $X^{(1)}$) and Site 2 (sub-domain $X^{(2)}$) were both considered to be in domain X and patches from Site 3 to be in domain Y. While in single sub-domain training setup, patches from Site 1 were considered to be in domain X and Site 3 to be in domain Y. For training both X and Y had 16000 patches where $X^{(1)}$ contributed 10817 and $X^{(2)}$ 5183 patches. Testing metrics were computed on 1500 patches in each sub-domain.

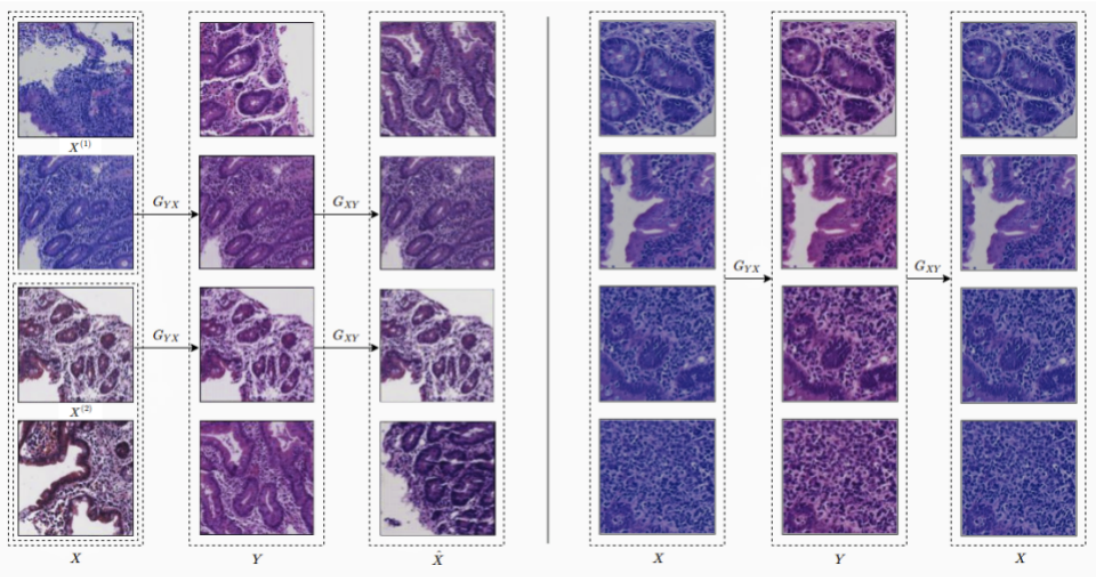


Figure 4.3: *Left*: Results when mapping was done from two sub-domains of X to Y. Patches from both domains $X^{(1)}$ and $X^{(2)}$ are translated to domain Y using G_{YX}. These generated images are then translated back to a new domain defined by a G_{XY} as a combination of stain distributions of sub-domains of X. Patches on either end of the second column are real images from domain Y and have been added to visually show the performance of G_{YX}. *Right*: Results when mapping was learnt using a single domain in X to Y.

4.4.2 Network Architecture

The generator network is a modified **U-Net** (Ronneberger, P. Fischer, and Brox 2015) which has been shown to generate excellent results in image translation tasks (Isola et al. 2017). U-Net is encoder-decoder network (G. E. Hinton and Salakhutdinov 2006) that uses skip connections between layers i and $n - i$ where n is the total number of layers in the network. In previous encoder-decoder architectures (Pathak et al. 2016; X. Wang and Gupta 2016; Yoo et al. 2016). The input is passed through a series of convolutional layers that downsample the input until a bottleneck is reached after which the information is upsampled to generate an output of the desired dimensions. Therefore, by design all information passes through the bottleneck.

In a stain normalization task, the input and the output of the network share a lot of general information that might get obscured through the flow of such a network. Skip connections in a U-Net solve this problem by circumventing the bottleneck and concatenating the output from the encoder layers to the input of the corresponding decoder layers.

The discriminator is a 4 block CNN, which eventually outputs the decision for each image. Every convolutional block in both the generator and the discriminator is a module consisting of a convolution-normalization-ReLU layers in that order. Both instance (Ulyanov, Vedaldi, and Lempitsky 2016) and batch (Ioffe and Szegedy 2015b) normalization were used; and batch normalization was empirically chosen for the final network. The convolutional layers have kernel size of 4 and stride 2, with the exception of the last layer in the discriminator which operates with stride 1.

Self-attention layers (Parikh et al. 2016) were added after every convolutional block in both the generator and the discriminator network. The self-attention mechanism complements the convolutions by establishing and leveraging long range dependencies across image regions. It help the generator synthesize images with finer details in regions based on a different spatial region in the image. Additionally the discriminator with self-attention layers is able to enforce more complex structural constraints on input images while making a decision. As described in SAGAN (H. Zhang et al. 2018), a non-local network (X. Wang, Girshick, et al. 2018) was used to apply the self-attention computation. The input features $x \in \mathbb{R}^{C \times N}$ are transformed using three different learnable functions $q(x), k(x), v(x)$ analogous to query, key and value setup in (Vaswani et al. 2017) as follows:

$$q(x) = W_q x; \qquad k(x) = W_k x; \qquad v(x) = W_v x \tag{4.10}$$

where $W_q \in \mathbb{R}^{\bar{C}\times C}$, $W_k \in \mathbb{R}^{\bar{C}\times C}$, and $W_v \in \mathbb{R}^{\bar{C}\times C}$. Also, C is the number of channels, $N = height * width$ of the feature map from the previous layer and $\bar{C}$ is an adjustable parameter. For our model, $\bar{C}$ was set as $C/8$. The attention map is further calculated as:

$$\begin{aligned} \alpha_{j,i} &= softmax(k(x_i)^T g(x_j)) \\ &= \frac{\exp\left(k(x_i)^T g(x_j)\right)}{\sum_{i=1}^{N} \exp\left(k(x_i)^T g(x_j)\right)} \end{aligned} \tag{4.11}$$

where $\alpha_{j,i}$ represents the attention placed on location i while synthesizing location j. The ouput $o \in \mathbb{R}^{C\times N}$ is calculated as:

$$o_j = \sum_{i=1}^{N} \alpha_{j,i} v(x_i) \tag{4.12}$$

The output o is then scaled and added to the initial input to give the final result,

$$y_i = \mu o_i + x_i \tag{4.13}$$

where μ is a learnable parameter that is initialized to 0.

Spectral normalization when applied on the layers of the discriminator network has been shown to stabilize the training of a GAN (Miyato et al. 2018). Moreover, based on the findings about the effect of a generator's conditioning on its performance, Zhang et al. (H. Zhang et al. 2018) argue that while training a self-attention based GAN, both the generator and the discriminator can benefit from using spectral normalization. Therefore, a spectral normalization (with spectral norm of all weight layers as 1) was added to all the networks.

4.4.3 Training Details

The parameter values of $\alpha = 10$, $\beta = 10$, $\gamma = 10$ and $\delta = 0.1$ were empirically chosen after experimentation for the evaluation model. Across all experiments, we used the Adam optimizer (Kingma and Ba 2014b) with a learning rate of 0.0002 and batch size 16. The model was trained for the first 50 epochs with a fixed learning rate and the next 50 epochs while linearly decaying the learning rate to 0. Instead of updating the discriminator with an image generated form the latest generator, a random image selected from a buffer of 50 previously generated images was used to perform the update cycle (Ashish Shrivastava et al. 2017). Least-squares adversarial loss inspired from LSGAN (Mao et al. 2017) was used instead of the described cross-entropy loss for some experiments. The least-squares loss stabilized the training but there was no significant visual difference in the results produced.

4.5 Results and Evaluation

To demonstrate the value of each introduced term in the designed loss function, an ablation study was performed. A competitive version of StainGAN (Shaban et al. 2019) was also implemented based on the information given in the paper. It was observed that the addition of self-attention layers helped the model to generate more vibrant results that preserved medically significant artifacts. For instance, the red blood cells in the second row of Figure 4.4 get visually merged with the surrounding cells when self-attention is not used. The ablation study shows shows that with the cycle consistency loss alone the forward mapping function (G_{XY}) is suppressed from providing a *many-to-one* mapping as the generated domain ($\hat{X}$) from the inverse function (G_{YX}) will overlap more with the dominant domain in the training set.

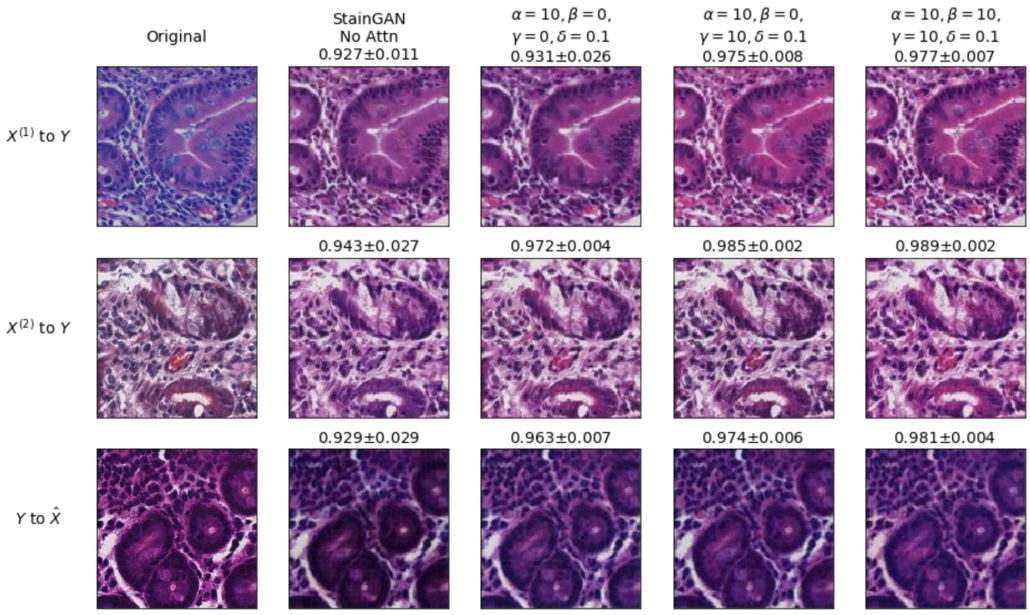


Figure 4.4: Visual and quantitative comparison of performance between StainGAN and ablation study on SAASN. The numbers indicate the overall mean ± standard deviation of the SSIM index for the transformation. All models were trained in a *many-to-one* setup.

Addition of the structural cycle consistency loss term alleviates this issue as it is stain agnostic and a combination of the said losses gives a more compelling result.

To evaluate the stain transfer, the Structural Similarity (SSIM) index is again utilized. SSIM is calculated by comparing the normalized image with the original. Both images are converted to gray-scale before beginning SSIM calculations. Our approach is compared to two of the most popular unsupervised stain normalization techniques, Macenko (Macenko et al. 2009) and Vahadane (Vahadane et al. 2016). The popular supervised approach by Khan (A. M. Khan et al. 2014) could not be tested due to lack of pixel-level labeling in our data. These results are compiled in Table 4.1. For the $X^{(1)}$ to Y and the $X^{(2)}$ to Y stain transfers, the values for SAASN are higher than the other two normalization techniques and the variance is significantly smaller. This demonstrates that SAASN is not only better at preserving structure, but also

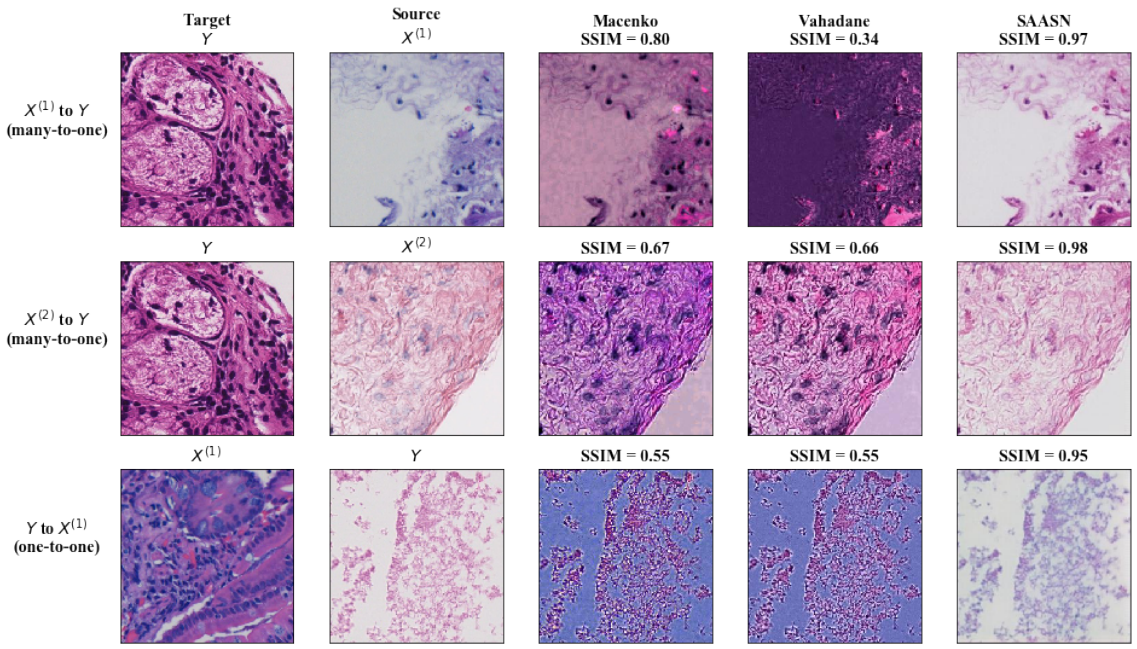


Figure 4.5: Visual comparison of performance in cases where Macenko and Vahadane techniques struggle to properly transfer stain in each scenario. The target image only applies to the Macenko and Vahadane techniques.

consistently transfers stain without major anomalies. The traditional approaches (Vahadane and Macenko) approaches can struggle if the source has a much different stain distribution than the target. This can lead to the stains appearing in the wrong areas on the normalized image. SAASN is able to leverage information from entire stain domains and therefore is not as affected by this issue. These results demonstrate that SAASN can be trusted to produce consistent stain transfers on a robust set of stain patterns in WSI patches.

In addition to assessing the structure-preserving ability of the stain normalization methods, visual comparisons are essential to ensure that the stains have transferred properly. In Figure 4.5, results are displayed for the three stain transfers. The images with the smallest $L2$-norm for combined Macenko and Vahadane SSIM values were selected to demonstrate the performance of SAASN. For $X^{(1)}$ to Y and $X^{(2)}$ to Y, the

Table 4.1: Mean ± Standard deviation of the SSIM index values for normalization across domains. For StainGAN and SAASN all values are computed for a *many-to-one* setup.

Method	$X^{(1)}$ to Y	$X^{(2)}$ to Y	Y to $X^{(1)}$
Vahadane	0.861 ± 0.108	0.919 ± 0.029	0.932 ± 0.033
Macenko	0.942 ± 0.033	0.934 ± 0.022	0.941 ± 0.020
StainGAN	0.927 ± 0.011	0.943 ± 0.027	0.929 ± 0.021
SAASN	$\mathbf{0.977 \pm 0.007}$	$\mathbf{0.989 \pm 0.002}$	$\mathbf{0.981 \pm 0.004}$

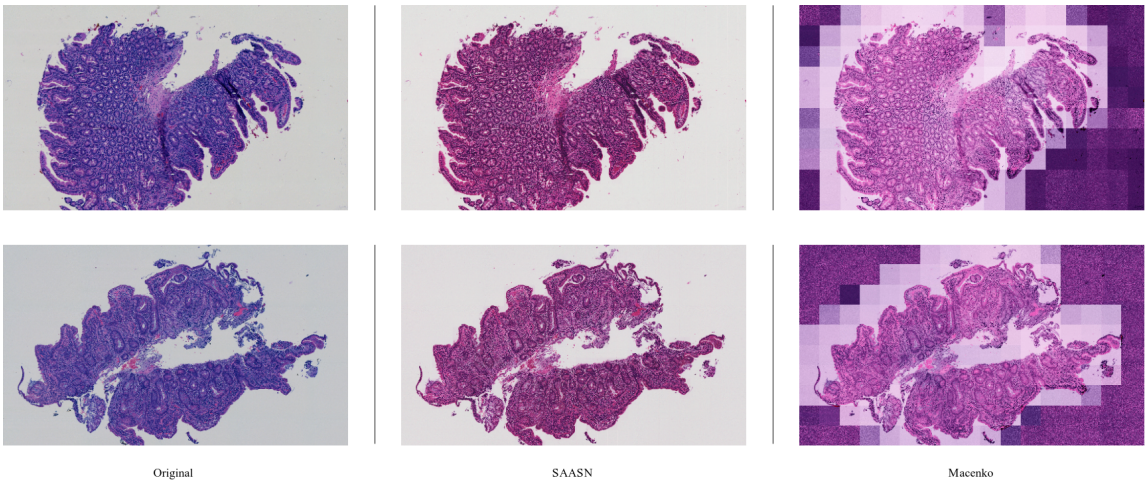


Figure 4.6: Normalized Whole Slide Image using ours and traditional approaches. Macenko was chosen because it performed better than Vahadane on our dataset. The target slide for Macenko was empirically selected to give the best translation.

same target image from domain Y is used. For Y to $X^{(1)}$, a target image from domain $X^{(1)}$ is used. The three selected source images are similar in that they all have a large majority of pixels containing connective tissue or background. The unsupervised approaches can struggle executing color deconvolution on these types of images. This is apparent in the Macenko and Vahadane normalizations shown in Figure 4.5. The stains are either inverted (hematoxylin-like color transferred to the background) or confusing connective tissue as an actual cell structure. Meanwhile, SAASN did not have difficulty identifying the connective tissue or background pixels in the source

image.

A similar visual can be obtained using the highest $L2$-norm values. These are the examples where the traditional methods performed the best[1]. We found that even though the SSIMs were all similar and very high for all three normalizations, the stains were not all transferred properly. Vahadane and Macenko are able to maintain structure, but may not visually match the target distribution or the proper background pixel color.

Stain normalization is crucial for bias-free visual examination of Whole Slide Images (WSIs) and diagnosis by medical practitioners in control trial settings. WSIs have very large dimensions and cannot be normalized without resizing to a computationally tractable size which results in a significant loss in resolution. To normalize WSIs, they must be split into patches, normalized and then stitched back together. Traditional methods perform computations for transformation independently on these patches. As a result, it is impossible to reconstruct a WSI that has a consistent stain and is indistinguishable from an original image in the target domain. As demonstrated in Figure 4.6, for our method, since the trained weights of the mapping function are constant during this transformation, the reconstructed WSI could not be distinguished from original images and thus is easier for medical professionals to hold diagnosis trails.

In order to a validate a successful translation three medical professionals, including a board-certified pathologist, completed a blind review of WSIs normalized via traditional and our method as shown in Figure 4.6. The pathologist confirmed that medically relevant cell types (polymorphonuclear neutrophils, epithelial cells, eosinophils, lymphocytes, goblet cells, paneth cells, neuroendocrine cells) were not lost during

[1]Please refer supplemental material for the figure and additional results.

translation. The pathologist further observed that our method was able to completely preserve the structure and the density of all of these cell types which traditional methods only partially preserved. Specifically, the eosinophilic granules in paneth cells, neuroendocrine cells and eosinophils were not appreciated in traditionally stain normalized WSIs which made it difficult to differentiating these cells from each other.

4.6 Discussion

The proposed framework is successful in effective translation of the stain appearance of histopathological images while preserving the biological features in the process. This setup was specifically designed to accommodate a *many-to-one* stain transfer situation in which multiple stains are converted to a common domain. SAASN is compared to other leading stain normalization techniques using duodenal biopsy image data originating from three sites with different stain appearances. SAASN consistently performed successful stain transfers even when the other techniques failed due to large variations between the source and target image stains and unconventional input image structures. Results also show that SAASN outperformed traditional methods at preserving the cellular structures. We contend that the proposed unsupervised image to image translation approach can be successfully applied to general *many-to-one* image translation problems outside the medical domain as well. Results for out of domain implementation is added to the supplemental material.

5

Nuclei-Aware Semantic Histopathology Image Generation Using Diffusion Models

In recent years, computational pathology has seen tremendous progress driven by deep learning methods in segmentation and classification tasks aiding prognostic and diagnostic settings. Nuclei segmentation, for instance, is an important task for diagnosing different cancers. However, training deep learning models for nuclei segmentation requires large amounts of annotated data, which is expensive to collect and label. This necessitates explorations into generative modeling of histopathological images. In this work, we use recent advances in conditional diffusion modeling to formulate a first-of-its-kind nuclei-aware semantic tissue generation framework to generate synthetic tissue patches given a semantic instance mask of up to six different nuclei types. Our method enables pixel-perfect nuclei localization in generated samples. These synthetic images are useful in applications in pathology pedagogy, validation of models, and supplementation of existing nuclei segmentation datasets. Implementation: https://github.com/4m4n5/NASDM.

5.1 Introduction

Histopathology relies on hematoxylin and eosin (H&E) stained biopsies for microscopic inspection to identify visual evidence of diseases. Pathologists examine highlighted tissue characteristics to diagnose diseases, including different cancers. A correct diagnosis, therefore, is dependent on the pathologist's training and prior exposure to a wide variety of disease subtypes (Xie et al. 2020). This presents a challenge, as some disease variants are extremely rare, making visual identification difficult. In recent years, deep learning methods have aimed to alleviate this problem by designing discriminative frameworks that aid diagnosis (Van der Laak, Litjens, and Ciompi 2021; Y. Wu et al. 2022). Segmentation models find applications in spatial identification of different nuclei types (Graham, Vu, et al. 2019). However, generative modeling in histopathology is relatively unexplored. Generative models can be used to generate histopathology images with specific characteristics, such as visual patterns identifying rare cancer subtypes (Fajardo et al. 2021). As such, generative models can be sampled to emphasize each disease subtype equally and generate more balanced datasets, thus preventing dataset biases getting amplified by the models (Hall et al. 2022). Generative models have the potential to improve the pedagogy, trustworthiness, generalization, and coverage of disease diagnosis in the field of histology by aiding both deep learning models and human pathologists. Synthetic datasets can also tackle privacy concerns surrounding medical data sharing. Additionally, conditional generation of annotated data adds even further value to the proposition as labeling medical images involves tremendous time, labor, and training costs. Recently, denoising diffusion probabilistic models (DDPMs) (Ho, Jain, and Abbeel 2020) have achieved tremendous success in conditional and unconditional generation of real-world images (Dhariwal and A. Nichol 2021). Further, the semantic diffusion

model (SDM) demonstrated the use of DDPMs for generating images given semantic layout (W. Wang et al. 2022). In this work, (1) we leverage recently discovered capabilities of DDPMs to design a first-of-its-kind nuclei-aware semantic diffusion model (NASDM) that can generate realistic tissue patches given a semantic mask comprising of multiple nuclei types, (2) we train our framework on the Lizard dataset (Graham, Jahanifar, et al. 2021) consisting of colon histology images and achieve state-of-the-art generation capabilities, and (3) we perform extensive ablative, qualitative, and quantitative analyses to establish the proficiency of our framework on this tissue generation task.

5.2 Background

Deep learning based generative models for histopathology images have seen tremendous progress in recent years due to advances in digital pathology, compute power, and neural network architectures. Several GAN-based generative models have been proposed to generate histology patches (Levine et al. 2020; Xue et al. 2021; Zhou and Yin 2022). However, GANs suffer from problems of frequent mode collapse and overfitting their discriminator (Xiao, Kreis, and Vahdat 2021). It is also challenging to capture long-tailed distributions and synthesize rare samples from imbalanced datasets using GANs. More recently, denoising diffusion models have been shown to generate highly compelling images by incrementally adding information to noise (Ho, Jain, and Abbeel 2020). Success of diffusion models in generating realistic images led to various conditional (Kawar et al. 2022; Saharia, Chan, et al. 2022; Saharia, Ho, et al. 2022) and unconditional (Dhariwal and A. Nichol 2021; Ho, Saharia, et al. 2022; A. Q. Nichol and Dhariwal 2021) diffusion models that generate realistic sam-

ples with high fidelity. Following this, a morphology-focused diffusion model has been presented for generating tissue patches based on genotype (Moghadam et al. 2023). Semantic image synthesis is a task involving generating diverse realistic images from semantic layouts. GAN-based semantic image synthesis works (Tan, Chai, et al. 2021; Tan, D. Chen, et al. 2021; Park et al. 2019) generally struggled at generating high quality and enforcing semantic correspondence at the same time. To this end, a semantic diffusion model has been proposed that uses conditional denoising diffusion probabilistic model and achieves both better fidelity and diversity (W. Wang et al. 2022). We use this progress in the field of conditional diffusion models and semantic image synthesis to formulate our NASDM framework.

5.3 Method

5.3.1 Denoising Diffusion Probabilistic Models

Denoising diffusion probabilistic models (DDPMs) (Ho, Jain, and Abbeel 2020) represent a fairly recent and significant advance in generative modeling, harnessing a sequential denoising process inspired by principles of non-equilibrium thermodynamics to synthesize high-fidelity data. A DDPM comprises of a forward diffusion process that iteratively perturbs data with Gaussian noise, transforming it into a tractable noise distribution through a Markov chain of latent variables. The reverse diffusion process, which is the key innovation of DDPMs, involves training a neural network to approximate the reverse transitions, effectively learning to denoise the perturbed data step-by-step. This reverse process is modeled as a series of Gaussian transitions conditioned on the current state, with the neural network effectively denoising the

perturbed samples. This method ensures stable training dynamics, mitigating issues commonly encountered in Generative Adversarial Networks (GANs) (Goodfellow et al. 2014), and achieves state-of-the-art performance in various generative tasks, including high-resolution image synthesis, audio generation, and more. Consequently, DDPMs have established themselves as a robust and versatile framework for high-dimensional data generation with remarkable fidelity and diversity. The following subsections describe the formulations of the forward and the reverse diffusion process in detail.

5.3.2 Forward Diffusion Process

DDPMs are formulated from the variational perspective where the forward diffusion systematically transforms data into a noise distribution through a series of incremental additions of Gaussian noise. Formally, this process yields a Markov Chain of latent variables $\{x_t\}_{t=0}^{T}$, which are of the same dimensionality as the original data, where x_0 is the original data sample, and x_T converges to an isotropic Gaussian distribution. The data is sampled from $q(x_0)$, which represents the real data distribution. At each time step t, Gaussian noise is added to the data controlled by a predefined variance schedule controlled by parameters $\{\beta\}_{t=1}^{T}$. Specifically, each step of the forward diffusion is defined as,

$$q(x_t \mid x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}), \tag{5.1}$$

$$q(x_{1:T} \mid x_0) = \prod_{t=1}^{T} q(x_t \mid x_{t-1}), \tag{5.2}$$

where $\{\beta\}_{t=1}^{T} \in [0, 1)$ is the variance schedule across diffusion steps, $\mathbf{I}$ is the identity matrix and $\mathcal{N}(x; \mu, \sigma)$ represents a normal distribution with mean μ and covariance σ. Note that a key property of Gaussian distributions is that the composition of multiple Gaussian perturbations remains Gaussian. This means if we add Gaussian noise to a Gaussian-distributed variable, the resulting distribution is still Gaussian. This property allows us to combine the noise addition steps over multiple time steps into a single Gaussian distribution. Given the forward process transitions, we can derive the marginal distribution of x_t conditioned on the original data x_0 by recursively applying the transition probabilities. Due to the linear nature of the noise addition and the properties of Gaussian distributions, the marginal distribution $q(x_t \mid x_0)$ can be expressed as a Gaussian distribution,

$$q(x_t \mid x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t)\mathbf{I}), \tag{5.3}$$

where $\alpha_t = (1 - \beta_t)$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$. This property is particularly advantageous, as it enables the efficient sampling of noisy data at any intermediate time step without requiring an iterative simulation from x_0 to x_t.

5.3.3 Reverse Diffusion Process

The reverse diffusion process in DDPMs is a generative mechanism which inverts the forward diffusion process through a sequence of learned denoising steps. This process is designed to transform samples from the noise distribution back into coherent data samples. Specifically, the reverse process aims to approximate the conditional distributions $p_\theta(x_{t-1}|x_t)$ through a neural network, where each x_{t-1} depends only on x_t. The reverse transitions are modeled as Gaussian distributions as follows,

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)), \tag{5.4}$$

$$p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t), \tag{5.5}$$

where p_θ is a neural network that represents the learned reverse process with parameters θ. During sampling, the model begins with a noise vector $x_T \sim \mathcal{N}(0, \mathbf{I})$ and iteratively applies the reverse transitions $p_\theta(x_{t-1} \mid x_t)$ using the trained denoising neural network to generate a sequence of latent variables that culminate in the reconstructed data sample x_0. This iterative process effectively denoises the initial noise, step by step, reconstructing the data distribution in reverse order. The success of the reverse diffusion process is contingent on a well-trained denoising network p_θ as it ensures that the final samples are realistic and diverse, closely matching the original data distribution.

5.3.4 Training The Model

Optimizing the parameters θ of the denoising neural network involves minimizing a variational lower bound on the negative log-likelihood of the data,

$$\mathbb{E}[-\log p_\theta(x_0)] \leq \mathbb{E}_q \left[-\log \frac{p_\theta(x_{0:T})}{q(x_{1:T} \mid x_0)} \right] \tag{5.6}$$

$$\leq \mathbb{E}_q \left[-\log p\left(x_T\right) - \sum_{t \geq 1} \log \frac{p_\theta\left(x_{t-1} \mid x_t\right)}{q\left(x_t \mid x_{t-1}\right)} \right] = L, \tag{5.7}$$

which decomposes into a series of Kullback-Leibler (KL) divergence terms between the true posterior of the forward process and the learned reverse process, along with a reconstruction term:

$$
\begin{aligned}
L &= L_T + \sum_{t>1} L_{t-1} + L_0, && (5.8)\\
L_T &= D_{KL}\left(q\left(x_T \mid x_0\right) \| p\left(x_T\right)\right), && (5.9)\\
L_{t-1} &= D_{KL}\left(q\left(x_{t-1} \mid x_t, x_0\right) \| p_\theta\left(x_{t-1} \mid x_t\right)\right), && (5.10)\\
L_0 &= -\log p_\theta\left(x_0 \mid x_1\right). && (5.11)
\end{aligned}
$$

Except for L_0, each term of the decomposition in eq 5.8 is a KL-divergence between two Gaussian distributions and hence has a closed-form solution. The KL-divergence terms ensure that the neural network accurately captures the denoising process by aligning the learned distributions $p_\theta(x_{t-1}|x_t)$ with the true posteriors $q(x_{t-1}|x_t, x_0)$. Notice that L_T does not depend on the parameters θ and can be ignored safely during optimization. Upon simplification via Bayes theorem, the posteriors $q(x_{t-1}|x_t, x_0)$ can be represented in terms of parameters β_t and $\bar{\alpha}_t$ as follows,

$$
q\left(x_{t-1} \mid x_t, x_0\right) = \mathcal{N}\left(x_{t-1}; \tilde{\mu}\left(x_t, x_0\right), \tilde{\beta}_t \mathbf{I}\right), \tag{5.12}
$$

where,

$$
\begin{aligned}
\tilde{\mu}_t\left(x_t, x_0\right) &= \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} x_0 + \frac{\sqrt{\alpha_t}\left(1-\bar{\alpha}_{t-1}\right)}{1-\bar{\alpha}_t} x_t, && (5.13)\\
\tilde{\beta}_t &= \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t. && (5.14)
\end{aligned}
$$

For $p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$, the original DDPM work (Ho, Jain,

and Abbeel 2020) suggests setting $\Sigma_\theta(x_t, t) = \sigma_t^2 \mathbf{I}$ to untrained time-dependent constants. They find that both extremes of $\sigma_t^2 = \beta_t$ and $\sigma_t^2 = \tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$ performed similarly. Now with $p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2 \mathbf{I})$, the loss terms can be calculated in a Rao-Blackwellized fashion with closed-form expressions as follows,

$$L_{t-1} = \mathbb{E}_q \left[\frac{1}{2\sigma_t^2} \left\| \tilde{\mu}_t(x_t, x_0) - \mu_\theta(x_t, t) \right\|^2 \right] + C, \tag{5.15}$$

where C includes the constant terms independent of θ. There are multiple other ways to parameterize $\mu_\theta(x_t, t)$. For instance, the network could also predict the noise ϵ added to x_0, and this noise could be used to predict x_0 via

$$x_0 = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon \right). \tag{5.16}$$

Ho et al. (Ho, Jain, and Abbeel 2020) found that predicting ϵ works best with the following simplified loss function,

$$L_{t-1} = \mathbb{E}_{x_0, \epsilon} \left[\frac{\beta_t^2}{2\sigma_t^2 \alpha_t (1-\bar{\alpha}_t)} \left\| \epsilon - \epsilon_\theta \left(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t} \epsilon, t \right) \right\|^2 \right]. \tag{5.17}$$

The network is trained using stochastic gradient descent, where each training step involves adding noise to a data sample and then predicting the noise to minimize the objective function. The amount of noise added can be determined by uniformly sampling t for each image in each minibatch. Overall, the reverse process mean function approximator, μ_θ, can be used to predict $\tilde{\mu}_t$, or, it can be reparameterized to instead predict ϵ. Ho et al. (Ho, Jain, and Abbeel 2020) report that the ϵ-prediction parameterization not only resembles Langevin dynamics but also simplifies the diffusion model's variational bound to an objective akin to denoising score matching (Song

et al. 2020). Therefore, efficient training can be achieved by optimizing random terms of L using stochastic gradient descent.

5.3.5 Generating Samples

Sampling from a diffusion model involves simulating the reverse denoising process to systematically transform noise into data through a sequence of learned probabilistic steps. This process begins with an initial sample drawn from a standard Gaussian distribution which serves as the prior. Specifically, the process starts by initializing a noise vector $x_T \sim \mathcal{N}(0, \mathbf{I})$, where T represents the total number of diffusion steps. The idea of the reverse diffusion process is to iteratively apply the reverse transition model to progressively denoise the sample. At each time step t, from T down to 1, the model computes x_{t-1} using the following Gaussian distribution:

$$x_{t-1} \sim p_\theta \left(x_{t-1} \mid x_t \right) = \mathcal{N} \left(x_{t-1}; \mu_\theta \left(x_t, t \right), \Sigma_\theta \left(x_t, t \right) \right). \tag{5.18}$$

Here, $\mu_\theta(x_t, t)$ and $\Sigma_\theta(x_t, t)$ are the mean and variance predicted by a neural network parameterized by θ. As described above, typically, the neural network predicts the mean, while the variance can either be fixed or predicted by the network as well. Alternatively, when using the ϵ-based parameterization involves predicting the noise added at each step, instead of predicting the mean directly. This approach can be formalized as:

$$\mu_\theta \left(x_t, t \right) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta \left(x_t, t \right) \right). \tag{5.19}$$

The iterative denoising process involves repeating the sampling step for each time step, gradually refining x_t until x_0 is obtained. This stepwise process effectively

removes the noise added during the forward diffusion, reconstructing a sample from the data distribution. By the end of the iterations, the final output, x_0, represents a sample from the learned data distribution. Due to the stochastic nature of each reverse transition, each run of this process can generate a unique data sample.

In summary, the diffusion model uses the learned reverse transitions to convert initial noise into high-quality data samples, effectively reversing the forward diffusion process. This ensures that the generated samples are consistent with the original data distribution, showcasing the model's ability to produce realistic and diverse outputs.

5.3.6 Conditional Sampling Using Guidance

Diffusion models can be used to generate samples conditioned on desired information such as class labels, text descriptions, or other attributes. This is achieved by incorporating a mechanism known as guidance in the sampling process. Guidance-based sampling in diffusion models is a technique designed to enhance the fidelity and controllability of the generated samples by incorporating additional information or constraints into the sampling process. This approach modifies the reverse diffusion process to include guidance from an auxiliary model or a predefined condition, which can steer the generative model towards more desirable outputs. One common implementation of guidance-based sampling involves using a classifier to guide the diffusion model, where the gradients from the classifier are combined with the reverse diffusion steps to bias the sample generation towards specific classes or features. Another approach, known as classifier-free guidance, directly conditions the diffusion model on the desired attributes, enabling the generation of samples that adhere to specified conditions without requiring an explicit classifier. By integrat-

ing these guidance mechanisms, diffusion models can produce higher quality, more targeted samples, thereby expanding their applicability in tasks requiring controlled generation such as conditional image synthesis, text-to-image generation, and other domains where adherence to specific criteria is crucial. Following sections describe both these mechanisms in further detail.

Classifier Guidance

Classifier guidance (Ho and Salimans 2022) is a mechanism used in diffusion probabilistic models to perform conditional generation by incorporating gradients from a pretrained classifier into the sampling process. This method involves using a separate classifier to guide the diffusion model towards generating samples that satisfy specific conditions, such as class labels. The primary objective of classifier guidance is to bias the reverse diffusion process such that the generated samples adhere to a desired condition. This is achieved by using the gradient of an independently trained classifier's output with respect to the input data, effectively steering the generation towards higher probability regions of the conditioned distribution.

Training: For classifier guidance, the training phase of the diffusion model remains unchanged. The model is trained to learn the reverse denoising processes without any conditioning. The forward process progressively adds noise to the data, while the reverse process learns to denoise, reconstructing the original data distribution as described above. Critically, a separate classifier $p_\phi(y \mid x_t)$, with parameters ϕ, is trained to predict the condition y (e.g., a class label) given a noisy sample x_t from the diffusion process. Note that this classifier needs to be trained on noisy data generated by the forward diffusion process at various time steps t in order to provide meaningful guidance.

Sampling: During the sampling phase, classifier guidance modifies the standard reverse diffusion process to incorporate guidance signal from the classifier using its gradients. Specifically, the sampling process begins with initializing an initial noise vector $x_T \sim \mathcal{N}(0, \mathbf{I})$. Then for each time step t from T down to 1, the reverse transition is adjusted from the one highlighted in eq. 5.18 using the gradient of the classifier's log-probability with respect to x_t, resulting in

$$x_{t-1} \sim p_{\theta\phi}(x_{t-1} \mid x_t, y), \tag{5.20}$$

where,

$$p_{\theta\phi}(x_{t-1} \mid x_t, y) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t) + \alpha \nabla_{x_t} \log p_\phi(y \mid x_t), \Sigma_\theta(x_t, t)). \tag{5.21}$$

Here, α is a scaling factor that determines the strength of the guidance, $\mu_\theta(x_t, t)$ and $\Sigma_\theta(x_t, t)$ are the mean and variance predicted by the diffusion model, and $\nabla_{x_t} \log p_\phi(y|x_t)$ is the gradient provided by the classifier.

Overall, the classifier p_ϕ predicts the probability of the condition y given the current noisy sample x_t. Hence, the gradient $\nabla_{x_t} \log p_\phi(y|x_t)$ indicates the direction in which the sample x_t should be adjusted to increase the probability of the desired condition y. This computed gradient is scaled by a hyper-parameter α and added to the predicted mean $\mu_\theta(x_t, t)$ of the reverse transition. This adjustment effectively biases the sample generation process towards samples that the classifier deems more likely to belong to the desired condition y.

Conclusion: Classifier guidance enables the generation of high-quality conditional samples without needing to retrain the entire diffusion model with the conditions. However, there are some considerations. The scaling factor α must be carefully tuned.

If α is too high, the guidance may overly distort the samples, leading to poor quality. If too low, the guidance may be insufficient to influence the sampling effectively. Additionally, the effectiveness of this method depends on the robustness of the classifier. The classifier must accurately predict conditions from noisy data at various time steps. Finally, computing the gradients for each time step adds computational overhead, making the sampling process more resource-intensive.

Classifier-Free Guidance

Classifier-free guidance (Ho and Salimans 2022) is a technique used to generate conditional samples without relying on an explicit classifier to provide gradients. Instead, the model itself is trained to handle both conditional and unconditional sampling, allowing for a more integrated and flexible approach to conditional generation. This method involves training the model with and without conditioning, allowing it to take advantage of both types of information during sampling.

Training: For classifier-free guidance, during the training phase, the diffusion model is trained on both conditioned and unconditioned data. Specifically, the model learns to predict the reverse diffusion steps for samples with and without a given condition. This dual training approach involves augmenting the dataset with conditions (e.g., class labels or other attributes) and also training on the same data without these conditions to allow the model to generalize effectively. Formally, this involves training the model with two formulations both $p_\theta(x_{t-1} \mid x_t, y)$ and $p_\theta(x_{t-1} \mid x_t)$ where y is the condition. The model is trained to minimize the loss for both conditioned and unconditioned predictions, thereby learning to handle both scenarios. Practically, this is done by randomly dropping the condition during training for a certain percentage (e.g., $\sim 10\%$) of optimization iterations.

Sampling: During sampling, classifier-free guidance combines the predictions from the conditional and unconditional models to guide the generation process. The key idea is to leverage the unconditioned model to adjust the conditioned generation, ensuring that the samples adhere to the desired attributes while maintaining high quality. After starting with an initial noise vector $x_T \sim \mathcal{N}(0, \mathbf{I})$, for each time step t from T down to 1, compute the reverse transition for both the conditioned and unconditioned models,

$$x_{t-1}^{(cond)} \sim p_\theta(x_{t-1} \mid x_t, y), \tag{5.22}$$

$$x_{t-1}^{(uncond)} \sim p_\theta(x_{t-1} \mid x_t). \tag{5.23}$$

These transitions are then combined using a guidance scale factor w to control the influence of the condition as,

$$x_{t-1} \sim x_{t-1}^{(cond)} + w \left(x_{t-1}^{(cond)} - x_{t-1}^{(uncond)} \right), \tag{5.24}$$

where w adjusts the strength of the guidance, effectively interpolating between the conditioned and unconditioned predictions.

Conclusion: The combination of the conditioned and unconditioned predictions allows the model to generate samples that adhere to the desired conditions while leveraging the unconditioned model's ability to produce high-quality samples. By adjusting the guidance scale w, the generation process can be fine-tuned to balance adherence to the condition with overall sample quality. Classifier-free guidance offers several advantages over classifier-based methods. By integrating the condition directly into the model, it eliminates the need for a separate classifier, simplifying the overall architecture and reducing the potential for mismatches between the classifier

and the diffusion model. Additionally, this method provides more flexibility in handling various types of conditions, including those that may be difficult to encode with a classifier. However, careful tuning of the guidance scale w is essential to achieve the desired balance between conditional fidelity and sample quality. If w is too high, the generated samples may become distorted; if too low, the samples may not adequately reflect the desired conditions.

This section demonstrates a framework for generating tissue patches conditioned on semantic layouts of nuclei. Given a nuclei segmentation mask, the model aims to generate realistic synthetic patches. For this demonstration, (1) the first-of-its-kind Nuclei-Aware Semantic Diffusion Model (NASDM) (Aman Shrivastava and Fletcher 2023) is described that can generate realistic tissue patches given a semantic mask comprising multiple nuclei types, (2) it is trained on the graham2021lizard dataset (Graham, Jahanifar, et al. 2021) consisting of colon histology images, achieving state-of-the-art generation capabilities, and (3) extensive ablative, qualitative, and quantitative analyses are provided to establish the proficiency of the framework on this semantics driven tissue generation task.

5.3.7 Data Description

The lizard dataset (Graham, Jahanifar, et al. 2021) is used to demonstrate the NASDM method. This dataset comprises histology image regions of colon tissue from six different data sources at 20× objective magnification. Full segmentation annotation for different types of nuclei—namely, epithelial cells, connective tissue cells, lymphocytes, plasma cells, neutrophils, and eosinophils accompanies the images. A generative model trained on this dataset can be employed to effectively synthesize

colonic tumor micro-environments. The dataset includes 238 image regions, with an average size of 1055×934 pixels. Due to substantial visual variations across images, a representative test set is constructed by randomly sampling a 7.5% area from each image and its corresponding mask to be held out for testing. The test and train image regions are further divided into smaller image patches of 128×128 pixels at two different objective magnifications: (1) at $20\times$, the images are directly split into 128×128 pixel patches, whereas (2) at $10\times$, 256×256 patches are generated and resized to 128×128 for training. To utilize the data exhaustively, patching is performed with a 50% overlap in neighboring patches. Consequently, at (1) $20\times$, a total of 54,735 patches are extracted for training, with 4,991 patches held out, while at (2) $10\times$ magnification, 12,409 training patches are generated, and 655 patches are held out.

5.3.8 Stain Normalization

A common issue in training models with H&E stained histopathology slides is the visual bias introduced by variations in the staining protocol and the raw materials of chemicals, leading to different colors across slides prepared at different labs (Bejnordi et al. 2014). To address this, several stain-normalization methods have been proposed to normalize all tissue samples to mimic the stain distribution of a given target slide. The earliest approaches to stain normalization mainly involved basic style transfer techniques. One such method, histogram specification, aimed to match the histogram statistics of the source image with those of the target image (Coltuc, Bolon, and Chassery 2006). This technique is effective only when the source and target images have similar color distributions. Enforcing this normalization can introduce artifacts that compromise the structural integrity of the source image. Reinhard (Reinhard et

al. 2001) further demonstrated that color transfer using histogram specification could be conducted in the decorrelated CIELAB color space, which approximates the human visual system. For H&E stained histology images, the appropriate color space should accurately represent the presence or absence of each stain in each pixel. Researchers developed advanced stain normalization methods that surpass the performance of the histogram specification technique by utilizing stain separation. These methods begin by converting an RGB image into Optical Density (OD), using the formula $OD = \log \frac{I_0}{I}$, where I_0 represents the maximum possible illumination intensity of the image and I is the RGB image. In the OD space, color deconvolution (CD) becomes more straightforward because the stains exhibit a linear relationship with the OD values. The CD process is typically represented as $OD = VS$, where V is the matrix of stain vectors and S is the stain density map. The stain density map preserves the cell structures of the source image, while the stain vectors are adjusted to match the stain colors of the target image. One such method, the structure-preserving color normalization scheme introduced by Vahadane et al. (Vahadane et al. 2016) is used for its effectiveness and simplicity in this demonstration, to transform all slides to match the stain distribution of an empirically chosen slide from the training dataset.

5.3.9 Conditional Semantic Mask Generation

To generate semantic masks of histological nuclei encompassing six distinct types—epithelial cells, lymphocytes, connective, neutrophils, plasma, and eosinophil cells—we utilized a conditional diffusion model conditioned on a one-hot encoded vector specifying the nuclei types to include in the mask. The method involves training a diffusion probabilistic model on just the annotation masks extracted from the lizard dataset, where each nucleus was labeled according to its type. The one-hot encoded

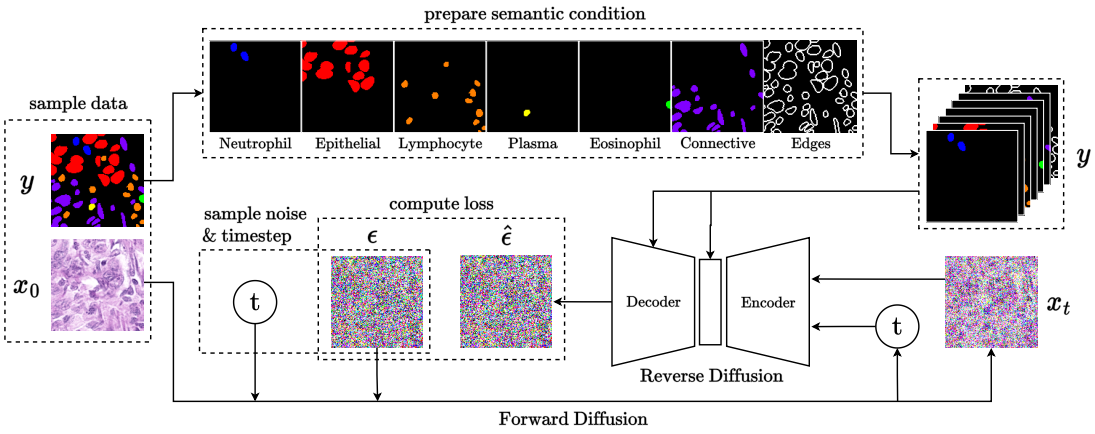


Figure 5.1: **NASDM training framework:** Given a real image x_0 and semantic mask y, we construct the conditioning signal by expanding the mask and adding an instance edge map. We sample timestep t and noise ϵ to perform forward diffusion and generate the noised input x_t. The corrupted image x_t, timestep t, and semantic condition y are then fed into the denoising model which predicts $\hat{\epsilon}$ as the amount of noise added to the model. Original noise ϵ and prediction $\hat{\epsilon}$ are used to compute the loss in (5.27).

vector served as a conditioning input, integrated into the diffusion process by passing the one-hot vector through a linear layer and adding the output to the time embedding. The diffusion model architecture used is based on a U-Net backbone enhanced with attention mechanisms to capture spatial dependencies and improve long range dependencies. During training, the model learns to reverse the diffusion process in a manner conditioned on the specified nuclei types, effectively generating semantic masks that includes only the nuclei indicated by the one-hot vector. This approach enables flexible and controllable generation of histological nuclei masks. These synthetic semantic masks can then be used with the trained NASDM model that is described in the following sections to enable infinite histological data generation that is already annotated.

5.3.10 Nuclei-Aware Semantic Diffusion Model

The formulation of NASDM derives from conditional diffusion models. As discussed, a conditional diffusion model aims to maximize the likelihood $p_\theta(x_0 \mid y)$, where data x_0 are sampled from the conditional data distribution, $x_0 \sim q(x_0 \mid y)$, and y represents the conditioning signal. As discussed above, a diffusion model consists of two intrinsic processes. The forward diffusion process that systematically destroys the information in a given sample and the reverse diffusion process which incrementally adds information by denoising a corrupted sample. When formulating a conditional diffusion model, the forward diffusion process can ignore the conditioning signal and Gaussian noise can be incrementally added to corrupt the data sample x_0 using the same description in Section 5.3.2. However, the denoising process is designed to incorporate the conditioning signal and is defined as a Markov chain with learned Gaussian transitions starting from pure noise, $p(x_T) \sim \mathcal{N}(0, \mathbf{I})$ and is parameterized as a neural network with parameters θ as

$$p_\theta(x_{0:T} \mid y) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t, y). \tag{5.25}$$

Hence, for each denoising step from t to $t-1$,

$$p_\theta(x_{t-1} \mid x_t, y) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, y, t), \Sigma_\theta(x_t, y, t)). \tag{5.26}$$

It has been shown that the combination of q and p here is a form of a variational auto-encoder (Kingma and Welling 2013), and hence the variational lower bound (VLB) can be described as a sum of independent terms, $L_{vlb} := L_0 + ... + L_{T-1} + L_T$, where each term corresponds to a noising step as described earlier in equation 5.8. As described

in previous sections, the time step t is randomly sampled during training, and the expectation $E_{t,x_0,y,\epsilon}$ is used to estimate the loss L_{vlb} and optimize the parameters θ. The denoising neural network, as discussed, can be parameterized in various ways. In NASDM, a noise prediction-based formulation results in superior image quality. Consequently, the NASDM denoising model is trained to predict the noise added to the input image given the semantic layout y and the time step t using the loss described below:

$$L_{\text{simple}} = E_{t,x,\epsilon}\left[\|\epsilon - \epsilon_\theta(x_t, y, t)\|_2\right]. \tag{5.27}$$

It is important to note that the given simplified loss function does not provide a training signal for $\Sigma_\theta(x_t, y, t)$. To address this, following the improved DDPMs strategy (Watson et al. 2021), a network is trained to predict an interpolation coefficient v for each dimension. This coefficient is then converted into variances,

$$\Sigma_\theta(x_t, y, t) = \exp\left(v \log \beta_t + (1 - v) \log \widetilde{\beta}_t\right). \tag{5.28}$$

This is then directly optimized using L_{vlb}, which is the KL divergence between the estimated distribution $p_\theta(x_{t-1} \mid x_t, y)$ and the diffusion posterior $q(x_{t-1} \mid x_t, x_0)$, formulated as,

$$L_{vlb} = D_{KL}(p_\theta(x_{t-1} \mid x_t, y) \parallel q(x_{t-1} \mid x_t, x_0)) \tag{5.29}$$

During this optimization, a stop gradient is applied to $\epsilon(x_t, y, t)$, allowing overall L_{vlb} to guide $\Sigma_\theta(x_t, y, t)$, while L_{simple} in equation 5.27 primarily guides $\epsilon(x_t, y, t)$. The overall loss is then a weighted sum of these two objectives, as follows:

$$L_{\text{hybrid}} = L_{\text{simple}} + \lambda L_{\text{vlb}}. \tag{5.30}$$

5.3.11 Conditioning on a Semantic Mask

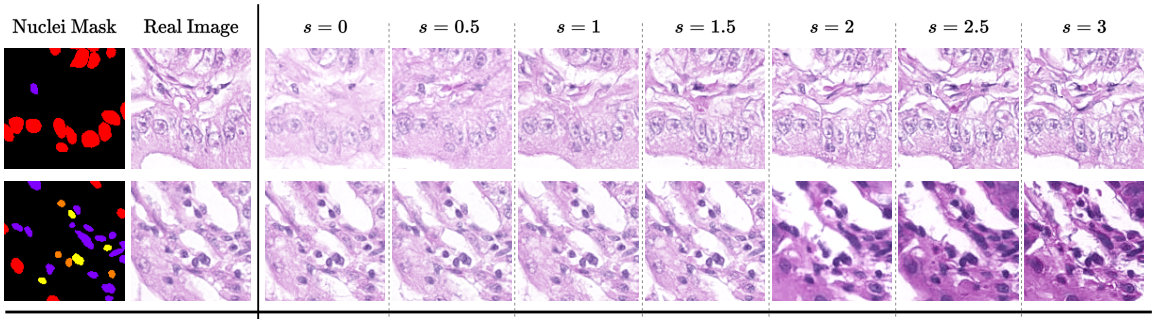
NASDM requires our neural network noise predictor $\epsilon_\theta(x_t, y, t)$ to effectively process the information from the nuclei semantic map. For this purpose, we leverage a modified U-Net architecture described in Wang et al. (W. Wang et al. 2022), where the time step is injected into the encoder of the denoising network via scaling and shifting features, while the semantic information is injected into the decoder using multi-layer, spatially-adaptive normalization operators.

Encoder: The encoder of the network processes the noisy image with stacked semantic diffusion encoder resblocks and attention blocks. These resblocks consists of convolution, SiLU and group normalization. Where SiLU (Paul et al. 2022) is a nonlinearity of the form $f(x) = x \cdot \text{sigmoid}(x)$ which tends to work better than ReLU on deeper models. In order to inject the time step t at different time steps, the resblock involves scaling and shifting the intermediate activation with learnable weight $w(t) \in \mathbb{R}$ and bias $b(t) \in \mathbb{R}$ formulated as, $f_{i+1} = w(t) \cdot f_i + b(t)$ where $f_i, f_{i+1} \in \mathbb{R}$ are the input and output features.

Decoder: The semantic label map is injected into the decoder of the denoising network by the semantic diffusion decoder resblock in multi-layer spatially adaptive manner. Different from the resblocks in the encoder, here the spatially-adaptive normalization is used instead of the group normalization. This normalization layer injects the semantic label map into the denoising streams by regulating the feature in a spatially-adaptive, learnable transformation, which is formulated as follows,

$$f^{i+1} = \gamma^i(x) \cdot \text{Norm}\left(f^i\right) + \beta^i(x), \tag{5.31}$$

where f^i and f^{i+1} are the input and output features and Norm$(\cdot)$ refers to the



FID	17.12	16.98	17.01	**15.70**	16.74	19.45	22.48
Inception Score	2.47	2.48	2.48	2.68	2.70	**2.73**	2.72

Figure 5.2: **Guidance Scale Ablation:** For a given mask, we generate images using different values of the guidance scale, s. The FID and IS metrics are computed by generating images for all masks in the test set at 20× magnification.

parameter-free group normalization. $\gamma^i(x), \beta^i(x)$ are the spatially-adaptive weight and bias learned from the semantic layout, respectively.

In the NASDM model, the conditioning signal is constructed using the semantic mask such that each channel of the signal corresponds to a unique nuclei type. In addition, a mask comprising of the edges of all nuclei to further demarcate nuclei instances is also concatenated to the signal.

5.4 Experiments

In this section, we first describe our implementation details and training procedure. Further, we establish the robustness of our model by performing an ablative study over objective magnification and classifier-guidance scale. We then perform quantitative and qualitative assessments to demonstrate the efficacy of our nuclei-aware semantic histopathology generation model.

Table 5.1: **Quantitative Assessment:** We report the performance of our method using Fréchet Inception Distance (FID) and Inception Score (IS) with the metrics reported in existing works. (-) denotes that corresponding information was not reported in original work. *Note that performance reported for best competing method on the colon data is from our own implementation, performances for both this and our method should improve with better tuning. Please refer to our github repo for updated statistics.

Method	**Tissue type**	**Conditioning**	**FID(↓)**	**IS(↑)**
BigGAN (Brock, Donahue, and Simonyan 2018)	bladder	none	158.4	-
AttributeGAN (Ye et al. 2021)	bladder	attributes	53.6	-
ProGAN (Karras et al. 2017)	glioma	morphology	53.8	1.7
Morph-Diffusion (Moghadam et al. 2023)	glioma	morphology	20.1	2.1
Morph-Diffusion* (Moghadam et al. 2023)	colon	morphology	18.8	2.2
NASDM (Real Masks)	colon	nuclei mask	**14.1**	**2.7**
MaskGen + NASDM (Generated Masks)	colon	syn. nuclei mask	**15.2**	**2.6**

5.4.1 Implementation Details

Our diffusion models for patch and mask generation is implemented using a semantic UNet architecture (Section 5.3.11), trained using the objective in (5.30). Following previous works (A. Q. Nichol and Dhariwal 2021), we set the trade-off parameter λ as 0.001. We use the AdamW optimizer to train our model. Additionally, we adopt an exponential moving average (EMA) of the denoising network weights with 0.999 decay. Following DDPM (Ho, Jain, and Abbeel 2020), we set the total number of diffusion steps as 1000 and use a linear noising schedule with respect to timestep t for the forward process. After normal training with a learning rate of $1e-4$, we decay the learning rate to $2e-5$ to further finetune the model with a drop rate of 0.2 to enhance the classifier-free guidance capability during sampling. The whole framework is implemented using Pytorch and trained on 4 NVIDIA Tesla A100 GPUs with a batch-size of 40 per GPU. Code will be made public on publication or request.

5.4.2 Ablation over Guidance Scale

In this study, we test the effectiveness of the classifier-free guidance strategy. We consider the variant without guidance as our baseline. As seen in Figure 5.2, increase in guidance scale initially results in better image quality as more detail is added to visual structures of nuclei. However, with further increase, the image quality degrades as the model overemphasizes the nuclei and staining textures.

5.4.3 Ablation over Objective Magnification

As described in Section 5.3.7, we generate patches at two different objective magnifications of 10× and 20×. In this section, we contrast the generative performance of the models trained on these magnification levels respectively. From the table on right, we observe that the model trained at 20× objective magnification produces better generative metrics. Note that we only train on a subset on 20× mag. to keep the size of the training data constant.

Obj. Mag.	FID(↓)	IS(↑)
10×	38.1	2.3
20×	**20.7**	**2.5**

5.4.4 Generative Metrics Evaluation

To the best of our knowledge, ours is the only work that is able to synthesize histology images given a semantic mask, making a direct quantitative comparison tricky. However, the standard generative metric Fréchet Inception Distance (FID) measures the distance between distributions of generated and real images in the Inception-V3 (Kynkäänniemi et al. 2022) latent space, where a lower FID indicates that the model is able to generate images that are very similar to real data. Therefore, we

compare FID and IS metrics with the values reported in existing works (Ye et al. 2021; Moghadam et al. 2023) (ref. Table 5.1) in their own settings. We can observe that our method outperforms all existing methods including both GANs-based methods as well as the recently proposed morphology-focused generative diffusion model.

5.4.5 Downstream Task Evaluation

Tissue biopsy analysis is a critical aspect of histopathology, wherein anatomic pathologists examine hematoxylin and eosin-stained (H&E) biopsies to identify structural and cellular features associated with various diseases. The interpretation of these features not only aids in diagnosing diseases but also helps determine disease severity and guide treatment decisions. However, this process is inherently subjective and susceptible to inter-observer variability due to the investigating pathologists' varying levels of experience and exposure to different disease states. As such, the use of deep learning models for medical image segmentation has been of enormous interest in recent years and has also shown remarkable results. These models rely heavily on large and heterogeneous datasets with pixel-wise expert annotations to produce precise outcomes. The creation of such expert-annotated medical datasets remains a substantial barrier to the development of these models, as it is both labor-intensive and time-consuming. In order to address this challenge, there is a growing interest in generative modeling for medical imaging. However, the effectiveness of these synthetic datasets in addressing segmentation challenges is largely unexplored. In this study, our objective is twofold: (1) to generate synthetic tissue patches from annotated semantic masks using a nuclei-aware semantic diffusion model and (2) to train and evaluate nuclei segmentation models investigating the potential of synthetic data in enhancing downstream segmentation performance. In this section, we (1) describe

the datasets used for training and validating the NASDM and HoVerNet models, along with the steps for data preparation, (2) detail the process of nuclei semantics conditioned patch generation using NASDM, and (3) outline the training and evaluation of nuclei segmentation model for downstream experiments.

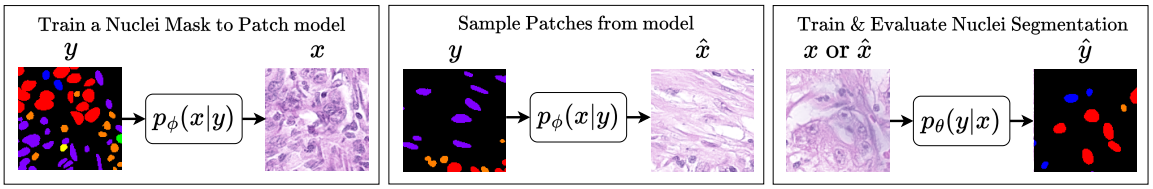


Figure 5.3: **Overall approach:** We have patches x sampled from conditional data distribution, $x \sim q(x \mid y)$, and masks y as the conditioning signal. We train a conditional generative model $p_\phi(x \mid y)$ (left), sample synthetic images (middle), and then evaluate the efficacy of synthetic images in training nuclei segmentation models $p_\theta(y \mid x)$ (right). Here ϕ and θ represent the parameters of NASDM and HoVerNet models.

In order to investigate the effectiveness of synthetic datasets in improving nuclei segmentation models, we perform three evaluation experiments: (1) Addition of Synthetic Patches (5.4.5) to the training of nuclei segmentation models. This experiment explores the impact of supplementing the training dataset of the HoVerNet model by adding synthetic images generated from our generative NASDM model. (2) Replacement with Synthetic Patches (5.4.5) for nuclei segmentation training. In this experiment, we evaluate the performance of segmentation models trained on datasets comprising different ratios of real to synthetic patches. And (3) Synthetic vs Real Patches (5.4.6) for training downstream nuclei segmentation models. This experiment determines how effective synthetically generated patches are for training nuclei segmentation models compared to their real counterparts. In all following experiments, the dice score is reported on a held-out real test set described in Table 5.2. The models are validated after every two epochs during training, and we report the metrics of the best-performing model on the test set.

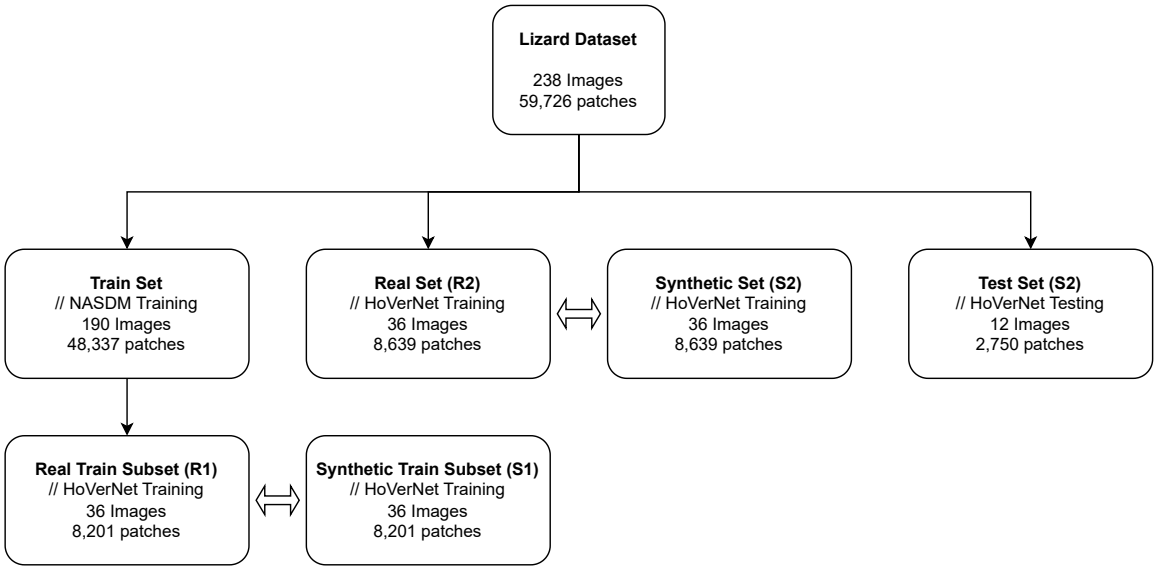


Figure 5.4: **Overview of data:** The figure describes the different subsets of Lizard dataset used for training and evaluation of NASDM and HoVerNet models in our experiments. Refer Table 5.2 for further details.

Dataset Setup

We use the publicly available Lizard dataset Graham, Jahanifar, et al. 2021, comprising histology image regions of colon tissue from six distinct sites. These tissue images, obtained at a 20× objective magnification, are annotated for epithelial cells, connective tissue cells, lymphocytes, plasma cells, neutrophils, and eosinophils. The dataset consists of 238 tissue images, with an average size of 1055×934 pixels. For computational viability, all the tissue images were divided into smaller image patches of 128×128 pixels at 20× objective magnification. Patching is performed with a 50% overlap in neighboring patches to ensure the information at the patch boundary is not lost. Patches with less than 50% tissue area were excluded from consideration. The tissue images yield a total of $59,726$ patches. We employ the structure-preserving color normalization approach Vahadane et al. 2016 to normalize the stain distribution of all patches. We harmonize the stain distribution with respect to a specifically

Table 5.2: **Overview of data:** Different subsets of Lizard dataset used for training and evaluation of NASDM and HoVerNet models.

Dataset	**ID**	**Type**	**NASDM Training**	**HoVerNet Training**	**# Images**	**# Patches**
Train Set	Train	Real	✓	✗	190	48,337
Real Train Subset	R1	Real	✓	✓	36	8,201
Synthetic Train Subset	S1	Synthetic	✗	✓	36	8,201
Real Set	R2	Real	✗	✓	36	8,639
Synthetic Set	S2	Synthetic	✗	✓	36	8,639
Test Set	Test	Real	✗	✗	12	2,750

chosen slide from the training dataset, mitigating the impact of staining variations on model performances.

We train the NASDM generative model on a Train Set containing 190 tissue images from the Lizard dataset tiled into $48,337$ patches. From this Train Set we select a subset R1 consisting of 36 images ($8,201$ patches) and generate a corresponding synthetic subset S1 from NASDM using R1's real nuclei masks. From the Lizard dataset, we select another subset R2 comprising of 36 images tiled into $8,639$ smaller patches. The images in R2 are not a part of the Train Set used for training the NASDM model. We also generate a synthetic set S2 using the nuclei masks of R2. Lastly, we reserve 12 images with $2,750$ patches, not included in any of the sets mentioned above, for testing the segmentation models trained in downstream experiments. Table 5.2 provides the details of the subsets of the dataset used in different tasks along with their designated nomenclature, number of images, and number of patches.

Synthetic Patch Generation Model

In order to generate patches given nuclei masks, we employ a nuclei-aware semantic diffusion model (NASDM) that generates hyper-realistic tissue patches conditioned

on semantic masks highlighting locations of six different types of nuclei. NASDM uses a conditional denoising diffusion probabilistic model which is trained to maximize the conditional likelihood of real data.

Nuclei Segmentation Model

In all our downstream experiments, we employ HoVerNet for nuclear segmentation. The training of HoVerNet is a two-stage process. In the initial stage, the model is initialized with pre-trained weights from the ImageNet dataset, and the decoder is trained exclusively for 50 epochs with a batch size of 16. In the second stage, all the layers are fine-tuned for another 50 epochs. In both stages, we train the model using Adam optimizer with an initial learning rate of 10^{-4} and then reduce it to a rate of 10^{-5} after 25 epochs. We use the best-performing model over the hundred epochs for testing. To assess the model's performance we compute the Dice score and Mean Intersection over Union (IoU) of the predicted masks with respect to their actual counterparts. Given ground truth annotation X and predicted annotation Y, the dice score and Mean IoU are defined as:

$$\text{Dice Score} = 2\,(X \cap Y) \div (|X| + |Y|) \tag{5.32}$$

$$\text{Mean IoU} = |X \cap Y| \div |X \cup Y| \tag{5.33}$$

The final metric is obtained by averaging scores first by channel and then by batch, providing a comprehensive evaluation of performance.

Table 5.3: **Addition of Synthetic Patches:** Segmentation performance of the nuclei segmentation model with consecutive augmenting of training set using synthetic data. We report the mean and standard deviation across three runs for both metrics.

Data	# Patches	Dice Score	Mean IoU
25% R2	2,159	0.7713 ± 0.0005	0.6409 ± 0.0008
25% R2 + 25% S2	4,318	0.7869 ± 0.0007	0.6608 ± 0.0012
25% R2 + 50% S2	6,478	0.7993 ± 0.0006	0.6771 ± 0.0005
25% R2 + 75% S2	8,639	$\mathbf{0.8092 \pm 0.0004}$	$\mathbf{0.6889 \pm 0.0007}$

Addition of Synthetic Patches

In this experiment, our objective is to assess the effectiveness of using synthetically generated data to augment existing datasets for nuclei segmentation tasks. Initially, we train a segmentation model exclusively on a 25% subset of R2. We then accumulate synthetic images from S2, which correspond to the masks of the remaining 75% of R2. We progressively incorporate subsets of these images from the synthetic image set S2 into the training. Note that the size of the training dataset increases with the addition of additional images. Also, note that we do not use all the images in the set R2 as the base set to make sure that the added images correspond to new masks that do not exist in the base training set we start with. The Dice scores and mean IoU of all the models on the Test set are presented in Table 5.3. We observe that as the training data is supplemented with synthetic patches, there is a discernible improvement in the model performance. This trend of gradual improvement underscores the beneficial impact of synthetically generated images in augmenting datasets and ultimately enhancing the accuracy of segmentation tasks.

With this experiment we intend to explores the impact of supplementing the training dataset of the HoVerNet model by adding synthetic images generated from our generative NASDM model. As can be seen in Table 5.3, the performance of the model

consistently improves with the addition of synthetic images in the training dataset of the nuclei segmentation model. Observations from this experiment support the contention that synthetic images generated using a state-of-the-art conditional diffusion model are already useful for augmenting existing expertly annotated datasets to improve the performances of downstream nuclei segmentation models trained on them. This demonstrates that augmenting real datasets with synthetic samples of rare disease subtypes and can help their identification and quantification in the wild.

Table 5.4: **Replacement with Synthetic Patches:** Performance of models trained on real data, synthetic data, and different combinations of both, given the same set of annotation masks. We report the mean and standard deviation across three runs for both metrics.

Data	**Dice Score**	**Mean IoU**
R2	0.8091 ± 0.0012	0.6890 ± 0.0014
75% R2 + 25% S2	$\mathbf{0.8098 \pm 0.0007}$	$\mathbf{0.6902 \pm 0.0009}$
50% R2 + 50% S2	0.8097 ± 0.0006	0.6898 ± 0.0007
25% R2 + 75% S2	0.8092 ± 0.0004	0.6889 ± 0.0007
S2	0.8087 ± 0.0004	0.6886 ± 0.0002

Replacement with Synthetic Patches

In this experiment, our goal is to compare the performance of a nuclei segmentation model when trained entirely on real data against when trained solely on synthetic data. For further clarity, we also evaluate models trained with combinations of real and synthetic data in different ratios while keeping the size of the dataset constant. The training utilized set R2 for real images and set S2 for synthetic images. Note that the set S2 is generated using the masks in the set R2. We first train a segmentation model on only R2 and then systematically replace a portion of patches in R2 with corresponding synthetic patches from S2, ensuring the total number of images and

the masks used stay the same. The Dice score and mean IoU on the Test Set are detailed in Table 5.4. Notably, performance across all runs is comparable indicating that there is no loss of performance on replacement with synthetic patches. This finding indicates that synthetic data performs just as effectively, if not better, in the training of nuclei segmentation problems.

Here, we evaluate the performance of segmentation models trained on datasets comprising different ratios of real to synthetic patches. Essentially, this experiment tests if synthetic patches are comparable to their real counterparts for the same nuclei mask for training a nuclei segmentation model. In this experiment, we progressively replace the real patches in the training dataset of the segmentation model with their synthetic counterparts generated conditionally using the generative model with their corresponding masks as the condition. This is done until the entire training dataset is made up of only synthetic images for the same masks as the real set we start with. As seen in Table 5.4, the performance of the trained nuclei segmentation stays unaffected by the replacement of real patches with their synthetic counterparts as models across different ratios in the training dataset perform comparably. This observation supports the hypothesis that synthetic images generated from state-of-the-art generative models are as effective for training nuclei segmentation models as real annotated images. This observation has tremendous implications as it demonstrates that once we can generate masks from scratch, an end-to-end generative model can be used to synthesize unlimited training data for training models for downstream tasks.

Table 5.5: **Comparison of Synthetic and Manual Annotations:** This table showcases the results of our investigation into the efficacy of annotations of synthetic patches generated by the NASDM model. We report the mean and standard deviation across three runs for both metrics.

Data	Dice Score	Mean IoU
R1	$\mathbf{0.8065 \pm 0.0005}$	$\mathbf{0.6854 \pm 0.0007}$
S1	0.8053 ± 0.0003	0.6840 ± 0.0003

5.4.6 Synthetic vs Manual Annotations

In this experiment, our objective is to assess whether synthetic patches generated using the same masks used for training the generative model yield a better and more precise set of annotations than the real patches themselves. This experiment tests the intuition that the generative model should be able to correct for manual errors between annotators and generate synthetic patches that are more consistent with the masks than their real counterparts. To test this hypothesis, we strategically select a subset of the training set used to train the NASDM model, denoted as R1, and generate synthetic patches based on their corresponding annotations, forming set S1. We employed both R1 and S1 to train the HoVerNet model independently. The rationale behind this approach was to evaluate whether the more precise annotations derived from the NASDM model could result in a more accurate representation of nuclei boundaries, thereby potentially yielding a superior Dice score or mean IoU. The results are reported in Table 5.5. The comparative analysis of performance in both cases revealed notable consistency. However, it is crucial to acknowledge that the models are evaluated using manually annotated patches in the test set. In this experiment, we intend to evaluate if this improvement in consistency with the mask leads to better segmentation models. We use a subset of the generative model's training data and generate synthetic patches for the masks in this set. A segmentation

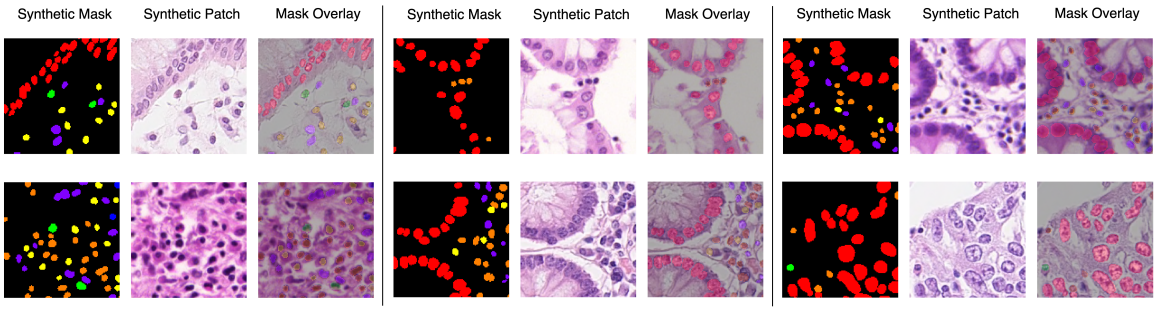


Figure 5.5: **Generation using synthetic masks:** We generate synthetic masks in different nuclei environments and these use these patches to generate synthetic tissue patches to demonstrate the proficiency of the model to generate realistic nuclei arrangements.

model is then trained on the real and the synthetic sets and evaluated on the test set. We observe that the models trained as such perform comparably with the model trained on the real set very slightly outperforming the other one. This highlights that the improvement in consistency with the mask that is observed qualitatively in the synthetic patches does not necessarily translate into better segmentation models.

5.4.7 Expert Evaluation

We have two expert pathologists review the synthetic patches generated using both real and synthetic masks as the condition. We use 60 patches for this review, 20 from the real set with their corresponding masks, 20 synthetic patches generated using real masks, 20 synthetic patches generated using synthetic masks from our mask generation model. The evaluation is performed on four criterion (Figure 5.7). First, The consistency of the patch with the corresponding mask where the experts are asked to evaluate if the patch and their corresponding masks match accurately in terms of the nuclei delineation. Second, if there are instances of unrecognizable nuclei types with respect to the annotated mask for each nuclei type in the patch.

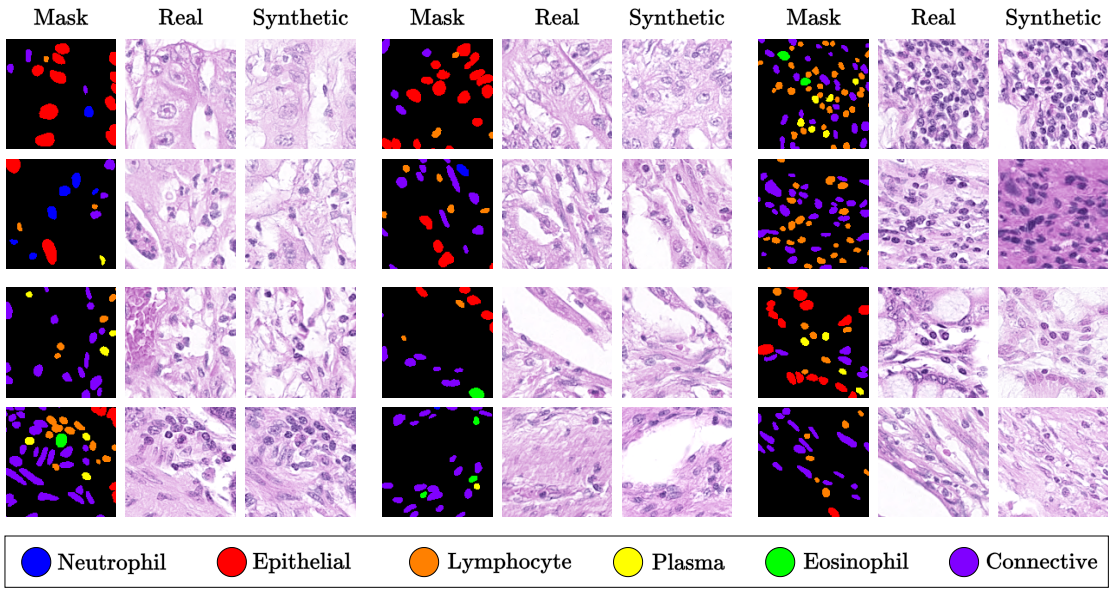


Figure 5.6: **Qualitative Results:** We generate synthetic images given masks with each type of nuclei in different environments to demonstrate the proficiency of the model to generate realistic nuclei arrangements. Legend at bottom denotes the mask color for each type of nuclei.

In this evaluation the experts are asked if the nuclei in the patch match their labels accurately. For each patch, the panel is asked to select the nuclei types that have at least one instance where they do not match the corresponding annotated label in the patch. The aim is to check if the model is able to generate convincing patterns in the synthetic patch for each type of nuclei. Third, if there are excess instances of nuclei in the mask with respect to the patch for each nuclei type, i.e. has the model failed to generate some nuclei that are present in the conditioning signal. The evaluators are asked to pick all nuclei types for which at least one instance is missing in the patch compared to the mask. The idea is to evaluate how effectively the model is able to generate all the nuclei instances that are present in the conditioning signal. Finally fourth, if there are excess instances of nuclei in the patch with respect to the mask for each nuclei type. Here, the panel is asked to select the nuclei types for which the model has generated an extra nuclei in the patch that is not present

in the conditioning signal. The idea here is to evaluate if the model is generating extra nuclei in the patch outside of the ones required by the mask it is conditioned on. Overall, the expert review demonstrates that our end-to-end histology patch generation method is able to synthesize patches that are reasonably comparable to real histology patches. The survey used for the review can be found on a public typeform survey[1]. In Figure. 5.6), we can see that the model is able to reasonably capture convincing visual structure for each type of nuclei.

5.5 Limitations

5.5.1 Evaluation based on patches

In this study, we emphasize that the expert evaluation presented is performed within a very controlled and specific setting, where evaluators are tasked with analyzing 128×128 pixel patches of Whole Slide Images (WSI). This approach significantly deviates from the typical method that pathologists use when interpreting histological tissues, which involves examining much larger regions of tissue at varying levels of magnification. The constrained focus on small image patches might limit the evaluators' ability to capture the broader context of tissue architecture and the relationships between different cellular structures, which could influence diagnostic accuracy. Therefore, the evaluation, as presented, should be interpreted within the confines of this artificial setup, and caution must be exercised when extending these results to more conventional histological evaluation scenarios.

[1] https://l7d0z1f5um1.typeform.com/to/IkAbnEOv

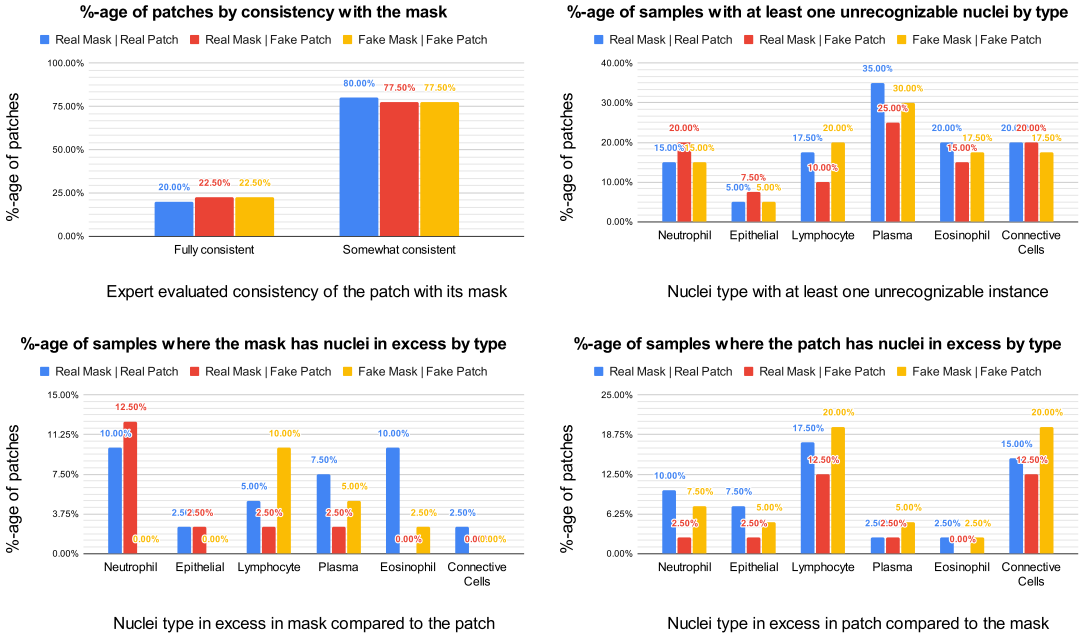


Figure 5.7: **Qualitative Review:** Compiled results from pathologist review. We have experts assess patches for, **top-left:** consistency of the patch with the corresponding mask, **top-right:** instances of unrecognizable nuclei types with respect to the annotated mask for each type in the patch, **bottom-left:** excess instances of nuclei in the mask with respect to the patch for each type, and **bottom-right:** excess instances of nuclei in the patch with respect to the mask for each type.

5.5.2 Scope of the evaluation

Moreover, the evaluation criteria were restricted to the four dimensions explicitly mentioned in the previous sections. Although these metrics are important for assessing certain aspects of model performance, they do not encompass the full complexity of histopathological diagnosis, which includes nuanced morphological patterns and clinical context. As a result, this limited scope may not fully capture the model's ability to generalize beyond these criteria, nor does it account for all possible ways in which the generated patches might be useful or flawed in broader medical practice. We recognize that there may be medically relevant patterns, particularly those

that pathologists rely on for nuanced diagnostic decisions, which the model does not replicate with high fidelity. These patterns may escape detection in the qualitative assessments made by our small panel of experts, especially given the relatively narrow set of criteria under consideration.

5.5.3 Size of the expert panel

It is also important to note that our panel consists of only two pathologists, which may introduce an element of subjectivity in the evaluation. While these experts bring considerable experience to their assessments, the small sample size of evaluators means that the results should be interpreted with caution. Different experts might have slightly different interpretations or levels of comfort in assessing the synthesized patches, which could potentially lead to variation in the evaluation outcomes if the panel were larger or more diverse. Therefore, any conclusions drawn from this evaluation need to be tempered with an understanding of this limitation.

5.5.4 Generalization to other tissue types

Furthermore, the model was specifically demonstrated on a dataset of colon tissue samples, and it remains uncertain whether the model's performance will hold when applied to other types of tissue or disease contexts. Histological structures and disease manifestations can vary widely between tissue types, and thus, the model's ability to generalize beyond the colon dataset should not be assumed. Future research should aim to extend the evaluation to include other datasets, encompassing a broader range of tissue types and pathologies, to better understand the generalizability and robustness of the model in diverse clinical scenarios.

5.5.5 Biases in the generated samples

One limitation inherent to generative models, and particularly relevant to this work, is the fact that they can only generate samples that resemble the data present in the training data. This means that the model is inherently biased toward replicating the patterns it has seen before, and it may struggle to generate plausible samples in the presence of novel or rare histological features not well represented in the training set. This replication of bias is a well-known issue with generative models, as they do not possess an intrinsic mechanism to correct for biases present in the training data. Consequently, while our model demonstrates a reasonable ability to capture and replicate the training data distribution, it must be recognized that any biases or limitations in the training data will likely propagate into the generated samples.

5.6 Future Work

5.6.1 Expanding conditional signals

In future work, it will be valuable to explore additional conditioning mechanisms that could improve the model's ability to generate more diverse and contextually accurate patches. For instance, conditioning the patch generation process on properties such as stain-distribution, tissue-type, disease-type, and other relevant clinical variables could allow the model to better capture the specific characteristics of different histological settings. By incorporating these properties into the generation process, the model could produce patches that are more reflective of the diverse range of tissue types and pathological conditions encountered in real-world practice. This would also enable the model to adapt to the unique characteristics of different staining protocols, which

vary between labs and can affect the appearance of histological samples.

5.6.2 Generating larger tissue areas

Furthermore, an interesting avenue for future research would be to explore the generation of patches conditioned on neighboring patches. This approach could enable the generation of larger tissue regions by stitching together individual patches, thus allowing for a more holistic representation of tissue architecture. By considering the spatial relationships between neighboring patches, the model could capture larger-scale tissue patterns that are critical for accurate histological analysis. This could open up new possibilities for the application of generative models in histopathology, enabling the synthesis of realistic tissue sections that can be used for various research and diagnostic purposes.

5.6.3 Addressing biases in future work

Finally, future studies could also consider further refining the model to address the biases present in the training data. Techniques such as domain adaptation, adversarial training, or incorporating real-world clinical feedback could be explored to mitigate these biases and ensure that the model produces more representative and equitable outputs across different tissue types and disease conditions.

Acknowledgements: We would like to thank Dr. Shyam Raghavan, M.D., Dr. Sahr Syed M.D., Dr. Ihab Lamzabi, M.D., Fisher Rhoads, B.S., and Dr. Lubaina Ehsan, M.D. for their invaluable inputs for our qualitative analysis.

6

Conclusion

In this thesis, I have explored the transformative potential of artificial intelligence in healthcare by developing tailored deep learning methods to address critical challenges in medical data analysis and diagnostics. My work focused on four key areas:

1. **Knowledge Distillation for Model Compression:** We created a flexible knowledge distillation approach that effectively compresses and transfers information from large vision models to smaller, domain-specific models. This method is optimized for resource-constrained settings and addresses the scarcity of annotated datasets in any domain. By leveraging the extensive knowledge of large vision models, we enabled efficient distillation into varied architectures using our novel mutual information maximization objectives.

2. **Information-Efficient Contrastive Learning:** We developed a vision-language alignment objective designed to learn effectively from paired medical images and textual data, even when annotated datasets are limited. Our method utilizes symbolic annotations from textual descriptions to improve vision models that extract visual patterns correlated with text. This approach achieves superior performance on standard benchmarks using significantly reduced amounts of data and smaller batch sizes, making it suitable for low-resource healthcare environments.

3. **Structure-Preserving Generative Adversarial Networks for Stain Normalization:** To address the issue of bias and reduced performance in neural network models due to variations in data collection and processing, we developed a novel adversarial approach for many-to-one domain stain normalization. Our custom training objective ensures the preservation of image structure during translation, enhancing the robustness and accuracy of downstream computational analyses. Comparative evaluations demonstrated superior performance in preserving structural integrity while transferring stain distributions across domains.

4. **Conditional Diffusion Models for Synthetic Medical Image Generation:** Recognizing the pervasive lack of data due to ethical and privacy concerns, we designed an end-to-end mechanism using conditional diffusion models to generate synthetic, hyper-realistic histopathological tissue slides. By synthesizing tissue patches conditioned on nuclei masks, this approach addresses data scarcity and privacy issues, aiding both deep learning systems and human pathologists. The generated synthetic datasets can mitigate dataset imbalances, reduce model training bias, and facilitate medical data sharing without compromising patient privacy.

By integrating these methodologies, this thesis presents innovative solutions to long-standing challenges in medical imaging and diagnostics. The tailored machine learning techniques developed herein have the potential to revolutionize healthcare practices by improving diagnostic accuracy, enhancing patient care, and optimizing treatment strategies. Our work underscores the critical role of specialized AI approaches in addressing the intricate challenges intrinsic to the healthcare domain.

Closing Remarks

The convergence of advanced machine learning techniques and healthcare has the potential to significantly impact patient outcomes and the overall efficiency of medical services. By focusing on the development of specialized AI methods tailored to the unique challenges of the healthcare domain, this thesis contributes to the foundational work necessary for the next generation of intelligent healthcare solutions. Continued interdisciplinary collaboration will be essential to realize the full potential of these technologies in clinical practice.

Ph.D. Timeline

1. **Qualifying Examination Proposal**: Fall 2021
2. **SAASN published at ICPR**: Fall 2021
3. **Qualifying Examination**: Spring 2022
4. **MIMKD published at CVPR Workshop**: Fall 2022
5. **CLIP-Lite published at AISTATS**: Spring 2023
6. **NASDM published at MICCAI**: Fall 2023
7. **Co-instructor for Geometry of Data**: Fall 2023
8. **Dissertation Proposal**: Fall 2023
9. **NASDM downstream evaluations**: Spring 2024
10. **Co-authored book chapter for Springer**: Spring 2024

11. **End-to-end annotated histopathology generation**: Fall 2024

12. **Ph.D. Dissertation Defence**: Fall 2024

Bibliography

Polyak, Boris T (1964). "Some methods of speeding up the convergence of iteration methods". In: *Ussr computational mathematics and mathematical physics* 4.5, pp. 1–17.

Donsker, Monroe D and SR Srinivasa Varadhan (1983). "Asymptotic evaluation of certain Markov process expectations for large time. IV". In: *Communications on Pure and Applied Mathematics* 36.2, pp. 183–212.

Reinhard, Erik et al. (2001). "Color transfer between images". In: *IEEE Computer graphics and applications* 21.5, pp. 34–41.

Paninski, Liam (2003). "Estimation of entropy and mutual information". In: *Neural computation* 15.6, pp. 1191–1253.

Wang, Zhou et al. (2004). "Image quality assessment: from error visibility to structural similarity". In: *IEEE transactions on image processing* 13.4, pp. 600–612.

Buciluǎ, Cristian, Rich Caruana, and Alexandru Niculescu-Mizil (2006). "Model compression". In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 535–541.

Coltuc, Dinu, Philippe Bolon, and J-M Chassery (2006). "Exact histogram specification". In: *IEEE Transactions on Image Processing* 15.5, pp. 1143–1152.

Hinton, Geoffrey E and Ruslan R Salakhutdinov (2006). "Reducing the dimensionality of data with neural networks". In: *science* 313.5786, pp. 504–507.

Quattoni, Ariadna, Michael Collins, and Trevor Darrell (2007). "Learning visual representations using images with captions". In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 1–8.

Fischer, Andrew H et al. (2008). "Hematoxylin and eosin staining of tissue and cell sections". In: *Cold Spring Harbor Protocols* 2008.5, pdb–prot4986.

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

Krizhevsky, Alex, Geoffrey Hinton, et al. (2009). "Learning multiple layers of features from tiny images". In.

Macenko, Marc et al. (2009). "A method for normalizing histology slides for quantitative analysis". In: *2009 IEEE international symposium on biomedical imaging: from nano to macro*. IEEE, pp. 1107–1110.

Everingham, Mark et al. (2010). "The pascal visual object classes (voc) challenge". In: *International journal of computer vision* 88.2, pp. 303–338.

Gutmann, Michael and Aapo Hyvärinen (2010). "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings, pp. 297–304.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25, pp. 1097–1105.

Ruderman, Avraham et al. (2012). "Tighter variational representations of f-divergences via restriction to probability measures". In: *arXiv preprint arXiv:1206.4664*.

Kingma, Diederik P and Max Welling (2013). "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114*.

Mikolov, Tomas et al. (2013). "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781*.

Sutskever, Ilya et al. (2013). "On the importance of initialization and momentum in deep learning". In: *International conference on machine learning*. PMLR, pp. 1139–1147.

Bejnordi, Babak Ehteshami et al. (2014). "Quantitative analysis of stain variability in histology slides and an algorithm for standardization". In: *Medical Imaging 2014: Digital Pathology*. Vol. 9041. SPIE, pp. 45–51.

Girshick, Ross et al. (2014). "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587.

Goodfellow, Ian et al. (2014). "Generative adversarial nets". In: *Advances in neural information processing systems*, pp. 2672–2680.

Khan, Adnan Mujahid et al. (2014). "A nonlinear mapping approach to stain normalization in digital histopathology images using image-specific color deconvolution". In: *IEEE Transactions on Biomedical Engineering* 61.6, pp. 1729–1738.

Kingma, Diederik P and Jimmy Ba (2014a). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

— (2014b). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

Romero, Adriana et al. (2014). "Fitnets: Hints for thin deep nets". In: *arXiv preprint arXiv:1412.6550*.

Simonyan, Karen and Andrew Zisserman (2014). "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556*.

Young, Peter et al. (2014). "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions". In: *Transactions of the Association for Computational Linguistics* 2, pp. 67–78.

Zeiler, Matthew D and Rob Fergus (2014). "Visualizing and understanding convolutional networks". In: *European conference on computer vision*. Springer, pp. 818–833.

Antol, Stanislaw et al. (2015). "Vqa: Visual question answering". In: *Proceedings of the IEEE international conference on computer vision*, pp. 2425–2433.

Chen, Xinlei, Hao Fang, et al. (2015). "Microsoft coco captions: Data collection and evaluation server". In: *arXiv preprint arXiv:1504.00325*.

Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean (2015). "Distilling the knowledge in a neural network". In: *arXiv preprint arXiv:1503.02531*.

Ioffe, Sergey and Christian Szegedy (2015a). "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *International conference on machine learning*. PMLR, pp. 448–456.

— (2015b). "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *arXiv preprint arXiv:1502.03167*.

Lei Ba, Jimmy, Kevin Swersky, Sanja Fidler, et al. (2015). "Predicting deep zero-shot convolutional neural networks using textual descriptions". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4247–4255.

Long, Jonathan, Evan Shelhamer, and Trevor Darrell (2015). "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer, pp. 234–241.

Russakovsky, Olga et al. (2015). "Imagenet large scale visual recognition challenge". In: *International journal of computer vision* 115.3, pp. 211–252.

Tishby, Naftali and Noga Zaslavsky (2015). "Deep learning and the information bottleneck principle". In: *2015 IEEE Information Theory Workshop (ITW)*. IEEE, pp. 1–5.

Vinyals, Oriol et al. (2015). "Show and tell: A neural image caption generator". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3156–3164.

Bolukbasi, Tolga et al. (2016). "Man is to computer programmer as woman is to homemaker? debiasing word embeddings". In: *Advances in neural information processing systems* 29, pp. 4349–4357.

He, Kaiming, Xiangyu Zhang, et al. (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Joulin, Armand et al. (2016). "Learning visual features from large weakly supervised data". In: *European Conference on Computer Vision*. Springer, pp. 67–84.

Loshchilov, Ilya and Frank Hutter (2016). "Sgdr: Stochastic gradient descent with warm restarts". In: *arXiv preprint arXiv:1608.03983*.

Nowozin, Sebastian, Botond Cseke, and Ryota Tomioka (2016). "f-gan: Training generative neural samplers using variational divergence minimization". In: *arXiv preprint arXiv:1606.00709*.

Parikh, Ankur P et al. (2016). "A decomposable attention model for natural language inference". In: *arXiv preprint arXiv:1606.01933*.

Pathak, Deepak et al. (2016). "Context encoders: Feature learning by inpainting". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2536–2544.

Sau, Bharat Bhusan and Vineeth N Balasubramanian (2016). "Deep model compression: Distilling knowledge from noisy teachers". In: *arXiv preprint arXiv:1610.09650*.

Taigman, Yaniv, Adam Polyak, and Lior Wolf (2016). "Unsupervised cross-domain image generation". In: *arXiv preprint arXiv:1611.02200*.

Ulyanov, Dmitry, Andrea Vedaldi, and Victor Lempitsky (2016). "Instance normalization: The missing ingredient for fast stylization". In: *arXiv preprint arXiv:1607.08022*.

Vahadane, Abhishek et al. (2016). "Structure-preserving color normalization and sparse stain separation for histological images". In: *IEEE transactions on medical imaging* 35.8, pp. 1962–1971.

Wang, Xiaolong and Abhinav Gupta (2016). "Generative image modeling using style and structure adversarial networks". In: *European Conference on Computer Vision*. Springer, pp. 318–335.

Yoo, Donggeun et al. (2016). "Pixel-level domain transfer". In: *European Conference on Computer Vision*. Springer, pp. 517–532.

Yu, Licheng et al. (2016). "Modeling context in referring expressions". In: *European Conference on Computer Vision*. Springer, pp. 69–85.

Zagoruyko, Sergey and Nikos Komodakis (2016a). "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer". In: *arXiv preprint arXiv:1612.03928*.

— (2016b). "Wide residual networks". In: *arXiv preprint arXiv:1605.07146*.

Zhu, Yuke et al. (2016). "Visual7w: Grounded question answering in images". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4995–5004.

Brakel, Philemon and Yoshua Bengio (2017). "Learning independent features with adversarial nets for non-linear ica". In: *arXiv preprint arXiv:1710.05050*.

Isola, Phillip et al. (2017). "Image-to-image translation with conditional adversarial networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134.

Janowczyk, Andrew, Ajay Basavanhally, and Anant Madabhushi (2017). "Stain normalization using sparse autoencoders (StaNoSA): Application to digital pathology". In: *Computerized Medical Imaging and Graphics* 57, pp. 50–61.

Karras, Tero et al. (2017). "Progressive growing of gans for improved quality, stability, and variation". In: *arXiv preprint arXiv:1710.10196*.

Kim, Taeksoo et al. (2017). "Learning to discover cross-domain relations with generative adversarial networks". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, pp. 1857–1865.

Li, Ang et al. (2017). "Learning visual n-grams from web data". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4183–4192.

Litjens, Geert et al. (2017). "A survey on deep learning in medical image analysis". In: *Medical image analysis* 42, pp. 60–88.

Liu, Ming-Yu, Thomas Breuel, and Jan Kautz (2017). "Unsupervised image-to-image translation networks". In: *Advances in neural information processing systems*, pp. 700–708.

Liu, Yun et al. (2017). *Detecting Cancer Metastases on Gigapixel Pathology Images*. Tech. rep. arXiv. URL: https://arxiv.org/abs/1703.02442.

Mao, Xudong et al. (2017). "Least squares generative adversarial networks". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2794–2802.

Selvaraju, Ramprasaath R et al. (2017). "Grad-cam: Visual explanations from deep networks via gradient-based localization". In: *Proceedings of the IEEE international conference on computer vision*, pp. 618–626.

Shrivastava, Ashish et al. (2017). "Learning from simulated and unsupervised images through adversarial training". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2107–2116.

Vaswani, Ashish et al. (2017). "Attention is all you need". In: *Advances in neural information processing systems*, pp. 5998–6008.

Yi, Zili et al. (2017). "Dualgan: Unsupervised dual learning for image-to-image translation". In: *Proceedings of the IEEE international conference on computer vision*, pp. 2849–2857.

Yim, Junho et al. (2017). "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4133–4141.

Zhao, Jieyu et al. (2017). "Men also like shopping: Reducing gender bias amplification using corpus-level constraints". In: *arXiv preprint arXiv:1707.09457*.

Zhu, Jun-Yan et al. (2017). "Unpaired image-to-image translation using cycle-consistent adversarial networks". In: *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232.

Belghazi, Mohamed Ishmael et al. (2018). "Mutual information neural estimation". In: *International Conference on Machine Learning*. PMLR, pp. 531–540.

Brock, Andrew, Jeff Donahue, and Karen Simonyan (2018). "Large scale GAN training for high fidelity natural image synthesis". In: *arXiv preprint arXiv:1809.11096*.

Devlin, Jacob et al. (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805*.

Hjelm, R Devon et al. (2018). "Learning deep representations by mutual information estimation and maximization". In: *arXiv preprint arXiv:1808.06670*.

Huang, Zehao and Naiyan Wang (2018). "Data-driven sparse structure selection for deep neural networks". In: *Proceedings of the European conference on computer vision (ECCV)*, pp. 304–320.

Miyato, Takeru et al. (2018). "Spectral normalization for generative adversarial networks". In: *arXiv preprint arXiv:1802.05957*.

Oord, Aaron van den, Yazhe Li, and Oriol Vinyals (2018). "Representation learning with contrastive predictive coding". In: *arXiv preprint arXiv:1807.03748*.

Roy, Santanu et al. (2018). "A study about color normalization methods for histopathology images". In: *Micron* 114, pp. 42–61.

Sandler, Mark et al. (2018). "Mobilenetv2: Inverted residuals and linear bottlenecks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520.

Stratos, Karl (2018). "Mutual information maximization for simple and accurate part-of-speech induction". In: *arXiv preprint arXiv:1804.07849*.

Van Horn, Grant et al. (2018). "The inaturalist species classification and detection dataset". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8769–8778.

Veličković, Petar et al. (2018). "Deep graph infomax". In: *arXiv preprint arXiv:1809.10341*.

Wang, Xiaolong, Ross Girshick, et al. (2018). "Non-local neural networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7794–7803.

Wu, Zhirong et al. (2018). "Unsupervised feature learning via non-parametric instance discrimination". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3733–3742.

Zhang, Han et al. (2018). "Self-attention generative adversarial networks". In: *arXiv preprint arXiv:1805.08318*.

Zhang, Xiangyu et al. (2018). "Shufflenet: An extremely efficient convolutional neural network for mobile devices". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6848–6856.

Ahn, Sungsoo et al. (2019). "Variational information distillation for knowledge transfer". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9163–9171.

Bachman, Philip, R Devon Hjelm, and William Buchwalter (2019). "Learning representations by maximizing mutual information across views". In: *arXiv preprint arXiv:1906.00910*.

Goyal, Priya et al. (2019). "Scaling and benchmarking self-supervised visual representation learning". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6391–6400.

Graham, Simon, Quoc Dang Vu, et al. (2019). "Hover-net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images". In: *Medical Image Analysis* 58, p. 101563.

Park, Taesung et al. (2019). "Semantic image synthesis with spatially-adaptive normalization". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2337–2346.

Peng, Baoyun et al. (2019). "Correlation congruence for knowledge distillation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5007–5016.

Recht, Benjamin et al. (2019). "Do imagenet classifiers generalize to imagenet?" In: *International Conference on Machine Learning*. PMLR, pp. 5389–5400.

Shaban, M Tarek et al. (2019). "Staingan: Stain style transfer for digital histological images". In: *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*. IEEE, pp. 953–956.

Shrivastava, Aman, Karan Kant, et al. (2019). "Deep Learning for Visual Recognition of Environmental Enteropathy and Celiac Disease". In: *arXiv preprint arXiv:1908.03272*.

Tian, Yonglong, Dilip Krishnan, and Phillip Isola (2019). "Contrastive representation distillation". In: *arXiv preprint arXiv:1910.10699*.

Tung, Frederick and Greg Mori (2019). "Similarity-preserving knowledge distillation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1365–1374.

Wang, Tianlu et al. (2019). "Balanced datasets are not enough: Estimating and mitigating gender bias in deep image representations". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5310–5319.

Wei, Jason W et al. (2019). "Automated detection of celiac disease on duodenal biopsy slides: A deep learning approach". In: *Journal of pathology informatics* 10.

Zhang, Michael R et al. (2019). "Lookahead optimizer: k steps forward, 1 step back". In: *arXiv preprint arXiv:1907.08610*.

Caron, Mathilde et al. (2020). "Unsupervised learning of visual features by contrasting cluster assignments". In: *arXiv preprint arXiv:2006.09882*.

Chen, Ting et al. (2020). "A simple framework for contrastive learning of visual representations". In: *International conference on machine learning*. PMLR, pp. 1597–1607.

Chen, Xinlei, Haoqi Fan, et al. (2020). "Improved baselines with momentum contrastive learning". In: *arXiv preprint arXiv:2003.04297*.

Grill, Jean-Bastien et al. (2020). "Bootstrap your own latent: A new approach to self-supervised learning". In: *arXiv preprint arXiv:2006.07733*.

He, Kaiming, Haoqi Fan, et al. (2020). "Momentum contrast for unsupervised visual representation learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738.

Ho, Jonathan, Ajay Jain, and Pieter Abbeel (2020). "Denoising diffusion probabilistic models". In: *Advances in Neural Information Processing Systems* 33, pp. 6840–6851.

Jabri, Allan, Andrew Owens, and Alexei A Efros (2020). "Space-time correspondence as a contrastive random walk". In: *arXiv preprint arXiv:2006.14613*.

Levine, Adrian B et al. (2020). "Synthesis of diagnostic quality cancer pathology images by generative adversarial networks". In: *The Journal of pathology* 252.2, pp. 178–188.

Liang, Paul Pu et al. (2020). "Towards debiasing sentence representations". In: *arXiv preprint arXiv:2007.08100*.

McAllester, David and Karl Stratos (2020). "Formal limitations on the measurement of mutual information". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 875–884.

Sariyildiz, Mert Bulent, Julien Perez, and Diane Larlus (2020). "Learning visual representations with caption annotations". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*. Springer, pp. 153–170.

Song, Yang et al. (2020). "Score-based generative modeling through stochastic differential equations". In: *arXiv preprint arXiv:2011.13456*.

Stroud, Jonathan C et al. (2020). "Learning video representations from textual web supervision". In: *arXiv preprint arXiv:2007.14937*.

Tian, Yonglong, Chen Sun, et al. (2020). "What makes for good views for contrastive learning?" In: *arXiv preprint arXiv:2005.10243*.

Wang, Zeyu et al. (2020). "Towards fairness in visual recognition: Effective strategies for bias mitigation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8919–8928.

Wu, Mike et al. (2020). "On mutual information in contrastive learning for visual representations". In: *arXiv preprint arXiv:2005.13149*.

Xie, Lipeng et al. (2020). "Integrating deep convolutional neural networks with marker-controlled watershed for overlapping nuclei segmentation in histopathology images". In: *Neurocomputing* 376, pp. 166–179.

Desai, Karan and Justin Johnson (2021). "Virtex: Learning visual representations from textual annotations". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11162–11173.

Dhariwal, Prafulla and Alexander Nichol (2021). "Diffusion models beat gans on image synthesis". In: *Advances in Neural Information Processing Systems* 34, pp. 8780–8794.

Fajardo, Val Andrei et al. (2021). "On oversampling imbalanced data with deep conditional generative models". In: *Expert Systems with Applications* 169, p. 114463.

Graham, Simon, Mostafa Jahanifar, et al. (2021). "Lizard: A large-scale dataset for colonic nuclear instance segmentation and classification". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 684–693.

Hendrycks, Dan et al. (2021). "Natural adversarial examples". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15262–15271.

Nichol, Alexander Quinn and Prafulla Dhariwal (2021). "Improved denoising diffusion probabilistic models". In: *International Conference on Machine Learning*. PMLR, pp. 8162–8171.

Radford, Alec et al. (2021). "Learning transferable visual models from natural language supervision". In: *arXiv preprint arXiv:2103.00020*.

Tan, Zhentao, Menglei Chai, et al. (2021). "Diverse semantic image synthesis via probability distribution modeling". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7962–7971.

Tan, Zhentao, Dongdong Chen, et al. (2021). "Efficient semantic image synthesis via class-adaptive normalization". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.9, pp. 4852–4866.

Van der Laak, Jeroen, Geert Litjens, and Francesco Ciompi (2021). "Deep learning in histopathology: the path to the clinic". In: *Nature medicine* 27.5, pp. 775–784.

Watson, Daniel et al. (2021). "Learning to efficiently sample from diffusion probabilistic models". In: *arXiv preprint arXiv:2106.03802*.

Xiao, Zhisheng, Karsten Kreis, and Arash Vahdat (2021). "Tackling the generative learning trilemma with denoising diffusion GANs". In: *arXiv preprint arXiv:2112.07804*.

Xue, Yuan et al. (2021). "Selective synthetic augmentation with HistoGAN for improved histopathology image classification". In: *Medical image analysis* 67, p. 101816.

Ye, Jiarong et al. (2021). "A multi-attribute controllable generative model for histopathology image synthesis". In: *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part VIII 24*. Springer, pp. 613–623.

Hall, Melissa et al. (2022). "A systematic study of bias amplification". In: *arXiv preprint arXiv:2201.11706*.

Ho, Jonathan, Chitwan Saharia, et al. (2022). "Cascaded Diffusion Models for High Fidelity Image Generation." In: *J. Mach. Learn. Res.* 23.47, pp. 1–33.

Ho, Jonathan and Tim Salimans (2022). "Classifier-free diffusion guidance". In: *arXiv preprint arXiv:2207.12598*.

Kawar, Bahjat et al. (2022). "Denoising diffusion restoration models". In: *arXiv preprint arXiv:2201.11793*.

Kynkäänniemi, Tuomas et al. (2022). "The Role of ImageNet Classes in Fr\'echet Inception Distance". In: *arXiv preprint arXiv:2203.06026*.

Paul, Ashis et al. (2022). "SinLU: sinu-sigmoidal linear unit". In: *Mathematics* 10.3, p. 337.

Saharia, Chitwan, William Chan, et al. (2022). "Palette: Image-to-image diffusion models". In: *ACM SIGGRAPH 2022 Conference Proceedings*, pp. 1–10.

Saharia, Chitwan, Jonathan Ho, et al. (2022). "Image super-resolution via iterative refinement". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Wang, Weilun et al. (2022). "Semantic image synthesis via diffusion models". In: *arXiv preprint arXiv:2207.00050*.

Wu, Yawen et al. (2022). "Recent advances of deep learning for computational histopathology: Principles and applications". In: *Cancers* 14.5, p. 1199.

Zhou, Qifan and Hujun Yin (2022). "A u-net based progressive gan for microscopic image augmentation". In: *Medical Image Understanding and Analysis: 26th Annual Conference, MIUA 2022, Cambridge, UK, July 27–29, 2022, Proceedings*. Springer, pp. 458–468.

Moghadam, Puria Azadi et al. (2023). "A morphology focused diffusion probabilistic model for synthesis of histopathology images". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2000–2009.

Shrivastava, Aman and P Thomas Fletcher (2023). "Nasdm: Nuclei-aware semantic histopathology image generation using diffusion models". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 786–796.

Appendices

Appendix A

Estimating and Maximizing Mutual Information for Knowledge Distillation

A.1 Limitations and Broader Impacts

In this paper, we presented a novel Mutual Information Maximization based knowledge distillation framework (MIMKD). Our method uses the JSD based lower-bound on mutual information which is optimized using only one negative sample. However, despite its favorable properties, our lower-bound may be less tight on the mutual information than the infoNCE bound as it approximates the mutual information by being monotonically related with it. Additionally, as we use only one negative sample, the performance of the method may be hindered by the presence of false negatives. The performance of the method is also effected by the architecture of the discriminator functions which can be explored further. We presented three information maximization formulations and demonstrated the value of region-consistent information maximization on distillation performance. We observe that the performance is slightly-sensitive to the hyper-parameters that control the relative value of our global, local, and feature information maximization formulations. This has

been explored in great detail in our ablation sections and further demonstrated in figures A.1, A.2, and A.3. Our method transfers representations from the teacher to the student. As such, harmful biases that the teacher has learnt are transferred to the student as well. And further exploration is required to alleviate the transfer of such biases during distillation.

A.2 Hyper-parameters for other methods

The student is trained with the following loss function which is a combination of the distillation loss and the cross-entropy loss for classification:

$$\mathcal{L} = \alpha\mathcal{L}_{cls} + (1-\alpha)\mathcal{L}_{KD} + \beta\mathcal{L}_{dis} \tag{A.1}$$

Note that we set $\alpha = 1$ for all methods except KD ßG. Hinton, Vinyals, and Dean 2015 and the value of β is set to the value recommended in the original work as follows:

1. KD G. Hinton, Vinyals, and Dean 2015: $\alpha = 0.9, \beta = 0$
2. Fitnet Romero et al. 2014: $\beta = 100$
3. AT Zagoruyko and Komodakis 2016a: $\beta = 1000$
4. VID Ahn et al. 2019: $\beta = 1$
5. CRD Tian, Krishnan, and Isola 2019: $\beta = 0.8$, for CRD evaluation, we use a original work inspired self-implementation with 4096 negative samples and $i \neq j$ negative sampling methodology as described in the original work.

A.3 Pairing Intermediate Representations

A.3.1 Similar CNN Architectures.

Consider the case of distillation when the teacher network is a pre-trained WRN-40-2 and the student network is a WRN-16-1. We use 4 same-sized representations extracted from intermediate layers of the networks. Therefore, the set $\mathcal{R} = \{(f_t^{(k)}(x), f_s^{(k)}(x))\}_{k=1}^{K}$ contains k pairs of same-sized 2-dimensional representations. Table A.1 describes the sizes of the intermediate representations used for feature-based mutual information maximization. It can be seen that for this combination we use $k = 4$ in our formulation.

Table A.1: Dimensions of intermediate representation in the form $channels \times height \times width$ used for feature-level mutual information maximization between a teacher WRN-40-2 and a student WRN-16-1 network. Alternatively, each value of k represents a pair of elements in the set $\mathcal{R}$.

	WRN-40-2	WRN-16-1
k	$f_t^{(k)}(x)$	$f_s^{(k)}(x)$
1	$16 \times 32 \times 32$	$16 \times 32 \times 32$
2	$32 \times 32 \times 32$	$16 \times 32 \times 32$
3	$64 \times 16 \times 16$	$32 \times 16 \times 16$
4	$128 \times 8 \times 8$	$64 \times 8 \times 8$

A.3.2 Dissimilar CNN Architectures.

Similar approach of defining the set $\mathcal{R}$ is followed in cases where the teacher and student networks have significantly different architectures. For instance, Table A.2 shows the dimensions of intermediate representations used when the teacher network is a ResNet34 while the student is a ShuffleNetV2. Here $k = 4$ is used, however, for

some combinations of different standard architectures we use $k = 3$ if only 3 pairs intermediate representations from the teacher and the student have the same size. Note that our method is invariant to the number of channels in the representations. Therefore, mismatch in the number of channels in pairs of representations in $\mathcal{R}$ is inconsequential for the formulation of our losses.

Table A.2: Dimensions of intermediate representation in the form $channels \times height \times width$ used for feature-level mutual information maximization between a teacher WRN-40-2 and a student WRN-16-1 network. Alternatively, each value of k represents a pair of elements in the set $\mathcal{R}$.

	ResNet34	ShuffleNetV2
k	$f_t^{(k)}(x)$	$f_s^{(k)}(x)$
1	$64 \times 32 \times 32$	$24 \times 32 \times 32$
2	$512 \times 16 \times 16$	$116 \times 16 \times 16$
3	$1024 \times 8 \times 8$	$232 \times 8 \times 8$
4	$2048 \times 4 \times 4$	$464 \times 4 \times 4$

A.4 Mutual Information Discriminators

The parameterized mutual information discriminator functions (T_{ω_g}, T_{ω_l}, and T_{ω_f}) can be modeled as neural networks. In our experiments, we use two distinct discriminator architectures inspired from the functions presented in Deep InfoMax Hjelm et al. 2018.

A.4.1 Convolve Architecture.

In this method, the representations from the teacher and the student are concatenated together and passed through a series of layers to get the score. For global information maximization, the final representations from both networks is concatenated together

to get $[f_s(x), f_t(x)]$. This vector is then passed to a fully connected network with two 512-unit hidden layers, each followed by a *ReLU* non-linearity (ref. table A.3). The output is then passed through another linear layer to obtain the final score.

Table A.3: The architecture of the discriminator used for global information maximization. Here LL denotes Linear Layer and $d(v)$ refers to the number of dimensions in vector v.

Input	Operation	Output
$[f_t(x), f_s(x)]$	LL + ReLU	O_1
O_1	LL + ReLU	O_2
O_2	LL	score

For local information maximization, we replicate the final representation from the teacher $f_t(x)$ to match the $m_K \times m_K$ size of the student's last intermediate feature map ($f_s^{(K)}(x)$). The resulting replicated tensor is then concatenated with $f_s^{(K)}(x)$ to get $[f_t(x), f_s^{(K)}(x)]$ which serves as the input for the critic function (ref. table A.4).

Table A.4: The architecture of the discriminator used for local and feature mutual information maximization. Note that for feature mutual information maximization the input at the first layer is $[f_t^{(k)}(x), f_s^{(k)}(x)]$.

Input	Operation	Output
$[f_t(x), f_s^{(K)}(x)]$	1×1 Conv + ReLU	O_1
O_1	1×1 Conv + ReLU	O_2
O_2	1×1 Conv	scores

Similarly, consider feature mutual information maximization, for each pair in the set $\mathcal{R}$ we use a distinct discriminator $T_{\omega_f}^{(k)}$. For a given k, each pair of intermediate feature representations in the set $\mathcal{R}$ are concatenated together to get $[f_t^{(k)}(x), f_s^{(k)}(x)]$. Which is then passed through two convolutional (1×1 kernels and 512 filters) where each layer is followed by a ReLU non-linearity. The output obtained is then further passed

into a convolutional layer (1×1 kernels and 1 filter) to give $m_k \times m_k$ scores (ref. table A.4).

A.4.2 Project and Dot Architecture.

In this method, the representations from both the teacher and the student are first projected using an appropriate projection architecture with a linear shortcut. The dot-product of these projections is then computed to get the score. Positive and negative pairs of representations are passed through the discriminator to get respective scores to be passed into equation (2) to get the estimates on the lower bound of the mutual information. One-dimensional representations are projected using the architecture described in table A.5, whereas for two-dimensional intermediate feature maps, projection architecture described in table A.6 is used.

Table A.5: The projection architecture used for one-dimensional inputs. Here, LL denotes linear layer while LN denotes layer normalization. Both $f_t(x)$ and $f_s(x)$ are projected using this architecture and their dot product is computed to get scores.

Input	Operation	Output
$f_t(x)$ or $f_s(x)$	LL + ReLU + LL	O_1
$f_t(x)$ or $f_s(x)$	LL + ReLU	O_2
O_1 + O_2	LN	*proj*

Therefore, for (1) global information maximization, both $f_t(x)$ and $f_s(x)$ are projected using the one-dimensional projection architecture, for (2) local information maximization, the final teacher representation, $f_t(x)$, is projected using the one-dimensional projection architecture and duplicated to match the size of the projected intermediate student representation projected using the two-dimensional projection architecture, a dot product of these outputs is then computed to get the scores, while for (3) feature

information maximization, both representations in each pair of the set $\mathcal{R}$ is projected using a respective two-dimensional projection architecture.

Table A.6: The projection architecture used for two-dimensional inputs. Here, LL denotes linear layer while LN denotes layer normalization.

Input	Operation	Output
$f_s^{(k)}(x)$	1×1 Conv + ReLU + LL	O_1
$f_s^{(k)}(x)$	1×1 Conv + ReLU	O_2
$O_1 + O_2$	LN	$proj$

A.5 ImageNet results

In this experiment we train a student ResNet-18 with a pre-trained teacher ResNet-34 on the ImageNet dataset (ILSVRC). Note that we do not perform any hyper-parameter tuning specifically for this configuration and use the same values we obtained for the CIFAR-100 dataset i.e. $\alpha = 0.9$, $\lambda_g = 0.2$, $\lambda_l = 0.8$, $\lambda_f = 0.8$. We observed that our method is able to reduce the gap between the teacher and the student performance by 1.44%. Results are presented in Table A.7.

A.6 Shallow CNN Architectures

In this section, we describe our experiments where we distill knowledge from a standard teacher network into a shallow custom-designed CNN. This is done to demonstrate that it is feasible to design and distill information into light-weight models such that they perform competitively with standard CNN architectures while running faster. For our experiments we use 2 shallow CNNs; (1) Conv-4 with 4

Table A.7: Observed top-1 validation accuracy (in %) of the student network on the ImageNet dataset using our method (MIMKD) and other distillation frameworks. In similar settings, the more recent Contrastive Representation Distillation (CRD) method reports comparable performance with an improvement of +1.42 from a student network Tian, Krishnan, and Isola 2019.

Student Network **Teacher Network**	**ResNet-18** **ResNet-34**
Student Accuracy	68.88
Teacher Accuracy	$72.82_{\mathbf{+3.94}}$
Knowledge Distill. (KD)	$69.66_{+0.78}$
Attention Transfer (AT)	$69.70_{+0.82}$
MIMKD (this work)	$\mathbf{70.32}_{\mathbf{+1.44}}$

convolutional-blocks followed by average pooling operation and a linear layer, where each convolutional-block is made-up of a convolutional layer with kernel size 3×3 and stride 2 followed by batch-normalization and a ReLU non-linearity, (2) Conv-4-MP which has 4 convolutions blocks followed by average pooling and a linear layer at the end, where each convolutional-block contains a convolutional layer with kernel size 3×3 and stride 1 followed by batch-normalization, ReLU and a max-pooling layer. These architectures were chosen as they are compact and run relatively faster on standard CPUs. Table A.8 compiles our results compared to other distillation methods for custom-designed shallow CNN architectures. Notice how a simple model such as Conv-4-MP becomes competitive with ShuffleNetV2's base student accuracy. Our method is able to outperform all other methods in this setup. Additionally, we can see that distillation is most successful with ResNet-32x4 as the teacher than for other architectures. This could be because of the larger gap in the baseline accuracy of the networks. Under this more controlled experiment with fixed students, larger gaps between student-teacher pairs also led to larger gains after distillation.

Table A.8: Observed test accuracy (in %) of shallow student networks trained with teacher networks of higher capacity and standard architectures on the CIFAR100 dataset using our methods MIMKD and other distillation frameworks.

Student Net.	**Conv-4**			**Conv-4-MP**		
Teacher Net.	**ResNet-110**	**VGG-13**	**ResNet-32x4**	**ResNet-110**	**VGG-13**	**ResNet-32x4**
Student Acc.	59.97	59.97	59.97	66.09	66.09	66.09
Teacher Acc.	73.82$_{+13.85}$	74.62$_{+14.65}$	79.24$_{+19.27}$	73.82$_{+7.73}$	74.62$_{+8.53}$	79.24$_{+13.15}$
FitNets	60.58$_{+0.61}$	61.81$_{+1.84}$	62.89$_{+2.92}$	67.38$_{+1.29}$	66.52$_{+0.43}$	67.21$_{+1.12}$
AT	61.65$_{+1.68}$	62.16$_{+2.19}$	63.10$_{+3.13}$	67.52$_{+1.43}$	66.21$_{+0.12}$	66.03$_{-0.06}$
VID	61.93$_{+1.96}$	62.49$_{+2.52}$	63.45$_{+3.48}$	67.76$_{+1.67}$	67.40$_{+1.31}$	67.86$_{+1.77}$
KD	61.98$_{+2.01}$	62.10$_{+2.13}$	62.87$_{+2.90}$	67.51$_{+1.42}$	67.84$_{+1.75}$	68.04$_{+1.95}$
CRD	62.13$_{+2.16}$	62.54$_{+2.57}$	63.76$_{+3.79}$	67.96$_{+1.87}$	68.06$_{+1.97}$	68.52$_{+2.43}$
MIMKD (ours)	**62.91**$_{+2.94}$	**62.95**$_{+2.98}$	**64.32**$_{+4.35}$	**68.77**$_{+2.68}$	**68.91**$_{+2.82}$	**69.09**$_{+3.00}$

A.7 Computational cost and negative sampling.

We contextualize the memory and computational overhead of our work with respect to CRD. Our global MI objective has the same footprint as CRD (i.e. an additional 600MB over standard Resnet18 training for storing negatives). In addition, our feature and local MI objective use projection layers which add an additional 100MB of GPU memory. As the computation of our JSD-based objective is computationally trivial, we observe negligible reduction in training speed wrt CRD (2.2 epochs/hr v. 2.4 epochs/hr). Note that no additional memory is used for sampling negatives for local and feature information maximization. The 4096 negatives are only used for global MI as storing 1-D representations is relatively inexpensive.

A.8 Ablation Study

In this section we present additional accuracy landscape plots for our extensive ablation study that demonstrates the value of each component of our mutual information maximization objective. We use a ResNet-32x4 as the teacher network and ResNet-8x4 as the student network where the baseline accuracy of the teacher is 79.24% and that of the student network is 72.44%. The values of the hyper-parameters λ_g, λ_l and λ_f — that control the weight of the global, local and feature mutual information maximization objectives respectively – were varied between 0 and 1 with an increment of 0.25 while the weight for the cross-entropy loss, α was set to 1. The following contour plots shows the test accuracy landscape with respect to a pair of hyper-parameters when the third hyper-parameter is set to distinct values. Overall, this demonstrates the value of maximizing region-consistent local and feature-level mutual information between representations in addition to just global information maximization.

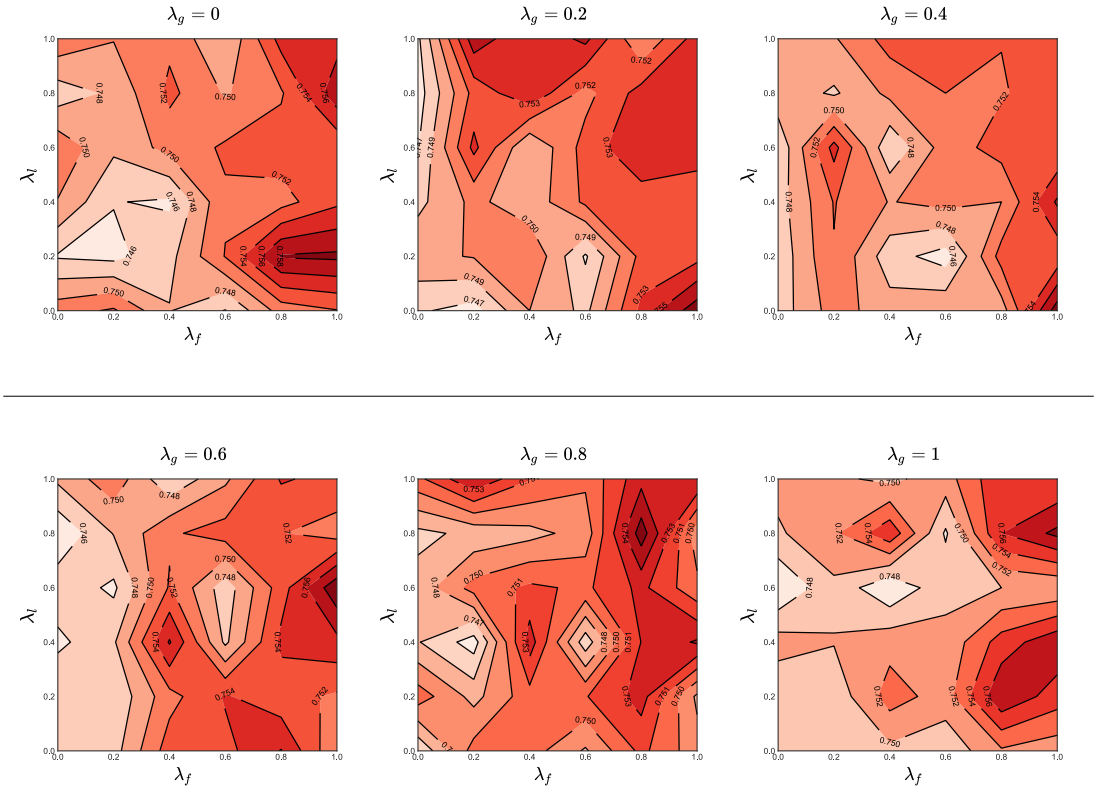


Figure A.1: Results from the ablation studies on CIFAR100 dataset using a student resnet8x4 (baseline acc. 72.44%) with teacher resnet32x4 (baseline acc. 79.24%). Contour lines represent the final test accuracy of the student. Grid search was performed by varying the values of λ_f, λ_g, λ_l from 0 to 1 with increments of 0.25. In each plot, the accuracy landscape is shown with λ_g set to a constant value.

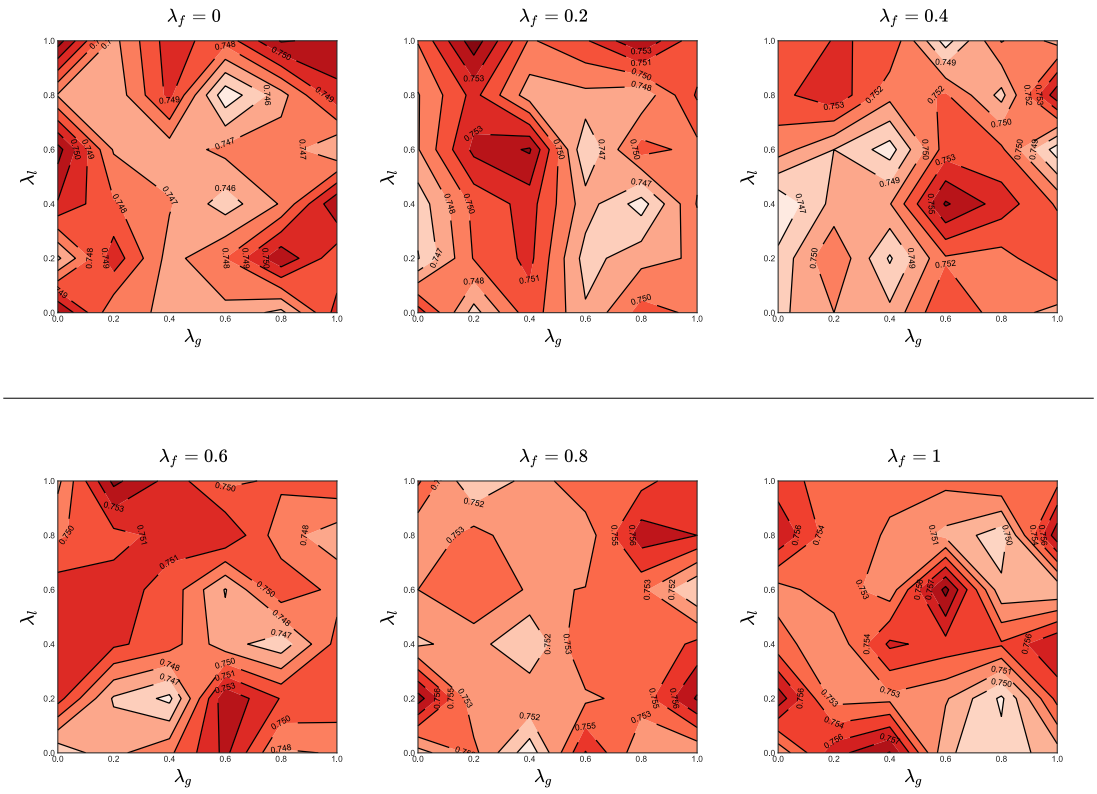


Figure A.2: Results from the ablation studies on CIFAR100 dataset using a student resnet8x4 (baseline acc. 72.44%) with teacher resnet32x4 (baseline acc. 79.24%). Contour lines represent the final test accuracy of the student. Grid search was performed by varying the values of λ_f, λ_g, λ_l from 0 to 1 with increments of 0.25. In each plot, the accuracy landscape is shown with λ_f set to a constant value.

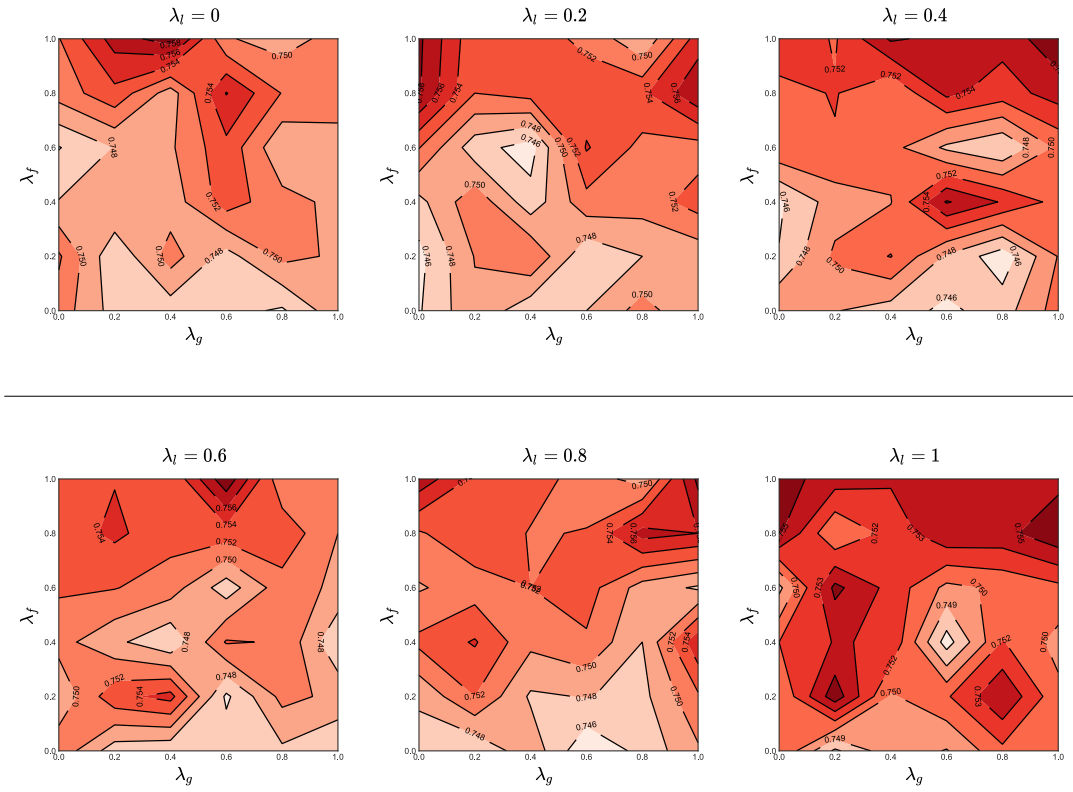


Figure A.3: Results from the ablation studies on CIFAR100 dataset using a student resnet8x4 (baseline acc. 72.44%) with teacher resnet32x4 (baseline acc. 79.24%). Contour lines represent the final test accuracy of the student. Grid search was performed by varying the values of λ_f, λ_g, λ_l from 0 to 1 with increments of 0.25. In each plot, the accuracy landscape is shown with λ_l set to a constant value.

Appendix B

CLIP-Lite: Information Efficient Visual Representation Learning with Language Supervision

B.1 Discussion on JSD-based lower bound on Mutual Information

Recall that for given random variables y and z, their mutual information is defined as a Kullback-Leibler (KL) divergence between their joint distribution $p(y, z)$ and the product of their marginal distributions, $p(y)p(z)$ as, $I(y; z) = D_{\mathrm{KL}}(p(y, z) \,||\, p(y)p(z))$. The above formulation of MI gives rise to the commonly used contrastive objective InfoNCE (Oord, Y. Li, and Vinyals 2018). Alternatively, the KL-divergence can be replaced with the Jensen-Shannon divergence (JSD) between the joint and the product of marginals as an estimate of the Pointwise Mutual Information(PMI) between two views of the data i.e. $I^{JSD}(y; z) = D_{\mathrm{JSD}}(p(y, z) \,||\, p(y)p(z))$. And as discussed in (Hjelm et al. 2018), this formulation of MI leads to the following relation,

$$JSD(p(y,z)||p(y)p(z)) \propto \mathbf{E}_{y\sim p(y)}[\mathbf{E}_{z\sim p(z|y)}[\log \tfrac{p(z|y)}{p(z)} - (1 + \tfrac{p(z)}{p(z|y)}) \log \left(1 + \tfrac{p(z|y)}{p(z)}\right)]]$$

Table B.1: CLIP-Lite outperforms CLIP-COCO on both VOC and ImageNet classification tasks, and performs comparably to VirTex. CLIP-Lite's performance is comparable or superior to both supervised and self-supervised learning models trained with images alone, even those trained with 10x more images. (IN-Sup. = ImageNet-supervised.)

Method	**# images**	**Annotations**	**VOC07**	**IN-1k**
COCO-Sup.	118K	labels	86.2	46.4
IN-Sup.	1.28M	labels	87.6	75.6
MoCo-COCO	118K	self-sup.	67.5	46.5
MoCo-IN v1	1.28M	self-sup.	79.4	60.8
PCL v1	1.28M	self-sup.	83.1	61.5
SwAV (200 ep.)	1.28M	self-sup.	87.9	72.7
ICMLM	118K	captions	87.5	47.9
VirTex	118K	captions	88.7	53.8
CLIP-COCO	118K	captions	74.2	33.2
CLIP-Lite	118K	captions	88.2	55.3

Now, the quantity inside the expectation above is a concave, monotonically increasing function of the ratio $p(z|y)/p(z)$, which is exactly the exponential of the Pointwise Mutual Information, i.e. $e^{PMI(y,z)}$.

B.2 Comparison with SSL Pretraining Methods

In this section, we evaluate the performance of our method against other pre-training frameworks and image-only SSL methods. We observe that CLIP-Lite is comparable or better to image-only SSL learning models trained on downstream ImageNet classification with a frozen ResNet-50 backbone, even though our method is trained on much fewer images, albeit with textual supervision.

B.3 Mutual Information Discriminator

As described in main paper, our JSD-based lower-bound on mutual information relies on a discriminator function, $T_\omega : \mathcal{Y} \times \mathcal{Z} \to \mathbb{R}$, which distinguishes between samples extracted from the joint distribution, $P(Y, Z)$ i.e. a positive image-caption pair and the product of marginals, $P(Y)P(Z)$ i.e. a negative image-caption pair. This discriminator function can be modelled as an arbitrary neural network with parameters ω that can be jointly optimized with the encoders during training (Belghazi et al. 2018). In this work, we use a projection and alignment based architecture similar to the one presented in Deep InfoMax (Hjelm et al. 2018).

Given a pair of input one-dimensional representations, both vectors are first projected using a projection module with two linear layers separated by a ReLU and a linear shortcut. A dot-product of these projections is then computed to get alignment scores. The projection function maps these representations to an aligned cross-modal latent space. Separate projection functions are used for image and text representations. Positive and negative pairs of image-text representations are passed through the discriminator to get respective scores which are then used to estimate and maximize mutual information using our objective. This architecture, in addition to being simple and computationally inexpensive, also offers alignment of the representations into a common cross-modal latent space which uses cosine similarity as the distance metric.

B.4 Ablations

Batch-size Ablations: A salient feature of our pre-training framework is that we use a lower-bound on the mutual information that can be optimized with only one negative sample. This allows us to use much smaller batch-sizes compared to the original CLIP (Radford et al. 2021) model. In this section, we evaluate the PASCAL VOC classification performance of the visual backbones trained with a batch sizes 64, 128, 256, 512 and 1024. These ablations are performed with a 2-layered BERT model as the text-encoder and a ResNet-50 as the image encoder for 200K iterations.

Table B.2: **Batch size Ablations:** We show the performance of a ResNet-50 trained with CLIP-Lite using varying batch-sizes. We observe that the performance drops marginally with the batch size 512. Additionally, we can see that the model is able to converge fairly well with the significantly lower batch size of 64.

Batch Size	VOC07
64	74.7
128	81.3
256	84.9
512	87.5
1024	87.9

Visual Encoder Ablations: In this section, we compare the performance of our pretraining method using a ResNet-18, ResNet-50, and ResNet-101 backbones using the downstream PASCAL VOC classification task. These ablations are performed with a 2-layered BERT model as the text-encoder with a batch-size of 512 for 200K iterations.

Text Encoder Ablations: In this section, we compare the downstream PASCAL VOC (Everingham et al. 2010) classification performance of a ResNet-50 visual back-

Table B.3: **Visual Encoder Ablations:** We show the performance of CLIP-Lite using 3 visual backbones of varying sizes.

Visual Backbone	VOC07
ResNet-18	83.8
ResNet-50	87.5
ResNet-101	87.8

bone pretrained using a text encoder transformer with varying capacities. We train 4 transformer variants, (1) pre-trained $\text{BERT}_{\text{base}}$ (Devlin et al. 2018), (2) 2-layered, (3) 4-layered, (4) 6-layered, and a (5) 12-layered BERT-like transformer. These ablations are performed with a ResNet-50 as the image encoder with a batch-size of 512 for 200K iterations.

Table B.4: **Text Encoder Ablations:** We show the performance of a ResNet-50 trained with CLIP-Lite using different text encoders. We observe that the performance drops marginally when training from scratch. Additionally, we also see that using a transformer with 2-layers works almost as well as a 12-layered transformer when trained from scratch.

Text Encoder	VOC07
$\text{BERT}_{\text{base}}$ init.	88.1
2-layers	87.5
4-layers	87.6
6-layers	87.6
12-layers	87.9

Zero-shot classification templates While performing zero-shot classification, we use the class names of target images to generate captions that the images should align with. The performance is compared when captions are generated using three different templates. We test three different class prompt templates and compare our performance against an equivalently trained CLIP model on the COCO dataset. As

seen in Table B.5, both CLIP and CLIP-Lite prefer more descriptive prompts.

Table B.5: **Zero-Shot Templates on CIFAR-10:** We evaluate different prompts and find the CLIP-Lite prefers more descriptive prompts.

Class Prompt	CLIP-COCO	CLIP-Lite
"a {class name}"	13.3	30.8
"a picture of a {class name}"	14.5	32.6
"a photo of a {class name}"	16.3	33.0

B.5 Training CLIP on COCO-Captions Dataset

We use a CLIP model trained on the COCO dataset as a baseline for several demonstrated tasks. For this purpose, we use an open-source implementation[1] of CLIP. We train a standard ResNet-50 (He, X. Zhang, et al. 2016) based CLIP model that takes in a 224×224 image and generates 2048-dimensional features at the pre-logit layer. For textual encoding, we use a transformer (Vaswani et al. 2017) model and use the output `[CLS]` token as the text representation. We use the COCO Captions dataset (X. Chen, Fang, et al. 2015) which has 118K images with five captions per image. During training time we apply (1) random cropping, (2) color jittering, (3) random horizontal flips while interchanging the words 'left' and 'right' in the caption, and (4) normalization using the ImageNet image mean. We train using the Adam Optimizer (Kingma and Ba 2014a) with decoupled weight decay regularization (Loshchilov and Hutter 2016) for all weights except gains or biases. We perform distributed training across 8 GPUs with batch normalization (Ioffe and Szegedy 2015a) per GPU with an overall batch-size of 1024. We warm-up to the initial learning rate in 10K steps and decay to zero with the cosine schedule. We found that using the learning rate of 10^4 works

[1] https://github.com/mlfoundations/open_clip

slightly better (+1.4% on VOC07) than the originally recommended 5×10^5. We also found that the performance incrementally improves (+1.9% on VOC07) with longer training therefore we train for 250K iterations, similar to ours. All other training details and hyper-parameters were kept the same as the original work (Radford et al. 2021). Please note that the ResNet-50 backed CLIP model trained by us on the COCO dataset outperforms (+1.2% Zero-shot Acc. on CIFAR10) publicly available weights[2].

[2] https://github.com/revantteotia/clip-training/blob/main/zero_shot_eval_output/coco_trained_clip_observations.md

Appendix C

SAASN: Self-Attentive Adversarial Stain Normalization

C.1 Additional results

We trained and tested the model in both a *one-to-one* ($K = 1$) and *many-to-one* ($K = 2$) setup. In this section we demonstrate the model performance, on test datasets, for visual inspection.

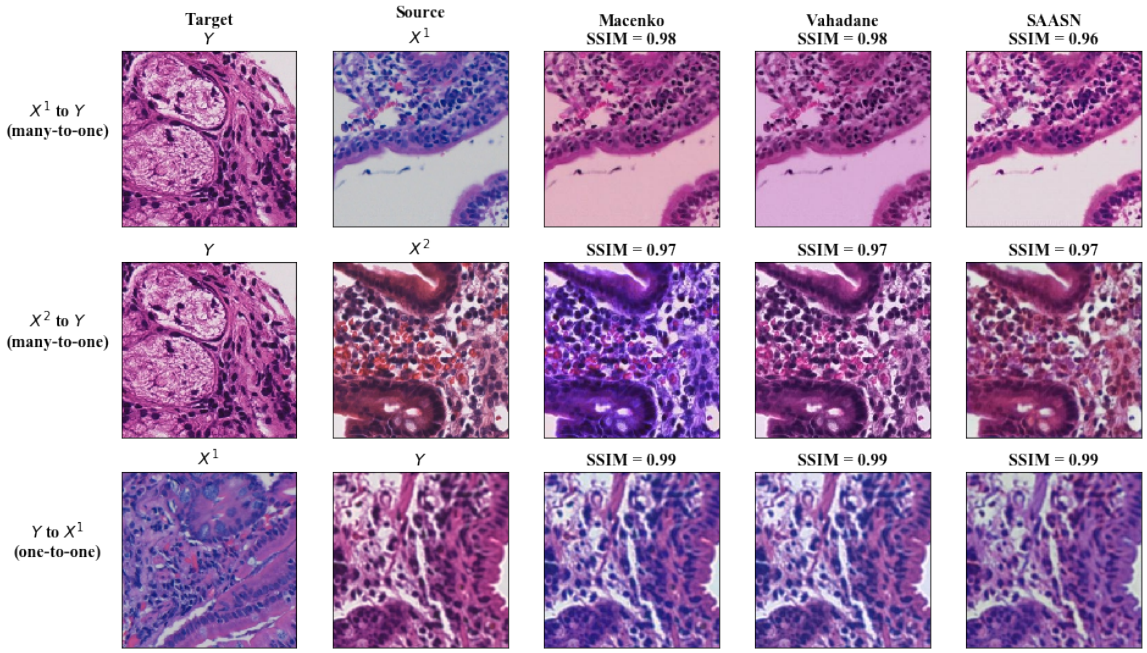


Figure C.1: Visual comparison of performance in cases where Macenko and Vahadane techniques perform very well according to a combined SSIM index. The target image only applies to the Macenko and Vahadane techniques. The main results section included a visual comparison of SAASN stain transfers with the worst performing Macenko and Vahadane images based on SSIM. Alternatively, SAASN is also compared to the best SSIM indexes for the other two techniques. Figure C.1 displayed the top three images in each stain transfer scenario based on the highest $L2$-norm of Macenko and Vahadane SSIM results. For the $X^{(1)}$ to Y transfer, SAASN was the only technique that properly maintained a whitish/gray background pixel color. For the $X^{(2)}$ to Y transfer, Macenko appeared to create a new stain distribution that was not close to the desired target image. All three normalizations performed well in the *one-to-one* transfer. The comparison in Figure C.1 demonstrates that SAASN can perform better at preserving structure and properly transferring stain domains, because both areas are incorporated into the network's loss functions.

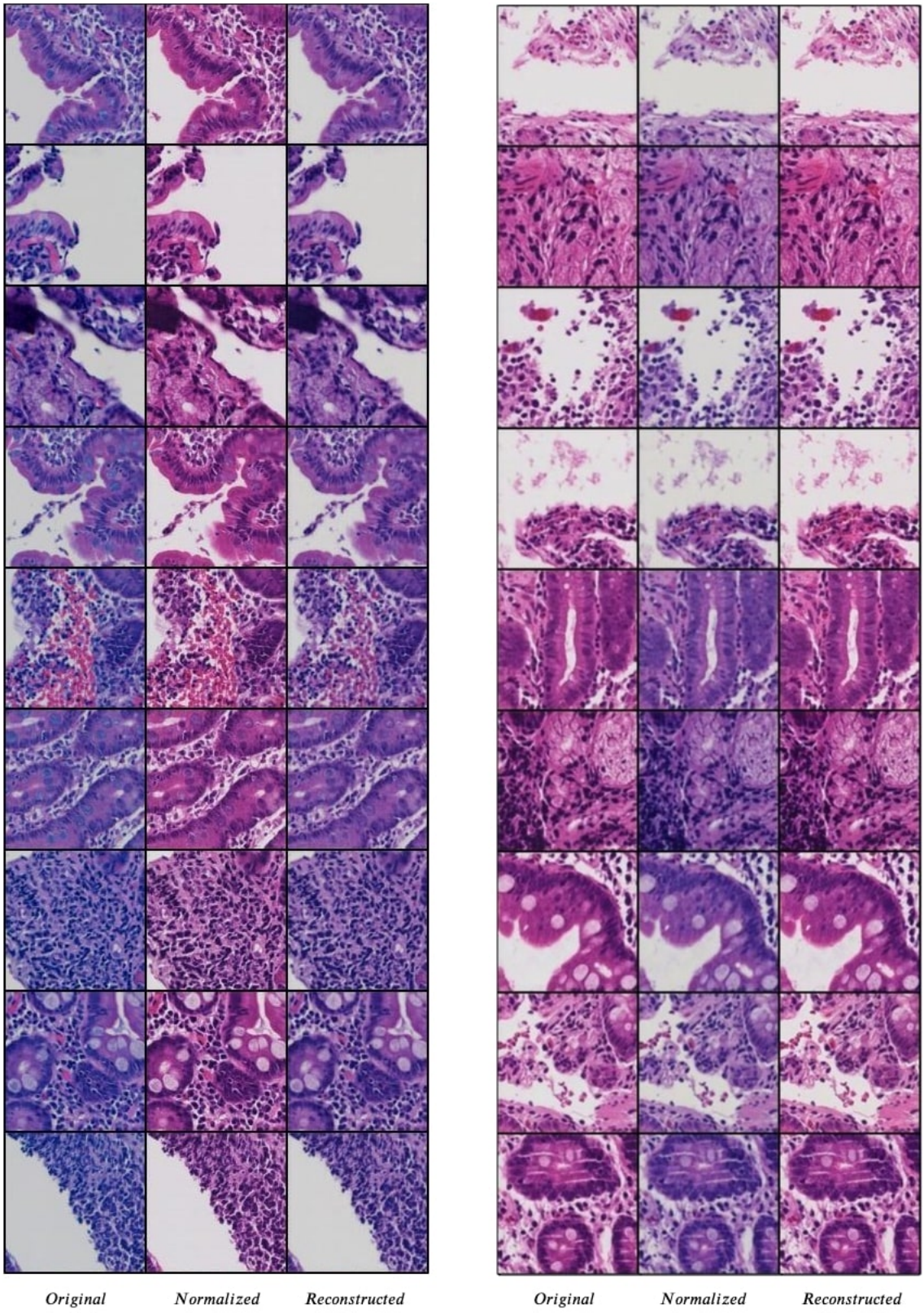


Figure C.2: *One-to-one* (K=1) model. **Left:** Translation from domain $X^{(1)}$ to Y and back to domain $X^{(1)}$. **Right:** Translation from domain Y to $X^{(1)}$ and back to Y.

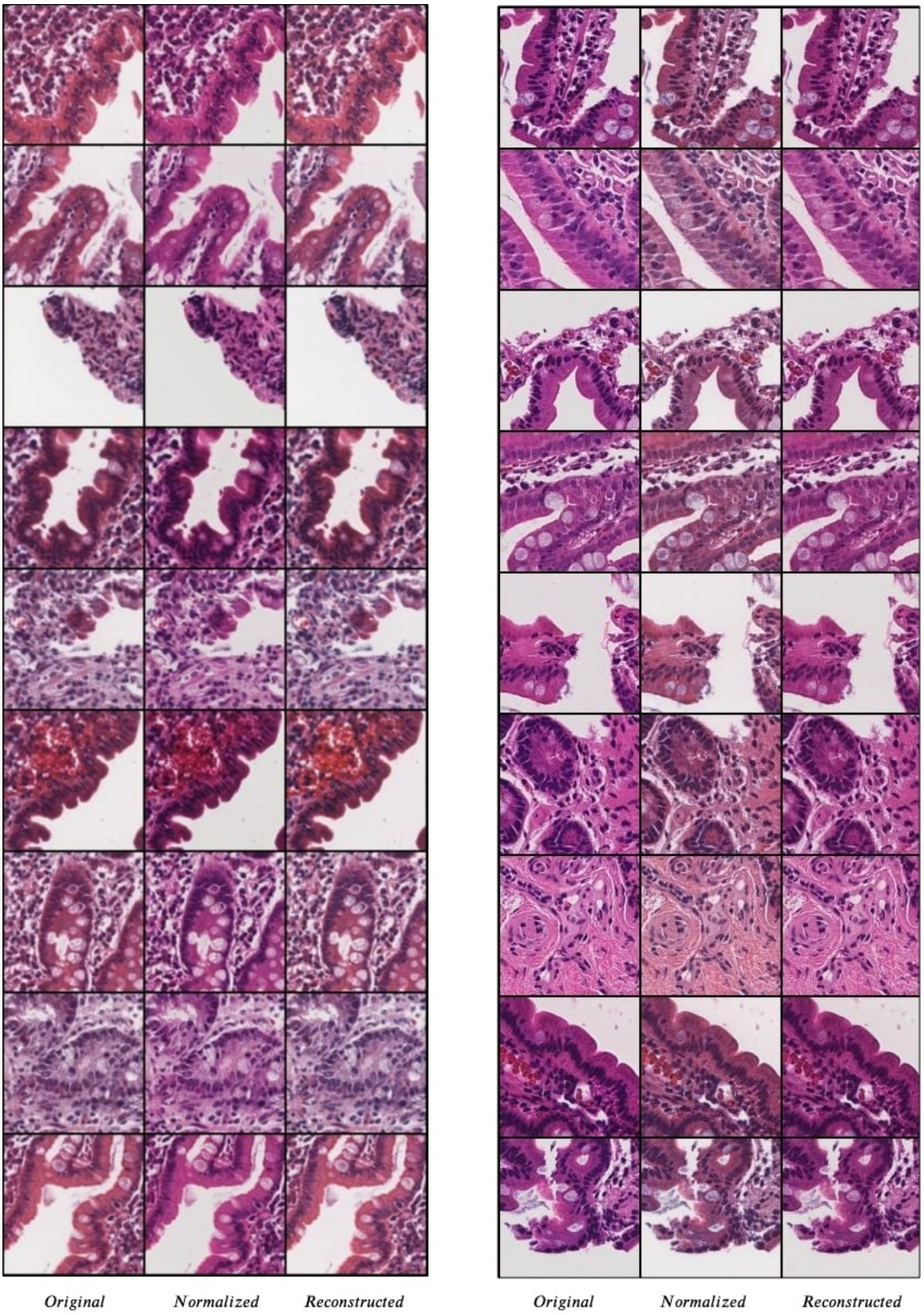


Figure C.3: *One-to-one* (K=1) model. **Left:** Translation from domain $X^{(2)}$ to Y and back to domain $X^{(2)}$. **Right:** Translation from domain Y to $X^{(2)}$ and back to Y.

Figure C.4: The model was also trained on Yosemite summer to winter dataset from the CycleGAN paper. **Left:** Translation from winter to summer and back to winter. **Right:** Translation from summer to winter and back to summer. The model was trained with the same parameters as for the stain normalization task.