Gen-SIS: <u>Gen</u>erative <u>S</u>elf-augmentation <u>I</u>mproves <u>S</u>elf-supervised Learning

Varun Belagali[1*], Srikar Yellapragada[1*], Alexandros Graikos[1], Saarthak Kapse[1], Zilinghan Li[2],
Tarak Nath Nandi[2,3], Ravi K Madduri[2,3], Prateek Prasanna[1], Joel Saltz[1], Dimitris Samaras[1]

[1]Stony Brook University   [2]Argonne National Laboratory   [3] University of Chicago

Abstract

Self-supervised learning (SSL) methods have emerged as strong visual representation learners by training an image encoder to maximize similarity between features of different views of the same image. To perform this view-invariance task, current SSL algorithms rely on hand-crafted augmentations such as random cropping and color jittering to create multiple views of an image. Recently, generative diffusion models have been shown to improve SSL by providing a wider range of data augmentations. However, these diffusion models require pre-training on large-scale image-text datasets, which might not be available for many specialized domains like histopathology. In this work, we introduce Gen-SIS, a diffusion-based augmentation technique trained exclusively on unlabeled image data, eliminating any reliance on external sources of supervision such as text captions. We first train an initial SSL encoder on a dataset using only hand-crafted augmentations. We then train a diffusion model conditioned on embeddings from that SSL encoder. Following training, given an embedding of the source image, this diffusion model can synthesize its diverse views. We show that these 'self-augmentations', i.e. generative augmentations based on the vanilla SSL encoder embeddings, facilitate the training of a stronger SSL encoder. Furthermore, based on the ability to interpolate between images in the encoder latent space, we introduce the novel pretext task of disentangling the two source images of an interpolated synthetic image. We validate Gen-SIS's effectiveness by demonstrating performance improvements across various downstream tasks in both natural images, which are generally object-centric, as well as digital histopathology images, which are typically context-based.

1. Introduction

In recent years, self-supervised learning (SSL) [1, 6, 10, 20, 23, 25, 40, 51] has emerged as a standard approach for
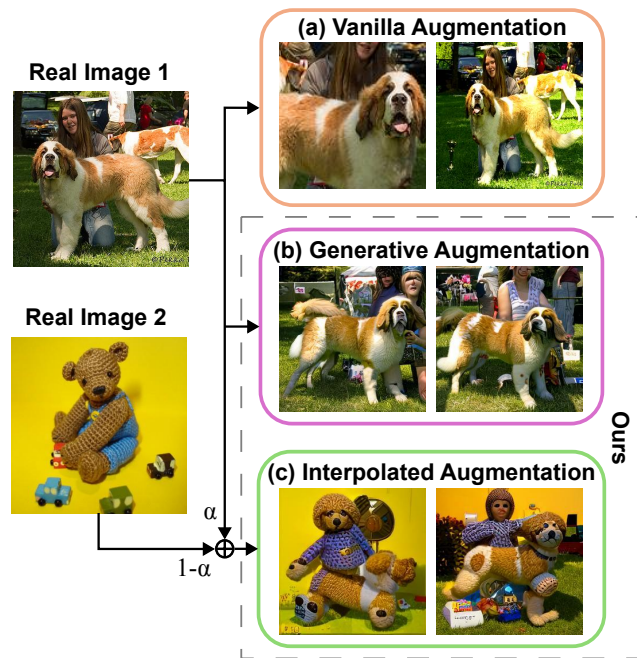

Figure 1. (a) Vanilla augmentations used in SSL such as random cropping, color jittering. (b) Generative augmentations (ours) are conditioned on a single source image. (c) Interpolated augmentations (ours) conditioned on a pair of images. In the Gen-SIS framework, we use (b) for view augmentation, and (c) for the disentanglement pretext task, both in conjunction with (a).

learning robust visual representations that excel across various downstream tasks. By optimizing the model weights on pretext tasks, like self-prediction or view invariance, SSL enables models to learn discriminative features without requiring labeled data. Specifically, approaches such as DINO [6], BYOL [20], and SimCLR [10] have achieved notable success, producing high-quality features that transfer effectively to diverse downstream applications. This success stems from view-invariance tasks, which encourage models to learn high-level discriminative features from the image. Formulating view-invariant tasks relies heavily on hand-crafted augmentations, such as cropping and color jit-

---

*Equal contribution. Correspondence to vbelagali@cs.stonybrook.edu
webpage: https://histodiffusion.github.io/docs/publications/gensis

tering, to create multiple views of an image. *Stronger augmentations typically lead to more robust features, as they increase the difficulty of the invariance task [20].*

On a parallel front, diffusion models have achieved impressive quality in image generation, driven by innovations in architecture [41, 48], sampling methods [52], and conditioning techniques [28, 46]. This success has led to an interest in using diffusion models, especially large foundation models like Stable Diffusion (SD), for data augmentation [54, 55]. Given SSL's reliance on augmentations, diffusion models could significantly improve SSL by generating images with *non-trivial* variations in background, shape, and position of objects, while preserving the original high-level semantics (Fig. 1 (b))

Recent work by Tian et al. [54] has investigated using synthetic data generated from Stable Diffusion (SD) as multiple views for SSL. However, employing SD as an augmenter in SSL has some drawbacks: (1) It is challenging to adapt SD in domains underrepresented in SD's training data, LAION-5B [50]. Since it is a general image foundation model, it is expected that it cannot generate high-quality images from specific domains such as histopathology. The low-quality images generated by SD cannot be used for SSL as they are highly inaccurate (see supplementary). (2) SD-scale foundation models are usually not available for other domains, outside natural images, and training them from scratch is a task beyond the scope of improving an SSL encoder. (3) Apart from synthesizing variations of an image, it is not straightforward to perform other kinds of augmentations by controlling the conditioning in text-to-image models. For instance, interpolating between two images would require using an LLM to first 'interpolate' the two captions and then synthesize a new image. (4) As a text-conditioned model, SD is trained on paired image-text data, which can be seen as conflicting with the SSL principle of training on unlabeled data.

To avoid these issues, in this paper, we introduce Gen-SIS, a method to train a diffusion model on the same unlabeled data as an SSL model and use it as an effective augmenter for the SSL without any additional supervision, such as text or class labels. We adopt the term *self-augmentation* to highlight the distinction between generative augmentations that rely on external supervision and our strictly self-supervised approach.

We begin by pre-training an SSL *encoder* on real images from the pre-training dataset, using the original hand-crafted augmentations. Next, we train a latent diffusion model [48] (LDM), conditioned on image embeddings extracted from this initial SSL *encoder*. Once trained, the LDM is then used to synthesize novel images for training a new enhanced SSL *encoder*.

Gen-SIS expands the data augmentation using self-augmentations from the diffusion model, moving beyond traditional hand-crafted augmentations. In a view-invariant setting, a pair of real and synthetic images from our diffusion model can act as different views of the same image, strengthening the augmentation process (Fig. 1 (b)).

Furthermore, we utilize the generative model's capabilities and propose a novel pretext task that complements the base SSL task by focusing on disentangling shared concepts between pairs of images. The trained LDM can interpolate between images by interpolating between the image embeddings provided as conditioning. The generated image semantically blends concepts from a given pair of real images (Fig. 1 (c)). We then task the visual encoder with identifying features from the original pair of images used in generating the interpolated image. This additional pretext task (termed as *disentanglement pretext task*) forces the model to learn and distinguish various object, texture, and shape-level features. Solving this task presents a greater challenge to the encoder, significantly enhancing its performance on downstream tasks.

In summary, our contributions are:

- We introduce Gen-SIS, the first generative diffusion-enhanced SSL approach that requires only unlabeled data.
- We propose a novel disentanglement task, as an additional pretext task in self-augmentation enhanced SSL training.
- We extensively evaluate our method on ImageNet-1K and benchmark the Gen-SIS pretrained encoder across a range of downstream tasks such as classification, retrieval, copy detection, and video segmentation, achieving notable performance gains over vanilla SSL.
- Using Gen-SIS, we extend self-augmented SSL to histopathology images, a domain with no foundation generative models, demonstrating the effectiveness of our self-contained approach.

2. Related work

Self-supervised Learning: Self-supervised learning [17] aims at learning generic representations from large-scale unlabeled data through a pretext task. Pretext tasks can be mainly classified into self-prediction and view-invariance tasks. Self-prediction methods (MAE [24], MaskFeat [58]) involve masking parts of an image and then training the model to reconstruct the missing information based on the remaining context. View-invariant methods task the model to output similar features for two augmented views of the same image. This involves contrastive methods like Sim-CLR [9], MoCo [22], NNCLR [15] and self-distillation methods like BYOL [21], DINO [6], iBOT [60], and DI-NOv2 [40]. View-invariant methods typically rely on hand-crafted augmentations to derive multiple views of the same image for pretext tasks.
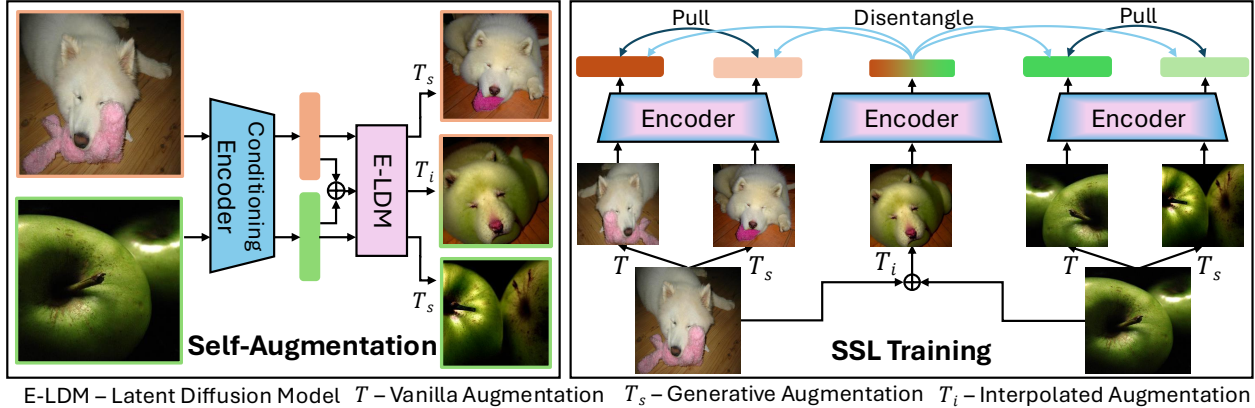
Figure 2. Overview of the Gen-SIS-framework: It contains 2 key steps 1) Self-Augmentation using Embedding conditioned LDM (E-LDM), 2) SSL training with augmentations from E-LDM. T represents vanilla augmentations, T_s represents generative augmentation from single image, and T_i represents interpolated augmentation from two images. Note that in conjunction with T_s and T_i, we applied vanilla augmentation. *Pull* represents the vanilla SSL pretext task, and *Disentangle* represents our proposed pretext task with interpolated augmentation.

Diffusion Models: Diffusion models were first introduced in the seminal work of Ho *et al.* [30]. Subsequent advancements included class-conditioning and guidance techniques for more controlled generation [29, 46], and accelerated sampling techniques [52]. Latent diffusion models [7, 41, 49] enable high-resolution image generation by performing the diffusion process in a smaller latent space. In specialized domains such as histopathology, where labeled image-text data is limited, prior works have adopted image embedding-conditioned diffusion models [19, 37] to overcome these constraints.

Data augmentation with Diffusion models: Recent research has utilized diffusion models for data augmentation, particularly in supervised settings [2, 18, 19, 55]. The studies most closely related to our research are Stable-rep [54] and SynCLR [31]. Stable-rep leverages captions from the CC-12M dataset to generate synthetic samples from Stable Diffusion [48] (SD), using them as multiple positive pairs in the SSL training.

SynCLR, following a similar approach to Stable-Rep, uses ImageNet object categories to construct text prompts. However, SD-scale text-to-image models are usually unavailable beyond natural images.

Moreover, models trained on large-scale internet datasets, like LAION-5B, may accidentally contain examples from common benchmarks such as ImageNet. Previous works [5, 55] have shown that pretrained diffusion models can leak training data, thus potentially inflating SSL performance.

3. Preliminary

DINO: In this study, we use DINO [6] as our vanilla self-supervised learning (SSL) method. DINO (self-**di**stillation

with **no** labels) is a teacher-student framework in which two augmented views of an image, I' and I'', are processed separately by the student g_{θ_s} and teacher g_{ϕ_t} networks. The two augmented views are generated using standard augmentations, including cropping, color jittering, Gaussian blur, and solarization. Both teacher and student share the same architecture, with a backbone encoder and a projection head, and output a probability distributions P over K dimensions.

$$L_s = g_{\theta_s}(I'), \quad P_s^k = \frac{\exp(L_s^k/\tau_s)}{\sum_{j=1}^K \exp(L_s^j/\tau_s)}, \quad (1)$$

$$L_t = g_{\phi_t}(I''), \quad P_t^k = \frac{\exp((L_t^k - c^k)/\tau_t)}{\sum_{j=1}^K \exp((L_t^j - c^j)/\tau_t)}, \quad (2)$$

$$H(P_t, P_s) = -P_t \log(P_s), \quad \theta_s \leftarrow \text{Optimizer}(H, \theta_s) \ (3)$$

The student's output (logits L_s) is sharpened using a low-temperature τ_s softmax (Eq. 1), while the teacher's output (logits L_t) undergoes centering with a moving average of the teacher outputs c and softmax sharpening with τ_t to prevent collapse during training (Eq. 2). The student network is optimized to match the teacher's probability distribution using a cross-entropy loss H (Eq. 3). The teacher network is updated as exponential moving average (EMA) of the student network's weights.

Latent Diffusion Models: Latent Diffusion Models (LDMs) [48] synthesize images efficiently by learning to draw samples from a compressed image latent space instead of operating directly on pixels. This latent space is defined by a learned Variational Autoencoder (VAE), with a VAE encoder that maps images from pixels to latents, and a VAE decoder that maps the latent back to pixel space. Using the diffusion denoising objective [30], LDMs train

a U-Net denoiser in the latent space. To control the generated images, the U-Net is usually conditioned on additional information about the images, such as class labels or text prompts. LDMs utilize a cross-attention mechanism between embeddings of the conditioning information and the U-Net features to guide the image synthesis, rendering the conditioning framework flexible to the choice of conditioning signals.

4. Method

In this section, we introduce Gen-SIS (see Fig. 2), a framework that leverages unlabeled data to train a diffusion model and subsequently enhances self-supervised learning (SSL) through novel self-augmentations using this learned diffusion model. First, in Sec. 4.1, we describe the embedding-conditioned Latent Diffusion Model (E-LDM), which generates synthetic images based on the embeddings of source images. Then in Sec. 4.2, we detail how synthetic images (self-augmentations) generated by the E-LDM can be integrated into SSL to improve it. We focus on two types of self-augmentations: (1) Generative augmentations, where augmentations are created from a single source image, and (2) Interpolated augmentations, where an interpolated image is generated from two source images and used in training for a novel disentanglement pretext task.

4.1. Embedding conditioned LDM

We follow the LDM [48] framework for synthetic image generation, conditioning the LDM with the embedding extracted from an image, and refer to this setup as E-LDM (embedding-conditioned LDM). Following the approach of prior work [19], we first train an image encoder on unlabeled real images using a standard SSL algorithm (DINO), and then use this encoder as the conditioning encoder to condition the diffusion model. This design allows our E-LDM to be trained in a fully self-supervised manner, without relying on any auxiliary information about the images. We term the synthetic images generated from E-LDM as self-augmentations. As conditioning, we choose the output of the DINO backbone, which is a D-dimensional vector e (embedding). Once trained, we can then prompt the E-LDM by giving it an embedding of a real image e; it will synthesize a variation $I_s = \text{E-LDM}(z, e)$, where $z \sim \mathcal{N}(0, I)$ is an initial Gaussian noise used in sampling. We use the deterministic DDIM [52] sampling algorithm, which maps every (z, e) pair to an image I_s.

4.2. Enhancing SSL using self-augmentations

With real images as sources for E-LDM conditioning, we use two types of self-augmentations: 1) Generative Augmentations, 2) Interpolated Augmentations.
Generative Augmentations: In the generative augmentation setting, a synthetic image is generated using a single

real image as the source. This involves first extracting an embedding e from the source image using the conditioning-encoder, and then guiding the image generation process with that embedding to create a synthetic image $I_s = \text{E-LDM}(z, e)$. As illustrated in Fig. 1 (b), generative augmentations introduce novel variations in the shape, size, and position of objects, as well as changes in the background, while preserving the semantic content of the objects in the image. As shown in Fig. 2, to integrate generative augmentations into SSL, we use the real image and a corresponding synthetic image as an input pair for the SSL pretext task. We also apply hand-crafted augmentations to both real and synthetic images.
Interpolated Augmentations: An interesting property of diffusion models is their ability to generate an image that partially resembles each source image when conditioned on embeddings interpolated from two sources, as demonstrated in prior works [19, 33, 56]. We leverage this property to produce an interpolated synthetic image from two real source images, which we use to perform a new pretext task during the SSL training. With embeddings e_1 and e_2 representing the two source images (I_1, I_2), and an interpolation ratio α, we compute an interpolated embedding e_{int} using spherical linear interpolation (SLERP) [56] $e_{\text{int}} = \text{SLERP}(e_1, e_2, \alpha)$. We choose SLERP over linear interpolation since high-dimensional vectors are concentrated near the surface of the unit sphere. This interpolated embedding serves as the conditioning to generate the synthetic interpolated image, $I_{\text{int}} = \text{E-LDM}(z, e_{\text{int}})$.

Since the interpolated image contains components of both source images, we propose a disentanglement task where the network learns to separate the distinct features of each source image used in the interpolation. Specifically, given two source images (I_1, I_2), an interpolating ratio (α), and the interpolated synthetic image (I_{int}), we pass I_{int} through the student network, to obtain the student probability P_{int}.

$$L_{\text{int}} = g_{\theta_s}(I_{\text{int}}), \quad P_{\text{int}}^k = \frac{\exp(L_{\text{int}}^k / \tau_s)}{\sum_{j=1}^{K} \exp(L_{\text{int}}^j / \tau_s)} \quad (4)$$

To derive a target teacher output for the disentanglement task, we pass I_1, I_2 to the teacher network individually, and interpolate the teacher head output (logits L_{ent}) using α:

$$L_{\text{ent}} = \alpha g_{\phi_t}(I_1) + (1 - \alpha) g_{\phi_t}(I_2). \quad (5)$$

This is then passed through the centering and sharpening operation to get the probability over the K dimensions

$$P_{\text{ent}}^k = \frac{\exp((L_{\text{ent}}^k - c^k) / \tau_t)}{\sum_{j=1}^{K} \exp((L_{\text{ent}}^j - c^k) / \tau_t)} \quad (6)$$

Finally, we compute the disentanglement loss Eq.7 using

the cross-entropy between the student and teacher predictions.

$$\mathcal{L}_{\text{disentangle}} = -P_{\text{ent}} \log(P_{\text{int}}) \quad (7)$$

To optimize this loss, the student must implicitly disentangle components of the pair of source images within the interpolated image, leading us to call this a *disentanglement pretext task*. This task is more challenging and can yield better representation learning compared to optimizing solely for single-source augmentations. With single-source images, the student only needs to extract features for a single dominant component to minimize the loss, whereas disentangling multiple components in an interpolated image can help the model learn more discriminative features.

In Gen-SIS, we use both types of self-augmentations, generative augmentation with vanilla dino loss and interpolated augmentation with $\mathcal{L}_{\text{disentangle}}$. We provide the pseudo code in the supplementary.

5. Experiments: Natural Images

In this section, we apply the Gen-SIS framework to enhance SSL pre-training in the natural image domain. Our experiments below empirically demonstrate improvements in encoder pre-training using Gen-SIS compared to the vanilla DINO on diverse downstream tasks: classification, retrieval, copy detection, and video segmentation. We also provide evaluation on out-of-distribution data in the supplementary. Although we conduct experiments with DINO, our self-augmentation technique is a general method that can be readily extended to other SSL approaches.

5.1. Setup

Training: Aligning closely with the experimental setup of DINO [6], we pre-train the models on the ImageNet-1K dataset [13]. To begin, we reproduce the pre-training of ViT-S/16 model using the DINO framework (trained only on real images) on a 100 epoch setting with DINO's codebase. We use this model as the baseline and conditioning encoder for our E-LDM. For our enhanced SSL training, we improve DINO with the Gen-SIS framework and call the method Gen-DINO. In Gen-DINO, we pre-train the ViT-S/16 model with generative and interpolated augmentations. Both DINO and Gen-DINO are trained for 100 epochs from scratch with a cosine annealing learning rate schedule with an initial value of 5×10^{-4}, a 10-epoch warmup period, and a linear scaling rule with respect to the batch size [11]. The weight decay also follows a cosine schedule, from 0.04 to 0.4. We use the AdamW optimizer with a batch size of 1024. We use generative and interpolation augmentation in Gen-SIS, in conjunction with the default handcrafted data augmentations of DINO, such as color jittering, cropping, flipping, Gaussian blur, solarization, and multi-crop. For both vanilla DINO and Gen-DINO, by default, we use 8

local crops; in ablations, we further show the performance without using local crops. For interpolated image generation, we use $\alpha = 0.5$.

We train the LDM as an embedding conditioned model following [19]. The LDM configuration includes a VQ-f4 autoencoder that downsamples images from $256 \times 256 \times 3$ to $64 \times 64 \times 3$. For ImageNet experiments, we train the U-Net denoiser from scratch. We set the learning rate to 10^{-4} with a warmup period of 1000 steps. To generate images, we use DDIM sampling [52] with 50 steps and apply classifier-free guidance [28]. We generate self-augmentations using E-LDM in an offline manner and read them from the disk during the Gen-DINO training. More details are provided in the supplementary.

Evaluation: We employ standard protocols used in DINO [6], such as the training-free k-nearest neighbor classifier (k-NN) and training a linear classifier (linear-probing) on frozen features. As highlighted in the DINO paper, linear probing is sensitive to hyperparameter variations, and hence we consider k-NN to be the preferred choice for evaluation given its robustness.

5.2. Comparing with DINO on ImageNet-1K

In Tab. 1, we compare the performance of ViT-S (patch size of 16) pre-trained using our Gen-DINO method against the vanilla DINO method with a 100-epoch schedule on the ImageNet-1K validation set. We observe that, compared to DINO, our method performs significantly better on k-NN evaluation, with an improvement of 1.5% in Top-1% accuracy. The linear probing evaluation shows an improvement of 0.5%. This evaluation indicates that Gen-DINO enhances representation learning through generative and interpolated augmentations, particularly by learning to solve the more challenging pretext task of disentangling two objects in the object-centric images found in ImageNet-1K. We further demonstrate the improvements of individual components in ablations (Sec. 5.5).

5.3. Nearest neighbor retrieval

Here, we investigate the effectiveness of Gen-DINO in enhancing performance compared to DINO on tasks that rely on nearest neighbor retrieval. Specifically, we evaluate its impact on image retrieval and copy detection tasks. We closely follow the settings described in DINO [6].

Image Retrieval: We utilized the Revisited [44] Oxford and Paris image retrieval datasets [42]. We used the corresponding Medium (M) and Hard (H) splits with query/database pairs and reported the Mean Average Precision (mAP). In Tab. 2, we compare the performance of ViT-S pre-trained with DINO and Gen-DINO on ImageNet-1K, using them as off-the-shelf frozen encoders for retrieval on these datasets. Following feature extraction, we apply k-

Table 1. Top-1% accuracy on **ImageNet-1K** validation set for ViT-S pre-trained through DINO and Gen-DINO and evaluated using k-NN and linear probing (LP) evaluation. k-NN is a training free evaluation.

Method	Epochs	k-nn	LP
DINO	100	69.4	74.0
Gen-DINO	100	**70.9**	**74.5**

Table 2. **Image retrieval.** We compared the mAP on the Oxford (ROx) and Paris (RPar) datasets using frozen features from ViT-S pre-trained with DINO and Gen-DINO on ImageNet-1K.

Method	Epochs	ROx		RPar	
		M	H	M	H
DINO	100	30.7	10.8	55.6	26.1
Gen-DINO	100	**33.3**	**11.2**	**57.2**	**26.9**

Table 3. **Copy detection.** We report performance (mAP) using the Copydays "strong" subset [14]. We compare the features from ViT-S pre-trained with DINO and Gen-DINO.

Method	Epochs	Dim	mAP
DINO	100	768	80.2
Gen-DINO	100	768	**82.5**

NN for retrieval. We observe that Gen-DINO features outperform DINO features for this retrieval task by up to 2.6% on the medium split and up to 0.8% on the hard split across the two datasets.

Copy Detection: We use the "strong" subset of the INRIA Copydays dataset [14] and report the mean average precision (mAP). The task is to identify images that have been distorted by blur, insertions, print and scan, among other modifications, similar to the protocol in DINO. We perform this task using cosine similarity on the frozen features obtained from ViT-S pre-trained with DINO and Gen-DINO. We use the concatenation of the output [CLS] token and the GeM [45] pooled output patch tokens, resulting in a 768-dimensional descriptor for ViT-S. In Tab. 3, we show that compared to vanilla DINO, our method substantially improves performance by 2.3%.

5.4. Discovering the semantic layout of scenes

Previously, DINO [6] demonstrated the emerging properties of self-supervised ViTs, particularly their ability to explicitly represent scene layouts, with object segmentation visible in the self-attention modules of the last block. Here, we investigate how Gen-SIS' disentanglement pretext task, based on interpolated images, further enhances the model's capability for object segmentation without any supervision.
Video Instance Segmentation: In Tab. 4, we evaluate the segmentation capabilities of self-supervised ViTs with Gen-DINO and compare them to vanilla DINO. Specifically, we used the DAVIS-2017 video instance segmentation benchmark [43]. Following the experimental protocol in DINO, we segment scenes using a nearest-neighbor approach between consecutive frames, utilizing the frozen features for the output patch tokens. We observe that our Gen-DINO pre-trained ViT-S performs significantly better for both the mean region similarity ($\mathcal{J}_m$) and mean contour-based accuracy ($\mathcal{F}_m$) metrics, demonstrating the effectiveness of the disentanglement task in enabling the model to more accurately understand object layout. We also compared it to Gen-DINO without the disentanglement task, i.e., DINO with only generative augmentation, and found that it performed worse than Gen-DINO. This is additional evidence that the disentanglement pretext task improves performance
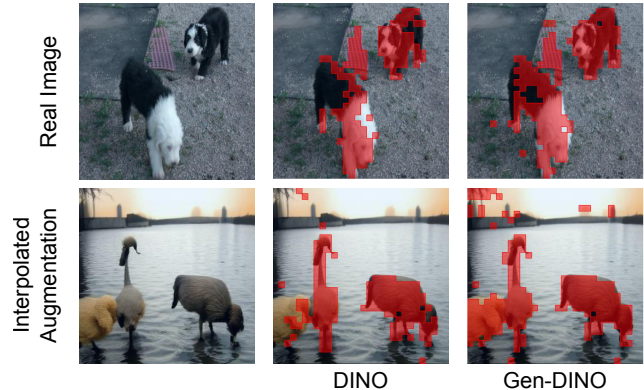

Figure 3. [CLS] token attention map of DINO and Gen-DINO averaged across all heads and overlayed on real and interpolated image. Gen-DINO's attention covers higher portion of object patches than DINO.

in understanding object details.

Table 4. **DAVIS 2017 Video Object Segmentation.** We compared the performance of frozen features from ViT-S pre-trained with DINO and Gen-DINO on ImageNet-1K for the task of video instance tracking. Mean region similarity ($\mathcal{J}_m$) and mean contour-based accuracy ($\mathcal{F}_m$) metrics are reported. We use an image resolution of 480p.

Method	Epochs	$(\mathcal{J}\&\mathcal{F})_m$	$\mathcal{J}_m$	$\mathcal{F}_m$
DINO	100	61.45	59.67	63.23
Gen-DINO w/o disent.	100	61.66	59.87	63.45
Gen-DINO	100	**62.07**	**60.52**	**63.62**

Probing the self-attention map: In Fig. 3, we visualize the self-attention of the [CLS] token overlayed on a sample real image and on a sample interpolated image using pre-trained ViT-S using DINO and our Gen-DINO model. Consistently, for both real and generated images, Gen-DINO's attention map covers the object patches (16×16 regions) more compared to DINO. This was also reflected in the significantly improved mean region similarity $\mathcal{J}_m$ in Tab. 4.

5.5. Ablations

Here, we study the effect of various components in Gen-DINO that are crucial for enhancing the performance of the encoder compared to vanilla DINO SSL. All ablations are conducted using ViT-S pre-trained for 100 epochs on ImageNet-1K and evaluated on its validation set. Top-1% k-NN classifier accuracy is reported.

Importance of Disentanglement pretext task: In Tab. 5, we investigate the effect of using only generative augmentation images without the proposed disentanglement pretext task (Gen-DINO wo/ disent.) and interpolated augmentation, comparing it to vanilla DINO and Gen-DINO. We observe that, by itself, generative augmentation provides a 0.5% improvement compared to the larger 1.5% improvement seen in Gen-DINO over vanilla DINO. This emphasizes that, beyond simple data augmentation, generative models can significantly enhance the SSL framework when used properly (in our case the interpolation augmentation and disentanglement pretext task), motivating future research.

Table 5. Effect of disentanglement (disent.) pretext task in Gen-DINO.

Method	k-NN
DINO	69.4
Gen-DINO wo/ disent.	69.9 (+0.5)
Gen-DINO	**70.9** (+1.5)

Table 6. Effect of interpolation ratio α on Gen-DINO.

α	k-NN
0.2, 0.4, 0.6, 0.8	70.0
0.4, 0.6	70.1
0.5	**70.9**

Effect of Interpolation Ratio: In Tab. 6, we explore the effect of interpolation ratio (α) in our framework. By default, we use $\alpha = 0.5$ for interpolated image generation. However, other values or even randomly chosen values can be used as well. Therefore, we experiment with $\alpha = \{0.2, 0.4, 0.6, 0.8\}$ and $\alpha = \{0.4, 0.6\}$. We found that using values other than $\alpha = 0.5$ reduces the model's performance.

To understand this drop, in Fig. 4, we visualize the generated images with different α values. We observe that for values close to the boundaries (0.2 and 0.8), the interpolation is barely visible, with the image mostly gravitating toward the dominant side, making the pretext task noisy. The images synthesized with values 0.4 and 0.6 are very close to each other making it harder for the model to distinguish the exact α used in interpolation. Furthermore, this can also lead to noisy training if the interpolated images do not exactly reflect the interpolation ratio. We believe this is a limitation of the generative capabilities of the diffusion model for highly diverse datasets like ImageNet-1K [56]. Hence, both intuitively and empirically, using $\alpha = 0.5$ is the optimal solution as the SSL encoder only needs to understand that the interpolated image is a combination of two other images rather than finding the exact interpolation value.

Effect of teacher entanglement position: In Tab. 7, we experiment with the entanglement position of teacher outputs used in the disentanglement pretext task. By default, we entangle the teacher head logits (after the projection head) of two source images as per Eq.5. We also explore performing the entanglement after the teacher backbone (before the projection head) and then passing the entangled embedding into the teacher head. Tab. 7 indicates that entangling before the projection head leads to a significant decrease in performance. This can be attributed to the low-dimensional teacher backbone output (384 in ViT-S), which allows less flexibility in feature entanglement within the low-dimensional space compared to the teacher projection head output, which is in much higher dimension (typically 65K).

Table 7. Effect of teacher entanglement position.

Method	k-nn
Before proj. head	69.6
After proj. head	**70.9**

Table 8. Comparison of DINO and Gen-DINO in case of only global crops.

Method	k-NN
DINO	58.6
Gen-DINO wo/ disent.	64.7
Gen-DINO	**67.4**

Training with Only Global Crops: By default, the DINO method uses multiple local crops and has shown that they are necessary to achieve high performance; therefore, for a fair comparison, we include local crops in our framework as well. In Tab. 8, we compare DINO and our method, Gen-DINO, without local crops. We observe that Gen-DINO performs significantly better, by 8.8%, compared to DINO. These findings could potentially help improve other SSL frameworks such as MoCov2 [12] and BYOL [21], which do not benefit much or may even degrade when using local crops [6], but may improve with our generative and interpolated augmentations.

6. Experiments: Histopathology Imaging

So far, we have evaluated Gen-DINO in the natural image domain, pre-training on the object-centric dataset ImageNet-1K. In this section, we explore its extension to histopathology, which is non-object-centric and instead involves a complex spatial layout of various tissue structures and nuclei types [8, 35]. Given the lack of large-scale text-to-image foundation diffusion models in histopathology, self-augmentations using our Gen-SIS framework have a large potential to improve SSL in this domain.

6.1. Setup

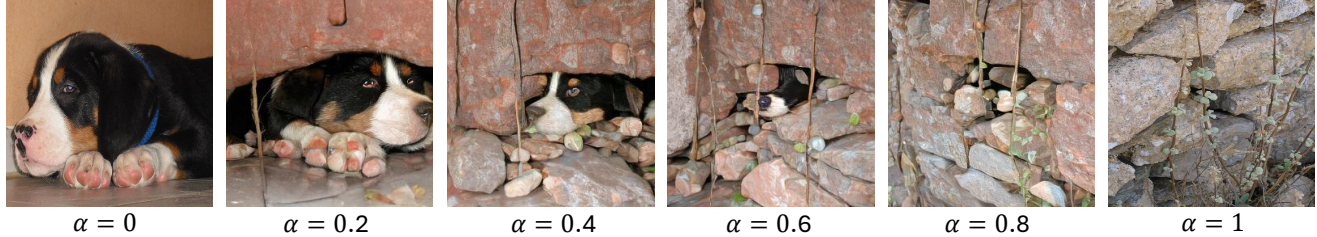**Dataset details:** We test our framework on two histopathology datasets: PANDA [4] and BRIGHT [3]. The

| $\alpha = 0$ | $\alpha = 0.2$ | $\alpha = 0.4$ | $\alpha = 0.6$ | $\alpha = 0.8$ | $\alpha = 1$ |

Figure 4. Interpolated augmentations ($\alpha = \{0.2, 0.4, 0.6, 0.8\}$) generated from 2 real images (α=0 and α=1). An example of interpolation between dog and stone image from ImageNet dataset is illustrated.

PANDA dataset comprises approximately 10K prostate cancer whole-slide images (WSIs) with ISUP grading (6-class classification). The WSIs are sourced from two sites: Karolinska and Radboud. We use the slides from Karolinska for training and the slides from Radboud for evaluation. The BRIGHT dataset is a breast cancer dataset containing 703 WSIs, divided into 424 for training, 80 for validation, and 200 for testing. It features a 3-class classification (Non-cancerous, Pre-cancerous, and Cancerous) task. Due to the current inactivity of the BRIGHT challenge and the unavailability of test set labels, all results are reported using the validation set as the test set.

Patch extraction and training: WSIs are of gigapixel size and, therefore, need to be tiled into multiple crops to fit within hardware constraints. For the BRIGHT dataset, we use $10\times$ magnification (1 micron per pixel), and for the PANDA dataset, we use $20\times$ magnification (0.5 microns per pixel), extracting crops of size 256×256 pixels from each WSI. This yields 2M and 2.1M crops for the train and test splits, respectively, for the PANDA dataset, and 1.2M and 0.2M million crops for the train and test splits, for the BRIGHT dataset. For both datasets, using the crops from the corresponding training set, we first pre-train a ViT-S from scratch with DINO, followed by training an E-LDM conditioned on this encoder. Finally, we pre-train a Gen-DINO using our Gen-SIS framework. We pre-train both DINO and Gen-DINO for 50 epochs on the PANDA dataset and 100 epochs on BRIGHT, using the same setting as ImageNet-1K. Following pre-training, we use the frozen encoders to extract embeddings for each crop in train-test set for both datasets. More details are provided in the supplementary.

MIL setting: Since we only have labels for each WSI, not individual crops, we treat a WSI as a bag of crops. We apply multiple instance learning (MIL) [32, 34, 36, 38, 53], a method traditionally used in this context, to pool crop embeddings from each WSI and perform WSI-level prediction. For this task, we use ABMIL [34]. To ensure robustness, we conduct 5-fold cross-validation on PANDA and report mean performance on the test set. Since BRIGHT is a relatively small dataset in terms of the number of WSIs, we train MIL

with 3 random seeds on the complete training set and report mean performance over the test set. The hyperparameter details used for MIL are provided in the supplementary.

6.2. Results

As observed in Tab. 9, MIL trained with features extracted from the Gen-DINO pre-trained encoder consistently outperforms those from the DINO pre-trained encoder across both datasets. In the PANDA dataset, our method improves performance by more than 3% in balanced accuracy. In the BRIGHT dataset, we observe an improvement of 1.7% in accuracy. It is important to note that the goal of this experiment is not to compare with the performance of foundational models on histopathology, but rather to enhance the DINO SSL method, which is a building block of all recent models in this field [8, 16, 39, 61], with the potential to improve foundational models when Gen-DINO is scaled with larger datasets.

Table 9. Performance of DINO and Gen-DINO on PANDA (6-class classification) and BRIGHT (3-class classification) datasets. For PANDA, we report the mean over 5-fold cross-validation, and for BRIGHT, we report the mean over three seeds.

Method	PANDA			BRIGHT		
	Acc.	F1	AUC	Acc.	F1	AUC
DINO	0.476	0.461	0.817	0.646	0.638	0.820
Gen-DINO	**0.508**	**0.480**	**0.826**	**0.663**	**0.655**	**0.857**

7. Conclusion

We presented Gen-SIS, a self-augmentation technique to enhance self-supervised learning. Self-augmentations are generated from a diffusion model that does not rely on auxiliary information (text or class labels), making our approach a self-contained one. Our enhanced DINO (Gen-DINO) trained with Gen-SIS framework using generative augmentations, and interpolated augmentation along with the disentanglement pretext task outperforms the vanilla

DINO in tasks such as image classification and nearest neighbor retrieval. More importantly, Gen-SIS pretraining enhances the self-supervised ViT's capability to explicitly represent semantic layout, as empirically proven through the video segmentation task. We showed that the disentanglement pretext task was the key contributor in enhancing this capability. We further extended our framework to non-object-centric histopathology images, showing consistent improvement across complex cancer grading tasks compared to DINO. Future work will explore novel approaches for flexible interpolation augmentation, including potential policies for selecting which image pairs to interpolate.

8. Acknowledgements

References

[1] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15619–15629, 2023. 1

[2] Shekoofeh Azizi, Simon Kornblith, Chitwan Saharia, Mohammad Norouzi, and David J Fleet. Synthetic data from diffusion models improves imagenet classification. *Transactions on Machine Learning Research*, 2023. 3

[3] Nadia Brancati, Anna Maria Anniciello, Pushpak Pati, Daniel Riccio, Giosuè Scognamiglio, Guillaume Jaume, Giuseppe De Pietro, Maurizio Di Bonito, Antonio Foncubierta, Gerardo Botti, et al. Bracs: A dataset for breast carcinoma subtyping in h&e histology images. *Database*, 2022: baac093, 2022. 7

[4] Wouter Bulten, Kimmo Kartasalo, Po-Hsuan Cameron Chen, Peter Ström, Hans Pinckaers, Kunal Nagpal, Yuannan Cai, David F Steiner, Hester Van Boven, Robert Vink, et al. Artificial intelligence for diagnosis and gleason grading of prostate cancer: the panda challenge. *Nature medicine*, 28(1):154–163, 2022. 7

[5] Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 5253–5270, 2023. 3

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 1, 2, 3, 5, 6, 7, 16

[7] Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-σ: Weak-to-strong training of diffusion transformer for 4k text-to-image generation. *arXiv preprint arXiv:2403.04692*, 2024. 3

[8] Richard J Chen, Tong Ding, Ming Y Lu, Drew FK Williamson, Guillaume Jaume, Bowen Chen, Andrew Zhang, Daniel Shao, Andrew H Song, Muhammad Shaban, et al. A general-purpose self-supervised model for computational pathology. *arXiv preprint arXiv:2308.15474*, 2023. 7, 8

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2

[10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 1

[11] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020. 5

[12] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 7

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[14] Matthijs Douze, Hervé Jégou, Harsimrat Sandhawalia, Laurent Amsaleg, and Cordelia Schmid. Evaluation of gist descriptors for web-scale image search. In *Proceedings of the ACM international conference on image and video retrieval*, pages 1–8, 2009. 6

[15] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9588–9597, 2021. 2

[16] Alexandre Filiot, Paul Jacob, Alice Mac Kain, and Charlie Saillard. Phikon-v2, a large and public feature extractor for biomarker prediction. *arXiv preprint arXiv:2409.09173*, 2024. 8

[17] Jonas Geiping, Quentin Garrido, Pierre Fernandez, Amir Bar, Hamed Pirsiavash, Yann LeCun, and Micah Goldblum. A cookbook of self-supervised learning. *arXiv preprint arXiv:2304.12210*, 2023. 2

[18] Alexandros Graikos, Srikar Yellapragada, and Dimitris Samaras. Conditional generation from unconditional diffusion models using denoiser representations. *arXiv preprint arXiv:2306.01900*, 2023. 3

[19] Alexandros Graikos, Srikar Yellapragada, Minh-Quan Le, Saarthak Kapse, Prateek Prasanna, Joel Saltz, and Dimitris Samaras. Learned representation-guided diffusion models for large-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8532–8542, 2024. 3, 4, 5, 14

[20] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. 1, 2

[21] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. 2, 7

[22] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 2

[23] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 1

[24] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 2

[25] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 1

[26] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *ICCV*, 2021. 12

[27] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *CVPR*, 2021. 12

[28] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 2, 5

[29] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 3

[30] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 3

[31] Rongjie Huang, Max W. Y. Lam, Jun Wang, Dan Su, Dong Yu, Zhou Zhao, and Yi Ren. SynCLR: A synthesis framework for contrastive learning of out-of-domain speech representations, 2022. 3

[32] Wentao Huang, Xiaoling Hu, Shahira Abousamra, Prateek Prasanna, and Chao Chen. Hard negative sample mining for whole slide image classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 144–154. Springer, 2024. 8

[33] Drew A Hudson, Daniel Zoran, Mateusz Malinowski, Andrew K Lampinen, Andrew Jaegle, James L McClelland, Loic Matthey, Felix Hill, and Alexander Lerchner. Soda: Bottleneck diffusion models for representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23115–23127, 2024. 4

[34] Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *International conference on machine learning*, pages 2127–2136. PMLR, 2018. 8, 17

[35] Saarthak Kapse, Srijan Das, Jingwei Zhang, Rajarsi R Gupta, Joel Saltz, Dimitris Samaras, and Prateek Prasanna. Attention de-sparsification matters: Inducing diversity in digital pathology representation learning. *Medical Image Analysis*, 93:103070, 2024. 7

[36] Saarthak Kapse, Pushpak Pati, Srijan Das, Jingwei Zhang, Chao Chen, Maria Vakalopoulou, Joel Saltz, Dimitris Samaras, Rajarsi R Gupta, and Prateek Prasanna. Si-mil: Taming deep mil for self-interpretability in gigapixel histopathology. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11226–11237, 2024. 8

[37] Minh-Quan Le, Alexandros Graikos, Srikar Yellapragada, Rajarsi Gupta, Joel Saltz, and Dimitris Samaras. ∞-brush: Controllable large image synthesis with diffusion models in infinite dimensions. *arXiv preprint arXiv:2407.14709*, 2024. 3

[38] Ming Y Lu, Drew FK Williamson, Tiffany Y Chen, Richard J Chen, Matteo Barbieri, and Faisal Mahmood. Data-efficient and weakly supervised computational pathology on whole-slide images. *Nature biomedical engineering*, 5(6):555–570, 2021. 8

[39] Ming Y Lu, Bowen Chen, Drew FK Williamson, Richard J Chen, Ivy Liang, Tong Ding, Guillaume Jaume, Igor Odintsov, Long Phi Le, Georg Gerber, et al. A visual-language foundation model for computational pathology. *Nature Medicine*, 30(3):863–874, 2024. 8

[40] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 1, 2, 12

[41] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 2, 3

[42] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008. 5

[43] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 6

[44] Filip Radenović, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Revisiting oxford and paris: Large-scale image retrieval benchmarking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5706–5715, 2018. 5

[45] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1655–1668, 2018. 6

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

[47] Sucheng Ren, Huiyu Wang, Zhengqi Gao, Shengfeng He, Alan Yuille, Yuyin Zhou, and Cihang Xie. A simple data mixing prior for improving self-supervised learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14595–14604, 2022. 12

[48] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2, 3, 4

[49] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 3

[50] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 2

[51] Mannat Singh, Quentin Duval, Kalyan Vasudev Alwala, Haoqi Fan, Vaibhav Aggarwal, Aaron Adcock, Armand Joulin, Piotr Dollár, Christoph Feichtenhofer, Ross Girshick, et al. The effectiveness of mae pre-pretraining for billion-scale pretraining. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5484–5494, 2023. 1

[52] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2, 3, 4, 5

[53] Kevin Thandiackal, Boqi Chen, Pushpak Pati, Guillaume Jaume, Drew FK Williamson, Maria Gabrani, and Orcun Goksel. Differentiable zooming for multiple instance learning on whole-slide images. In *European Conference on Computer Vision*, pages 699–715. Springer, 2022. 8

[54] Yonglong Tian, Lijie Fan, Phillip Isola, Huiwen Chang, and Dilip Krishnan. Stablerep: Synthetic images from text-to-image models make strong visual representation learners. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 3

[55] Brandon Trabucco, Kyle Doherty, Max A Gurinas, and Ruslan Salakhutdinov. Effective data augmentation with diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. 2, 3

[56] Clinton Wang and Polina Golland. Interpolating between images with diffusion models. *Workshop on Challenges in Deployable Generative AI at International Conference on Machine Learning (ICML), Honolulu, Hawaii*, 2023. 4, 7

[57] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, pages 10506–10518, 2019. 12

[58] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14668–14678, 2022. 2

[59] Srikar Yellapragada, Alexandros Graikos, Prateek Prasanna, Tahsin Kurc, Joel Saltz, and Dimitris Samaras. Pathldm: Text conditioned latent diffusion model for histopathology. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5182–5191, 2024. 14

[60] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021. 2

[61] Eric Zimmermann, Eugene Vorontsov, Julian Viret, Adam Casson, Michal Zelechowski, George Shaikovski, Neil Tenenholtz, James Hall, Thomas Fuchs, Nicolo Fusi, et al. Virchow 2: Scaling self-supervised mixed magnification models in pathology. *arXiv preprint arXiv:2408.00738*, 2024. 8

Gen-SIS: <u>Gen</u>erative <u>S</u>elf-augmentation <u>I</u>mproves <u>S</u>elf-supervised Learning

Supplementary Material

The supplementary is organized as follows:
- Robustness evaluation (section 9)
- Comparison with pixel disentanglement (section 10)
- Implementation details (section 11)

9. Robustness evaluation

To evaluate the robustness of Gen-DINO, we benchmark its performance on three challenging datasets: ImageNet-A (Im-A) [27], ImageNet-R (Im-R) [26], and ImageNet-Sketch (Im-S) [57]. These datasets test the model's resilience to out-of-distribution (OOD) variations. Im-A includes 7,500 adversarially filtered images across 200 classes of ImageNet. Im-R contains 30,000 images of renditions that are different from standard images from 200 classes of ImageNet. Sketch contains 50,000 black-and-white sketch images from all ImageNet classes. We directly evaluate the linear classifier trained on ImageNet-1K on these datasets. As shown in Tab. 10, Gen-DINO demonstrates improvements over the baseline on two robustness benchmarks. It achieves a substantial accuracy improvement on Im-R, increasing from 33.25 to 37.98, and a decent improvement on Sketch, rising from 61.93 to 62.3. These results suggest that the generative and interpolated augmentations in Gen-SIS enhance the model's ability to handle OOD images. Gen-DINO learns to encode more robust features, better capturing important characteristics of the images even under distribution shifts.

Table 10. **Robustness.** We evaluate the linear classifier trained on ImageNet-1K. Gen-DINO shows notable improvements on ImageNet-R (Im-R) and Sketch (Im-S), indicating an enhanced ability to generalize to diverse image variations.

Method	Epochs	Im-A	Im-R	Im-S
DINO	100	9.24	33.25	61.93
Gen-DINO	100	9.24	**37.98**	**62.30**

10. Comparison with pixel disentanglement

An important question to address is whether a generative model is the optimal way to interpolate between images, or if simpler techniques, such as pixel-level interpolation, could achieve similar results. To investigate this, we perform an ablation comparing Gen-SIS's interpolated augmentations, performed in the conditioning space of E-LDM, against pixel-level interpolation of real images. In this regard, we train DINO with the same disentanglement pretext task (as proposed in Eq.7 of the main text) but replace embedding space interpolation with pixel-level interpolation. We refer to this model as "*DINO w/ pixel disent.*"

As shown in Tab. 11, *DINO w/ pixel disent.* significantly underperforms Gen-DINO by 3.0% in terms of $k-$NN evaluation. This performance gap highlights the importance of interpolated augmentations in Gen-SIS, performed through E-LDM's conditioning space (embedding space). Interestingly, *DINO w/ pixel disent.* improves linear probing accuracy over vanilla DINO by 0.59% and achieves comparable performance to Gen-DINO in this metric. Improvement in linear probing of 0.4 % over DINO has also been observed by a previous work [47] that integrates interpolating real images in pixel space into DINO training. However, as noted by the authors of DINO, linear probing results are highly sensitive to hyperparameter tuning. Consequently, we prioritize k-NN evaluation as a more reliable metric. k-NN evaluation is training-free and provides a direct measure of the quality of learned representations, as its performance correlates with other downstream tasks like image retrieval and copy detection, which rely on nearest-neighbor comparisons in embedding space. The authors of DINOv2 [40] also emphasize using $k-$NN over linear probing to ablate key design choices.

In Fig. 5, we visualize the interpolated augmentation under the Gen-SIS framework versus pixel-level interpolation. In Gen-SIS, the E-LDM blends the pencil (Image 1) and grasshopper (Image 2) to form a new object whose shape is similar to pencils, but color and texture follow the grasshopper. In pixel-level interpolation, the resulting textures and shapes are very different from the ones seen in the training images; (i) the edges are less prominent, due to the misaligned blending of the two images, and (ii) the textures are 'unnatural' with mixtures of colors between the two images creating faded textures. Overall,

we posit that the E-LDM tries to synthesize an image with objects formed from coherent blending of features from source image objects instead of the abruptly blended samples that pixel-level interpolation produces.

Table 11. Top-1% accuracy on **ImageNet-1K** using DINO, DINO w/ pixel disent., and Gen-DINO. We report k-NN and linear probing (LP) evaluation.

Method	Epochs	k-NN	LP
DINO	100	69.4	73.97
DINO w/ pixel disent.	100	67.9	74.56
Gen-DINO	100	70.9	74.49

Interpolated Augmentation

Image 1	Image 2	Ours	Pixel

Figure 5. Interpolated augmentation using Gen-SIS framework (Ours) vs pixel-level interpolation. Image 1 and Image 2 are the source images used for interpolation ($\alpha = 0.5$).

11. Implementation details

11.1. Generation of self-augmentations

For ImageNet-1K, we synthesize four generative augmentations for each real image and save them to disk. We sample a random synthetic image out of four when training Gen-DINO. Fig. 6 shows sample synthetic image generation by E-LDM when using embedding from a single real image as conditioning. Synthetic images generated from E-LDM contain variations in orientation, object shape, and background compared to real images. In the case of interpolated augmentation, for each real primary image in the dataset, we pick a random secondary real image out of the whole dataset and perform the interpolated augmentation. We create a single interpolated augmentation for each primary image and interpolation ratio (α), and then read the interpolated augmentation from the disk when training. Fig. 10 presents the interpolated augmentation with various α values. We use α=0.0 and α=1.0 to represent the two real images used as sources for interpolated augmentation. Interpolated augmentations blend the shape, texture, and color of the objects visible in the two source images to form new, blended objects. As seen in Fig. 10, a key observation is that for α=0.2 and α=0.8 interpolated images are very similar to the closest source image and contain negligible components from the other end. Following the ablation in Tab.6 (in main text), we use α=0.5 in the training of Gen-DINO. For both generative and interpolated augmentation, we use classifier-free guidance of 6 with 50 DDIM steps.

In the case of histopathology (PANDA and BRIGHT datasets), we follow a similar setup as ImageNet-1K, and synthesize one generative augmentation and one interpolated augmentation for each image. Fig. 7 and Fig. 8 present generative augmentations, i.e., sample synthetic images generated using real images as source. In generative augmentation, the synthetic image varies in terms of the position and orientation of cells and tissue compared to the real source image. In the case of interpolation augmentation, for each real primary image (crop) in the dataset, we pick a random secondary real image (crop) from a different whole slide image and perform the interpolated augmentation. We create a single interpolated image for each primary image and given interpolation ratio (α) and read the interpolated image from the disk when training. Fig. 11 and Fig. 12 showcase the interpolated augmentations in PANDA and BRIGHT datasets, respectively. Unlike ImageNet, we observe that even α=0.2 and α=0.8 interpolated images contain some components from lower-weighted source images. Following this observation, we sample a random alpha from {0.2, 0.4, 0.6, 0.8} for the interpolated augmentation during the training of Gen-DINO. For both generative and interpolated augmentation, we use a guidance weight of 1.75 with 50 DDIM
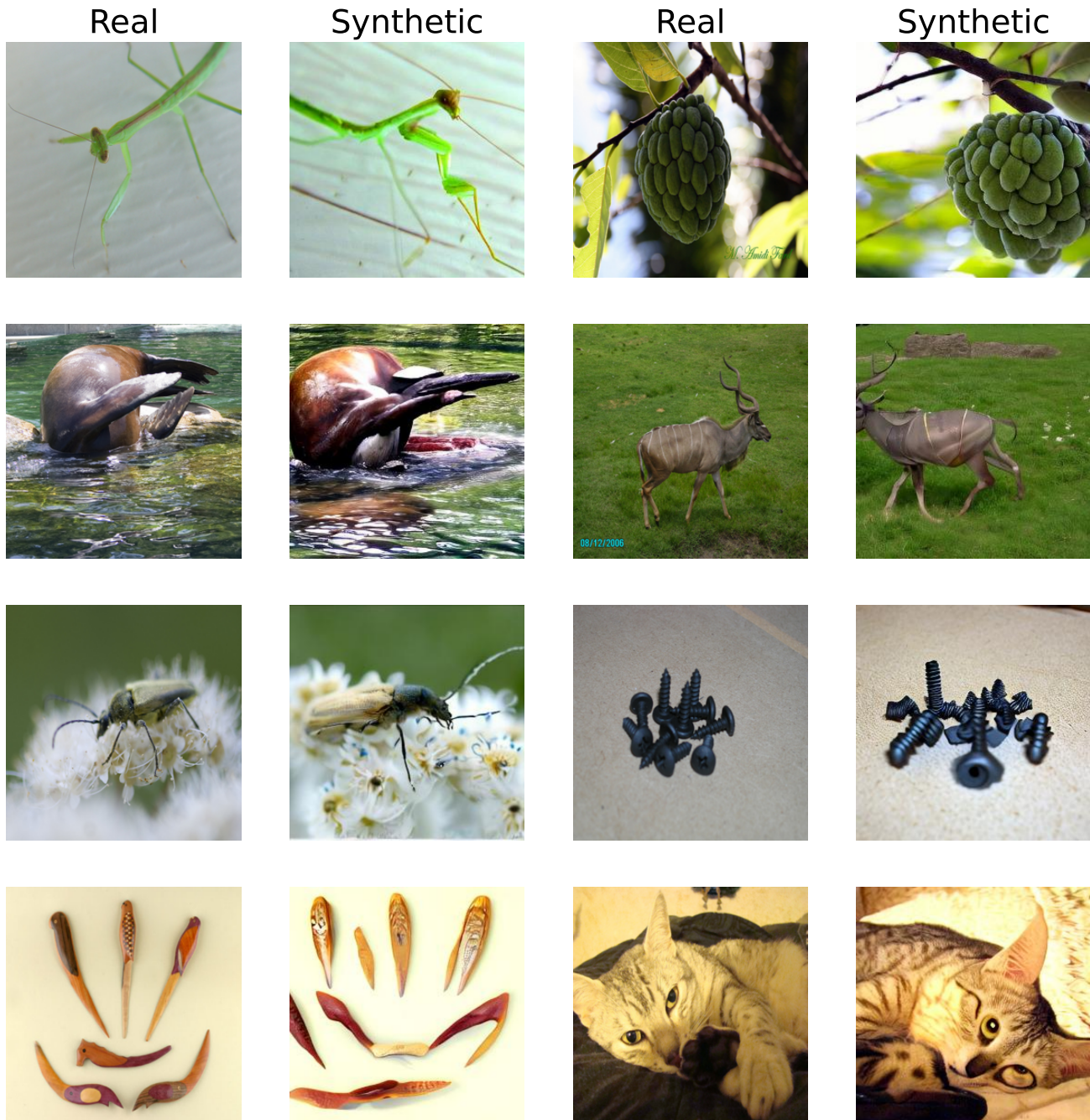
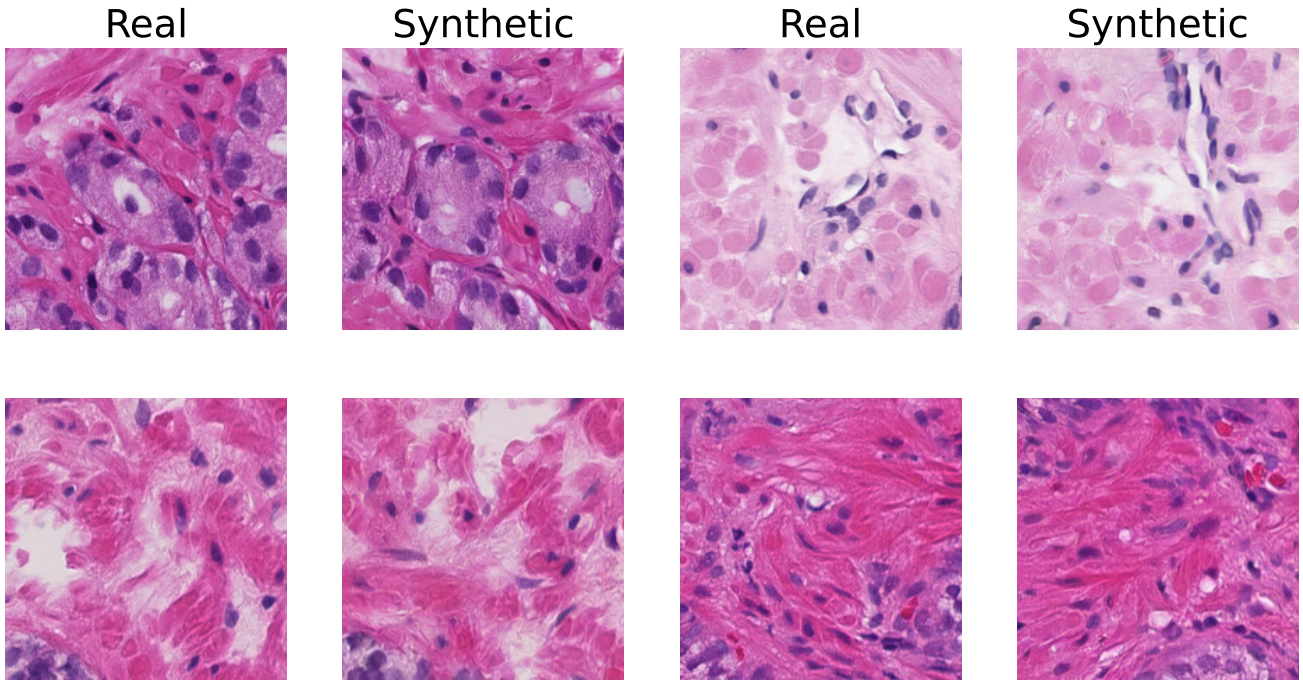13

| Real | Synthetic | Real | Synthetic |

Figure 6. Generative Augmentation on ImageNet-1K using E-LDM by conditioning it on a single real image's embedding. Real: denotes the real image in the dataset, Synthetic: denotes the generative augmentation.
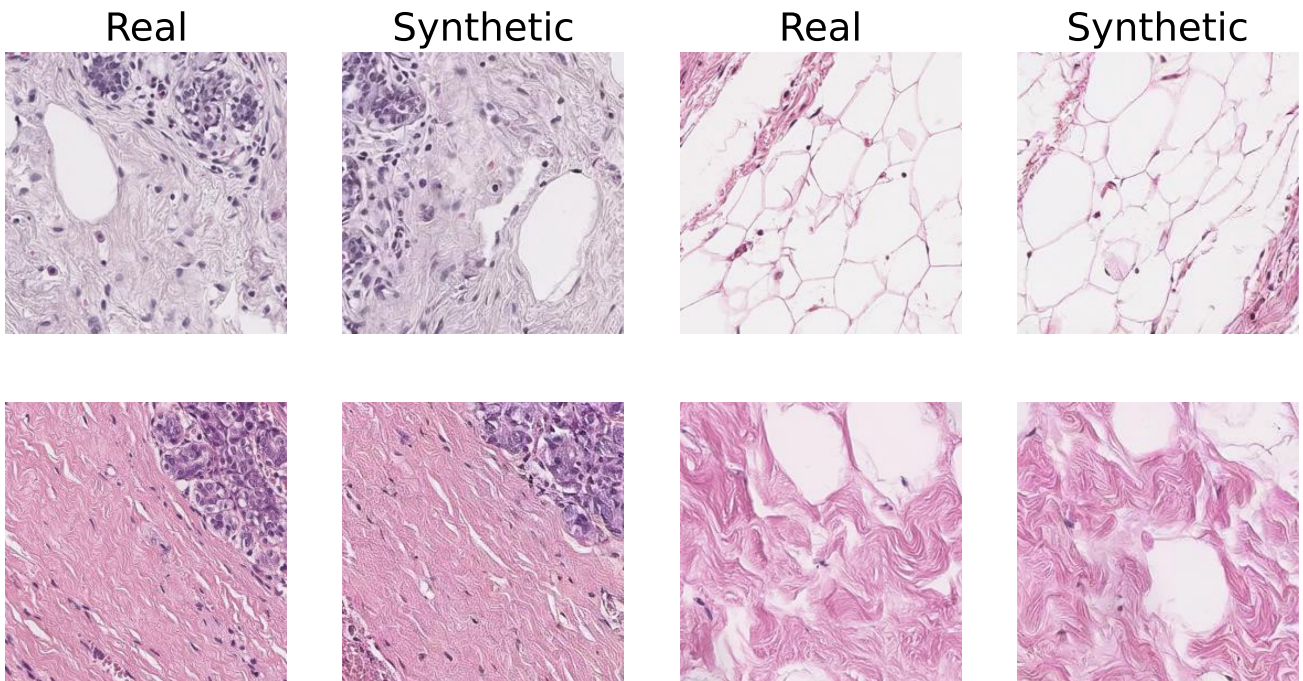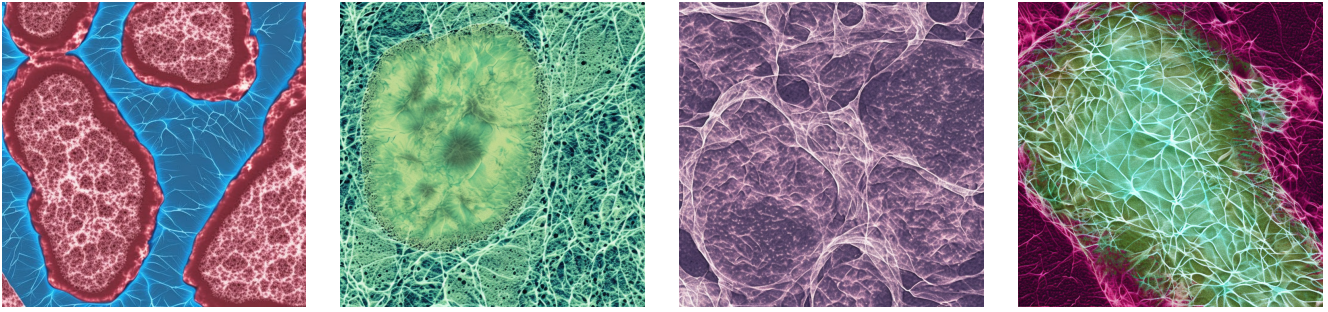
steps following the recent works [19, 59] on diffusion models in histopathology. We also present sample synthetic breast cancer images generated from Stable diffusion with text prompts in Fig 9. The images are highly inaccurate to be used in training. This reinforces our key design choice of using E-LDM for augmentations.

11.2. Pseudo code for disentanglement pretext task

Algorithm 1 presents the pseudo-code for the disentanglement pretext task. We only use global crops for this pretext task.

Figure 7. Generative Augmentation on PANDA using E-LDM by conditioning it on a single real image's embedding. Real: denotes the real image in the dataset, Synthetic: denotes the generative augmentation.

Figure 8. Generative Augmentation on BRIGHT using E-LDM by conditioning it on single real image's embedding. Real: denotes the real image in the dataset, Synthetic: denotes the generative augmentation.

Stable Diffusion 2.1

prompt = "*a digital histopathology image showing cancerous breast tissue*"

Figure 9. Breast cancer synthetic histopathology image generation using Stable Diffusion with text prompt as conditioning. The images generated do not resemble real breast cancer images that are found in typical datasets like BRIGHT (Fig 8).

Algorithm 1: PyTorch-style pseudo-code for disentanglement pretext task

```
# Input image:  img_1
# gs, gt:  student and teacher networks
# tps, tpt:  student and teacher temperatures
# c:  center
# alpha:  interpolation ratio
for img_1 in loader
    # Read secondary source image
    img_2 = ReadImage(secondary(img_1))
    # Read interpolated image of primary and secondary source image
    img_int = ReadInterpImage(img_1, img_2, alpha)
    # Apply vanilla dino augmentation to form a view of interpolation
    img_int_view = vanilla_augment(img_int)
    # Apply vanilla dino augmentation to form a view of primary
    img_1_view = vanilla_augment(img_1)
    # Apply vanilla dino augmentation to form a view of secondary
    img_2_view = vanilla_augment(img_2)
    # Get student output for interpolated image and teacher output for image 1 and image 2
    stud_int = gs(img_int_view)
    teach_1 = gt(img_1_view).detach()
    teach_2 = gt(img_2_view).detach()
    # Student sharpening
    stud_int = softmax(stud_int / tps, dim=1)
    # Entanglement of teacher output
    teach_ent = alpha * teach_1 + (1-alpha) * teach_2
    # Teacher sharpening and centering
    teach_ent = softmax((teach_ent - c) / tpt, dim=1)
    # Compute disentanglement loss
    disentanglement_loss = - (teach_ent * log(stud_int)).sum(dim=1).mean()
```

11.3. Evaluation details

ImageNet: We employ standard protocols as used in DINO [6], such as the training-free k-nearest neighbor classifier (k-NN) and the learning of a linear classifier, both applied to frozen features. For k-NN evaluation, we extract the features from the training data using the frozen pre-trained encoder. Next, the k-NN classifier compares the features of an image to the k nearest stored features and assigns a label. We explore various numbers of nearest neighbors and determine that 10-NN or 20-NN consistently yields the best results. In linear evaluation, random resize cropping and horizontal flip augmentation are applied during training, and test performance is reported on a central crop. We follow the same hyperparameter setup as DINO [6]. We perform a learning rate hyperparameter search to find the optimal choice. As highlighted in the DINO paper, linear probing is sensitive to hyperparameter variations, and we similarly observe a substantial variance in accuracy across learning rate. Therefore, in our study, we consider k-NN as a preferable choice for evaluation, given its robustness to challenges like hyperparameter tuning.

Histopathology: We employ multiple instance learning (MIL) to aggregate the frozen features of crops from a whole slide image, followed by a linear classifier applied to the pooled features. For our MIL framework, we utilize ABMIL [34]. The model is trained for 50 epochs using the AdamW optimizer with a learning rate of 0.0001 and a weight decay of 0.01. Given that whole slide images can contain varying numbers of crops, we use a batch size of 1 and accumulate gradients over 8 steps, achieving an effective batch size of 8.

α=0.0	α=0.2	α=0.4	α=0.5	α=0.6	α=0.8	α=1.0

Figure 10. Interpolated augmentations at various interpolating ratios on ImageNet-1K. α=0.0 and α=1.0 denote the two real images used as sources for interpolation. We interpolate the embeddings of the two source images, and then condition the E-LDM using the interpolated embedding to synthesize interpolated augmentations.
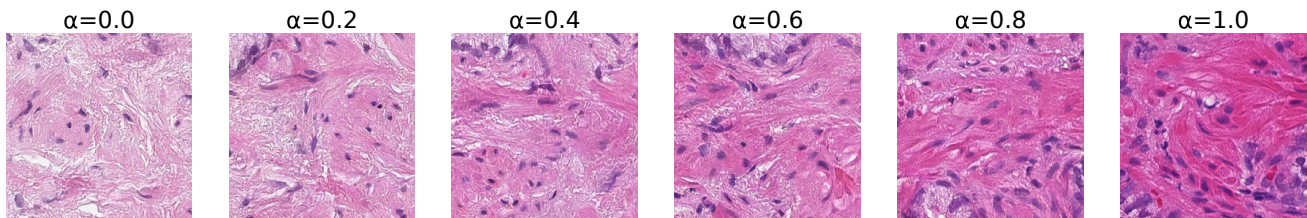
(a) : This image shows the transition between normal prostate stroma (α=0.0) and low-grade prostate (α=1.0) cancer. α=0.0 shows an image of prostate stroma. α=0.2 shows a partial gland in the lower right corner. α=0.4 shows a gland that can easily be identified as cancer. The gland lacks a basal cell layer, it has a sharp luminal border and the lumen is filled with secretions. More glands are visible in α=0.8, all of them meeting the morphological criteria of low-grade cancer.

(b) : This image shows the transition between a vascular structure in the stroma surrounded by fibroblasts, myofibroblasts and smooth muscle cells and another stromal patch that consists almost entirely of perpendicularly sectioned smooth muscle fibers.
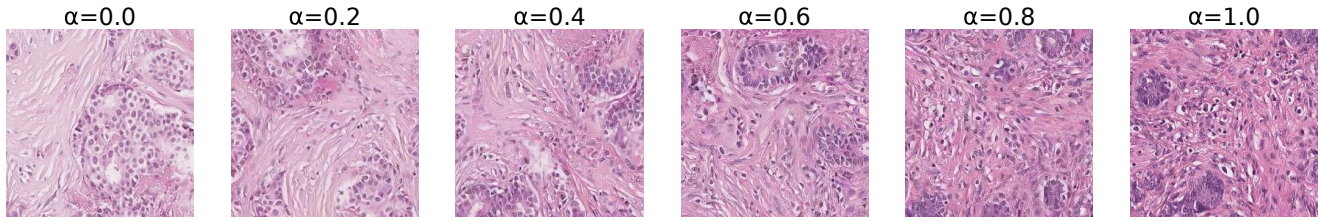

(c) : This image shows the transition of a large benign gland to a tile with prostate stroma. The amount of benign epithelium diminishes gradually.
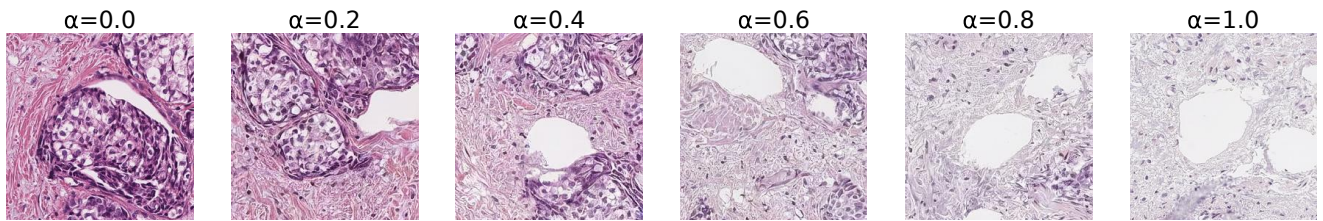

(d) : This image shows the transition of a loose extracellular matrix containing a few fibroblasts to a dense cellular stroma with fragments of benign glands. The most apparent fragment of a gland appears in α=0.6.
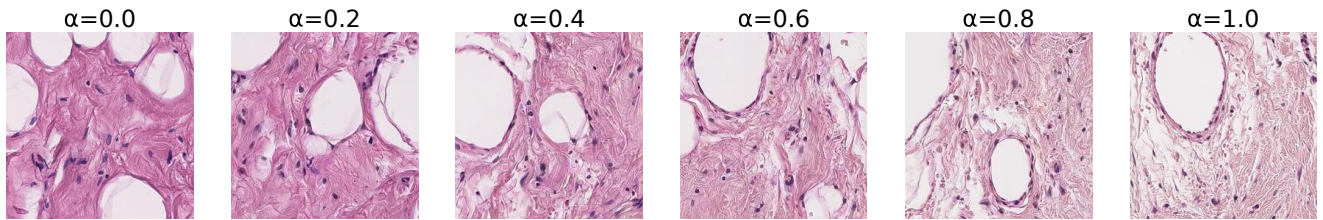
Figure 11. Interpolated augmentations at various interpolating ratios on PANDA. α=0.0 and α=1.0 denote the two real images used as sources for interpolation. We interpolate the embeddings of the two source images, and then condition the E-LDM using the interpolated embedding to synthesize interpolated augmentations. The captions below each row represent the description of interpolation annotated by a pathologist.
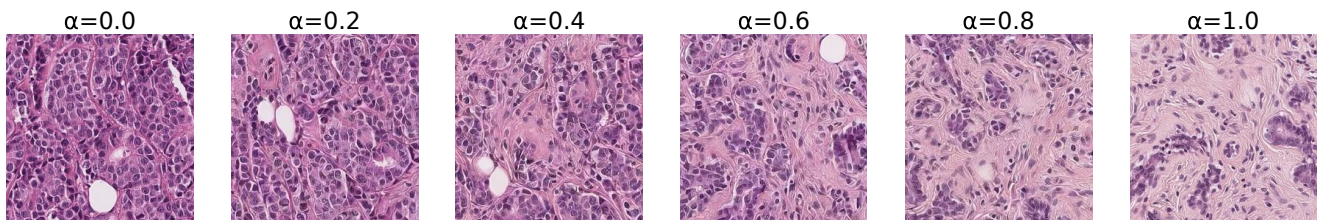
(a) : This image shows the transition of a patch containing well-demarcated epithelial glands with adjacent acellular stroma to a patch with cellular, inflamed stroma surrounding small glandular structures.

(b) : This image shows a transition between a patch with a glandular epithelial structure and a patch of acellular stroma with only a few morphological details.

(c) : This image shows the transition between a patch that contains stroma and adipose tissue and a patch with a vascular structure.

(d) : This image shows the transition between a patch with closely spaced epithelial glandular structures and a patch showing a few small glands embedded in an inflamed stroma. α=0.6 and 0.8 show a few lymphocytes in the stroma. The arrangement and morphology of the glands raise the possibility of cancer.

Figure 12. Interpolated augmentations at various interpolating ratios on BRIGHT. α=0.0 and α=1.0 denote the two real images used as sources for interpolation. We interpolate the embeddings of the two source images, and then condition the E-LDM using the interpolated embedding to synthesize interpolated augmentations. The captions below each row represent the description of interpolation annotated by a pathologist.