# Gen-SIS: Generative Self-augmentation Improves Self-supervised Learning

Supplementary Material

## 9. Technical Appendix

### 9.1. Robustness

We train a linear classifier on ImageNet-1K using frozen features of DINO and Gen-DINO, following the protocol of [40]. To evaluate the robustness of Gen-DINO, we benchmark its performance on three challenging ImageNet variation datasets: ImageNet-A (Im-A) [26], ImageNet-R (Im-R) [25], and ImageNet-Sketch (Im-S) [58]. Im-A includes 7,500 adversarially filtered images across a 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. These datasets test the model's robustness to out-of-distribution (OOD) variations.

As shown in Tab. 10, Gen-DINO demonstrates improvements over the baseline on two out of three robustness benchmarks. It achieves substantial accuracy gains on Im-R, from 33.25 to 37.98, and a decent improvement on Sketch, 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. By leveraging the interpolation task, Gen-DINO learns to encode images into more robust features that better capture the 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 enhanced ability to generalize to diverse image variations.

| Method | Im-1K | Im-A | Im-R | Im-S |
|---|---|---|---|---|
| DINO | 73.97 | 9.24 | 33.25 | 61.93 |
| Gen-DINO | 74.49 | 9.24 | **37.98** | **62.30** |

### 9.2. Gen-DINO with RCG

In Tab. 11, we evaluate Gen-SIS framework by training Gen-DINO using the Moco [22] conditioned ImageNet pretrained DiT model from RCG [36] as the E-LDM. We generate self-augmentations using the same setting as our E-LDM. As shown in the Tab. 11, the performance with off-the-shelf RCG's ELM is close to training with our E-LDM. This suggests that we can avoid the cost of training the diffusion model in Gen-SIS by using existing available E-LDMs pretrained on the same dataset.

Table 11. Top-1% $k$-NN accuracy on **ImageNet-1K** using DINO, Gen-DINO w/ RCG's E-LDM, and Gen-DINO w/ our E-LDM.

| Method | Epochs | $k$-NN |
|---|---|---|
| DINO | 100 | 69.4 |
| Gen-DINO w/ RCG's E-LDM | 100 | 70.7 |
| Gen-DINO w/ our E-LDM | 100 | 70.9 |

### 9.3. Comparison with pixel-level 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. 12, *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 12. 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 |

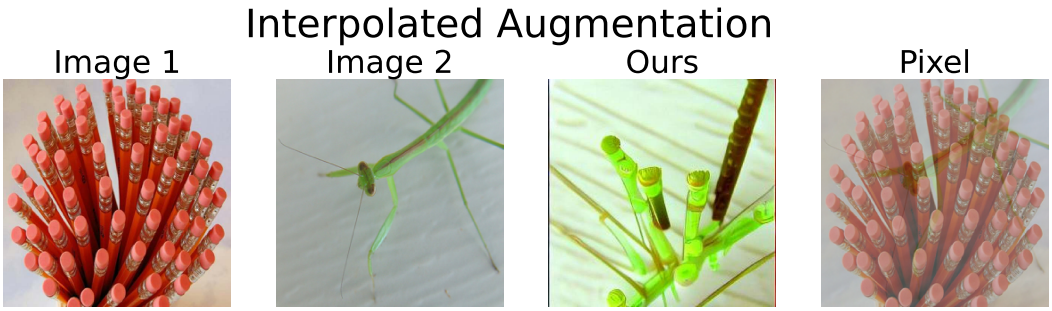



Figure 5. Interpolated augmentation using Gen-SIS framework (Ours) vs pixel-level interpolation. Image 1 and Image 2 are the source images used for interpolation ($\alpha = 0.5$).

### 9.4. Effect on the learned embedding space

In Fig. 6 we employ UMAP [39] to visualize the learned embedding space of DINO and the proposed Gen-DINO framework. Using three vastly different classes, we map the DINO embeddings to 2D to visualize them. Then, using the same mapping, we map the embeddings of generated interpolated samples between those classes. We believe that the proposed disentanglement pretext task helps the model better disambiguate between the newly introduced interpolated samples. We also hypothesize that just by introducing linearly-interpolated synthetic images during training, we 'fill' holes in the embedding manifold, creating new possible paths between clusters in the embedding space.

Although one could argue that linear interpolation may be suboptimal, depending on the embedding manifold we aim to learn, it already shows improvements over the original DINO, as shown in our empirical results. In future work we aim to explore different interpolation methods that follow the geodesics on the manifold, which could potentially further unlock the benefits of generative augmentations in self-supervised learning.

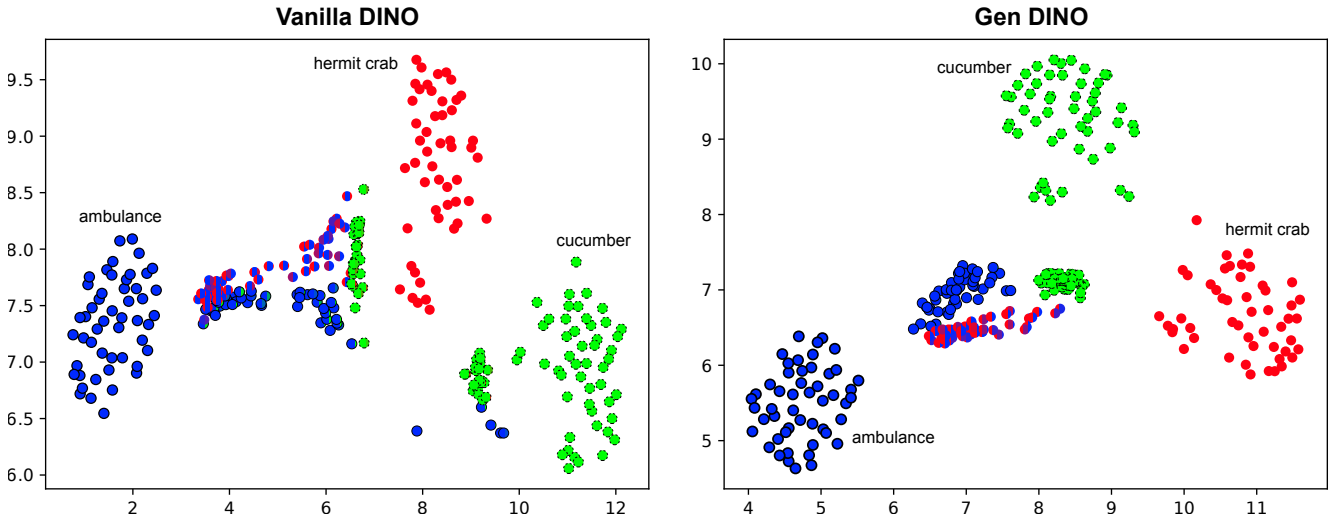



Figure 6. Visualization of embedding space of DINO and Gen-DINO using UMAP. We visualize real data samples from three distinct classes (ambulance, hermit crab, cucumber) along with interpolated samples using our E-LDM. The interpolated samples fall between the classes. Gen-DINO has learned to embed the interpolated samples better than the vanilla DINO model.

## 9.5. Implementation details

### 9.5.1. E-LDM training

The E-LDM configuration includes a VQ-f4 autoencoder that downsamples images from $256 \times 256 \times 3$ to $64 \times 64 \times 3$, and a U-Net denoiser which we train from scratch. We set the learning rate to $10^{-4}$ with a warmup period of 1000 steps. To generate images, we use DDIM sampler [52] with 50 steps and apply classifier-free guidance [27].

### 9.5.2. 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. 7 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 ($\alpha$), and then read the interpolated augmentation from the disk when training. Fig. 11 presents the interpolated augmentation with various $\alpha$ values. We use $\alpha$=0.0 and $\alpha$=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. 11, a key observation is that for $\alpha$=0.2 and $\alpha$=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.7 (in main text), we use $\alpha$=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. 8 and Fig. 9 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 ($\alpha$) and read the interpolated image from the disk when training. Fig. 12 and Fig. 13 showcase the interpolated augmentations in PANDA and BRIGHT datasets, respectively. Unlike ImageNet, we observe that even $\alpha$=0.2 and $\alpha$=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 steps following the recent works [18, 60] on diffusion models in histopathology. We also present sample synthetic breast

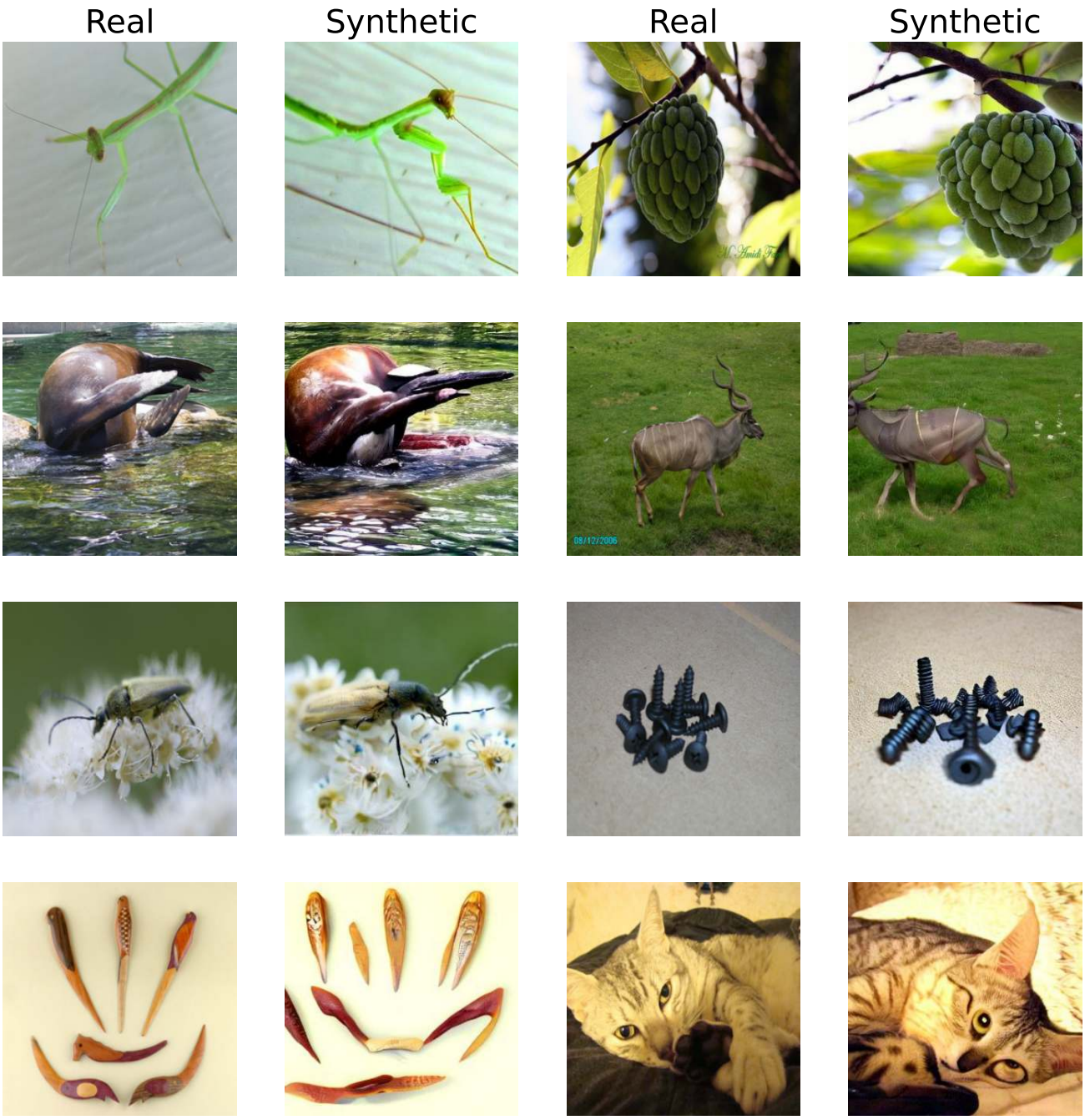



Figure 7. Generative Augmentation on ImageNet-1K using E-LDM by conditioning it on a single real image's embedding. Real: denotes the real image in the dataset, Synthetic: denotes the generative augmentation.

cancer images generated from Stable diffusion with text prompts in Fig 10. The images are highly inaccurate to be used in training. This reinforces our key design choice of using E-LDM for augmentations.

#### 9.5.3. SSL methods:

We demonstrate the effectiveness of the proposed Gen-SIS training using DINO [7], I-BOT [62], DINOv2 [40], and SimCLR [11] SSL methods. I-BOT extends DINO with an additional masked image modeling pretext task. DINOv2 builds on DINO with additional losses like *ibot* loss and *koleo* loss. SimCLR is a contrastive learning based SSL approach. For all

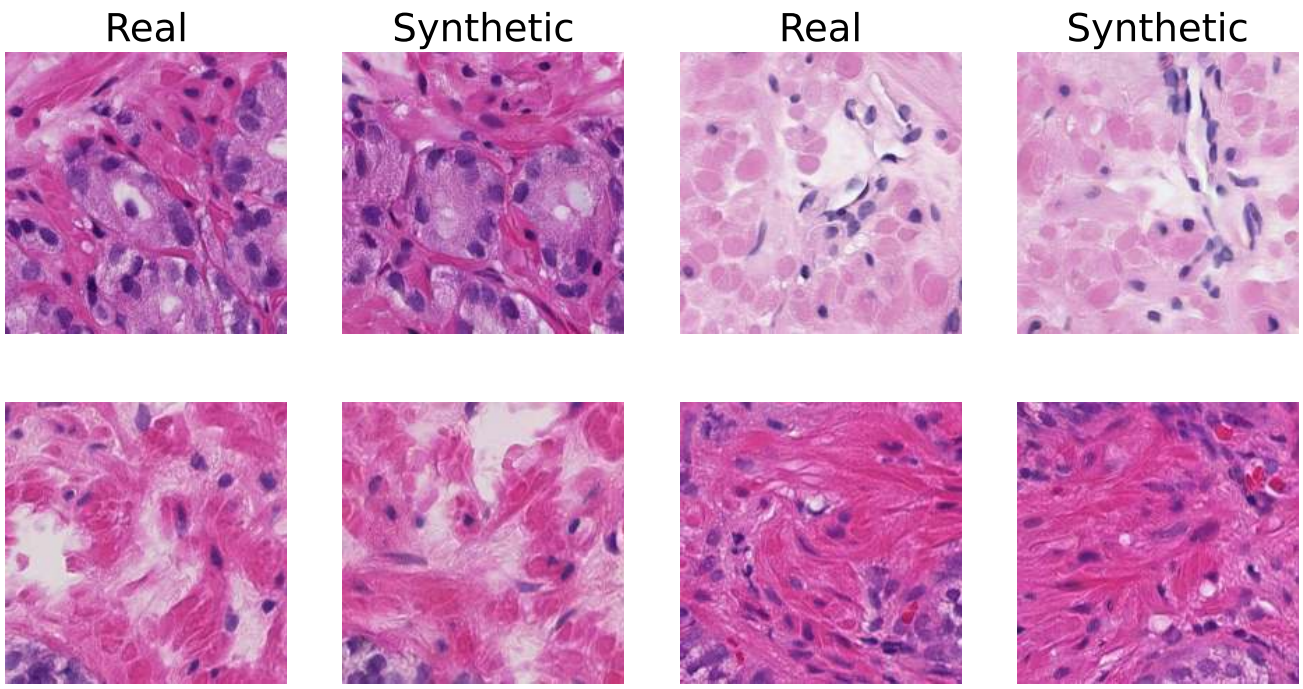



Figure 8. Generative Augmentation on PANDA using E-LDM by conditioning it on a single real image's embedding. Real: denotes the real image in the dataset, Synthetic: denotes the generative augmentation.

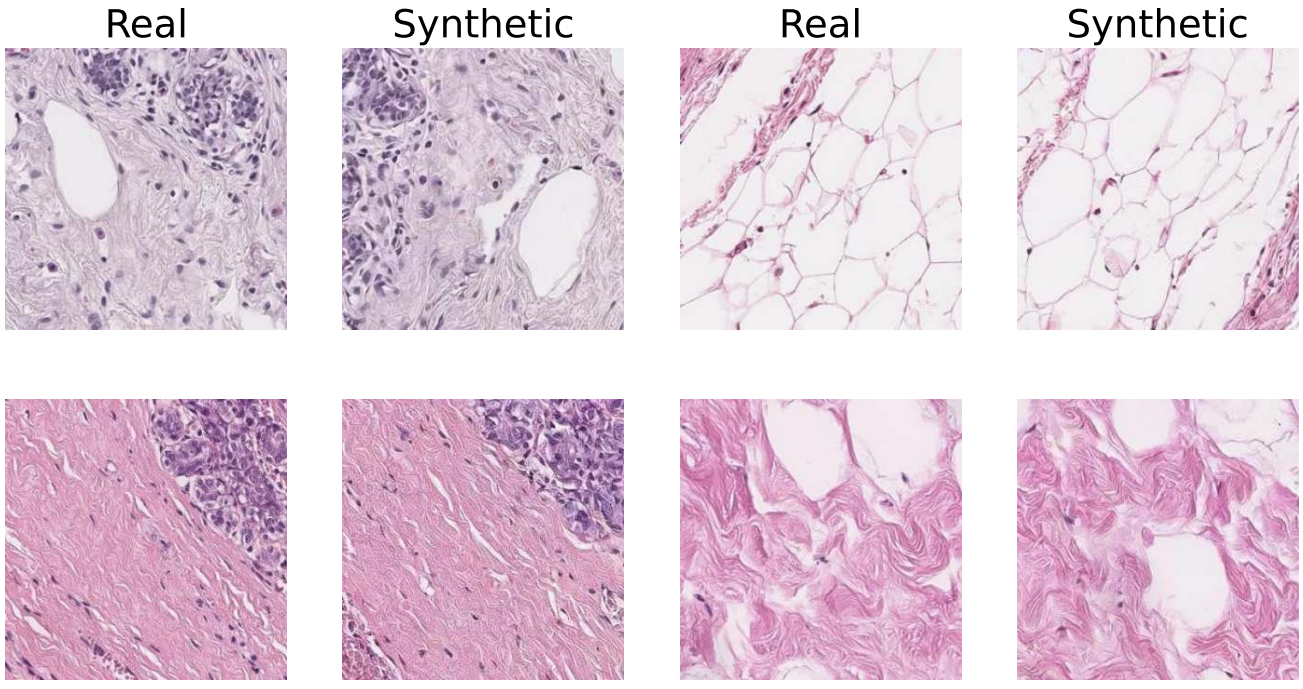
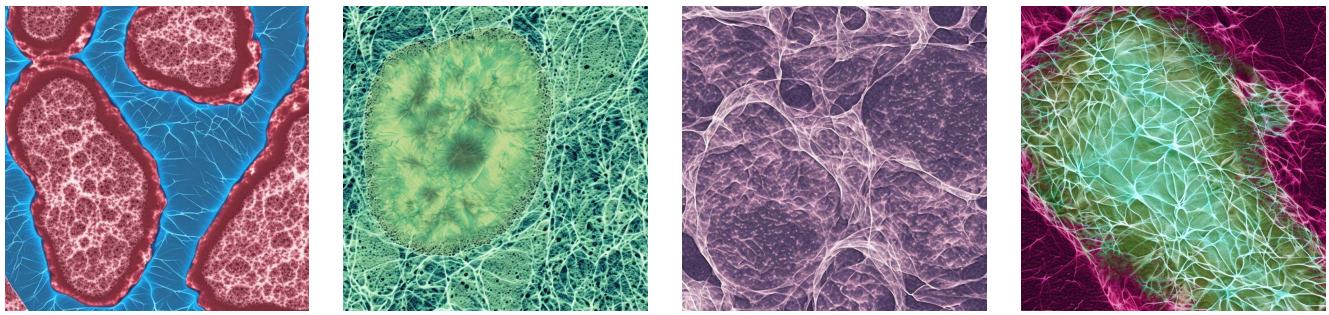



Figure 9. Generative Augmentation on BRIGHT using E-LDM by conditioning it on single real image's embedding. Real: denotes the real image in the dataset, Synthetic: denotes the generative augmentation.

methods except SimCLR, we build our code using the original repositories maintaining the hyperparameters (e.g., batch size

Stable Diffusion 2.1



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

Figure 10. 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 9).

of 1024). For SimCLR, we build using the SimCLR implementation from StableRep [55] as the original repository does not include training with transformer models. Following the original settings, we use 8 local crops for DINO, I-BOT, and DINOv2. Given that I-BOT and DINOv2 have patch level losses, we apply the disentanglement loss only at image level cls token without altering the patch level losses. The DINOv2 codebase does not provide training hyperparameters for ViT-S when training from scratch, hence we use the default hyperparameters (provided by DINOv2 repository) for our vanilla DINOv2 ViT-S training and achieve $k$-NN accuracy of 69.41%. Our Gen-DINOv2 improves the performance to 70.92% using the proposed self-augmentations and disentanglement pretext task.

For SimCLR, we adapt the disentangle pretext task to work with its contrastive learning objective. Given two source images $I_1, I_2$, and their interpolated image $I_{int}$ generated with interpolating ratio ($\alpha = 0.5$), we pass them through the encoder to extract features $f_1, f_2, f_{int}$. We treat $I_1$ and $I_2$ as positive views for $I_{int}$ in the SimCLR contrastive loss, while treating every other image in the batch as negative views for $I_{int}$.

$$\mathcal{L}^{SimCLR}_{disentangle} = -\alpha \times \log \frac{\exp(\text{sim}(f_{int}, f_1)/\tau)}{\sum_k \exp(\text{sim}(f_{int}, f_k)/\tau)} - (1-\alpha) \times \log \frac{\exp(\text{sim}(f_{int}, f_2)/\tau)}{\sum_k \exp(\text{sim}(f_{int}, f_k)/\tau)} \tag{8}$$

#### 9.5.4. ImageNet Evaluation details:

We employ standard protocols as used in DINO [7], 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 [7]. 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.

#### 9.5.5. ImageNet Downstream details:

**Copy Detection:** We use the "strong" subset of INRIA Copydays dataset [13] 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 DINO. We perform this task using cosine similarity on the frozen features obtained from ViT-S pre-trained with DINO and Gen-DINO for 100 epochs. 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.

**Image Retrieval:** We use the Revisited [44] Oxford and Paris image retrieval datasets [42]. We report the Mean Average Precision (mAP) on the corresponding Medium (M) and Hard (H) splits with query/database pairs. In Table 4, we compare the

performance of ViT-S pre-trained with DINO and Gen-DINO on ImageNet-1K, using them as off-the-shelf feature extractors, and apply $k$-NN for retrieval.

**Video Instance Segmentation:** In Table 5, we evaluate the segmentation capabilities of ViTs pretrained with vanilla DINO and Gen-DINO. The objective is to investigate how Gen-SIS's disentanglement pretext task further enhances the model's capability for object segmentation without supervision. We used the DAVIS-2017 video instance segmentation benchmark [43], and segment scenes using a nearest-neighbor approach between consecutive frames, utilizing the frozen features for the output patch tokens [7].

#### 9.5.6. Histopathology Evaluation details:

**Dataset:** We test our framework on two histopathology datasets: PANDA [5] and BRIGHT [4]. The 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.

**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/pixel), and for the PANDA dataset, we use $20\times$ magnification (0.5 microns/pixel), extracting crops of size $256 \times 256$ 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 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 all encoders for 50 epochs on the PANDA dataset and 100 epochs on BRIGHT, using the same hyperparameters as ImageNet-1K. Following pre-training, we use the frozen encoders to extract embeddings for each crop in train-test set for both datasets.

**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) [30, 32, 34, 37, 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 [32]. 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 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.

#### 9.5.7. 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.

#### 9.5.8. Machine details:

We use an NVIDIA DGX A100-based cluster for all our experiments. The DGX A100 node comprises eight NVIDIA A100 GPUs providing 320 GB of memory in total. The node contains two AMD EPYC 7742 CPUs with 256 cores and 1 TB of DDR4 memory. The machine uses an SSD for data storage, which offers up to 25 GB/s in bandwidth. In total, we used 1000 node hours for the paper, including exploratory experiments.

### 9.6. Societal Impact

Although generative models have significantly advanced the field of computer vision, they can have potential negative societal consequences. For example, they can be used to generate deepfakes. Gen-SIS, which incorporates a generative model component, inherently faces similar challenges. Furthermore, we use publicly available datasets for the training of all components, and Gen-SIS inherits the biases present in the pretraining datasets.

**Algorithm 1:** PyTorch-style pseudo-code for the proposed 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()
```

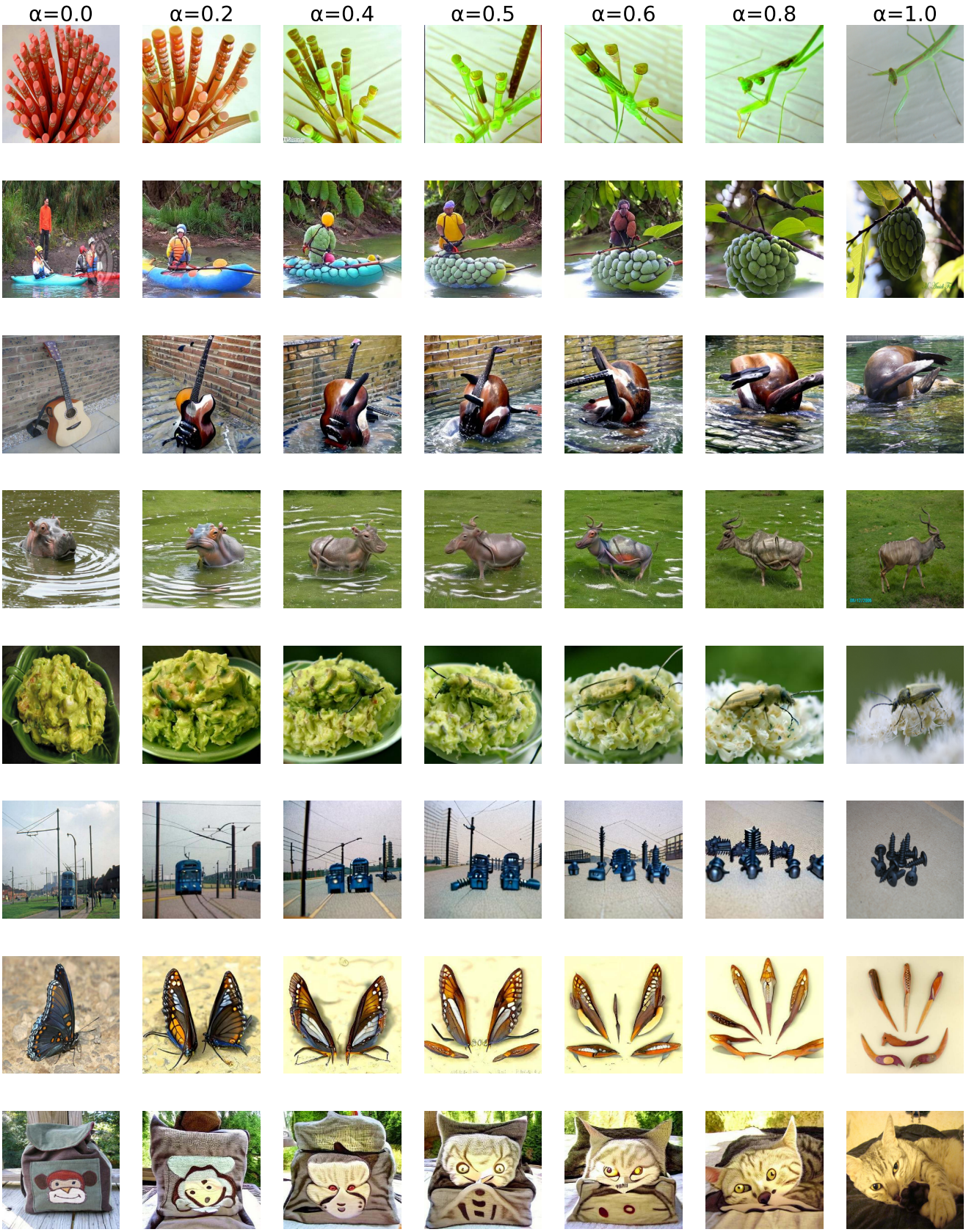
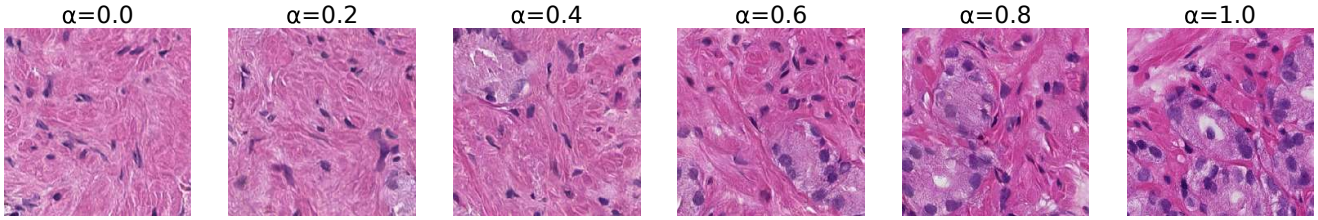
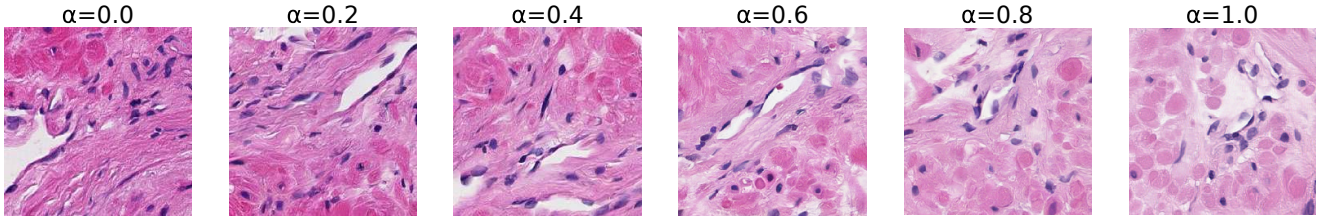


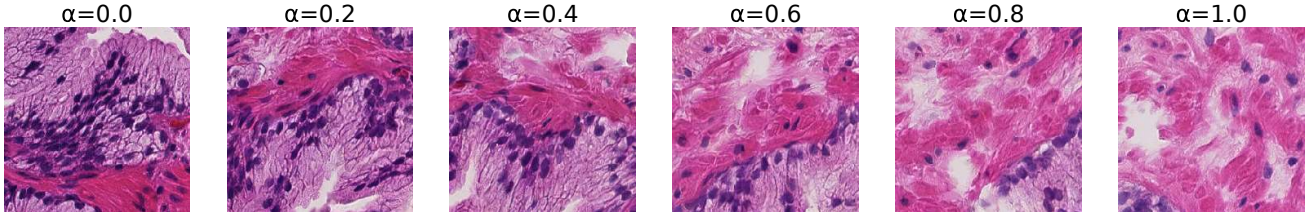

Figure 11. Interpolated augmentations at various interpolating ratios on ImageNet-1K. $\alpha$=0.0 and $\alpha$=1.0 denote the two real images used as sources for interpolation. We interpolate the embeddings of the two source images, and then condition the E-LDM using the interpolated embedding to synthesize interpolated augmentations.



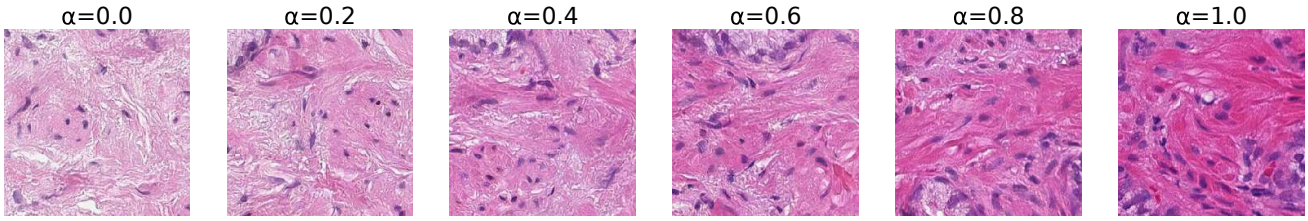

(a) : This image shows the transition between normal prostate stroma (α=0.0) and low-grade prostate (α=1.0) cancer. α=0.0 shows an image of prostate stroma. α=0.2 shows a partial gland in the lower right corner. α=0.4 shows a gland that can easily be identified as cancer. The gland lacks a basal cell 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.

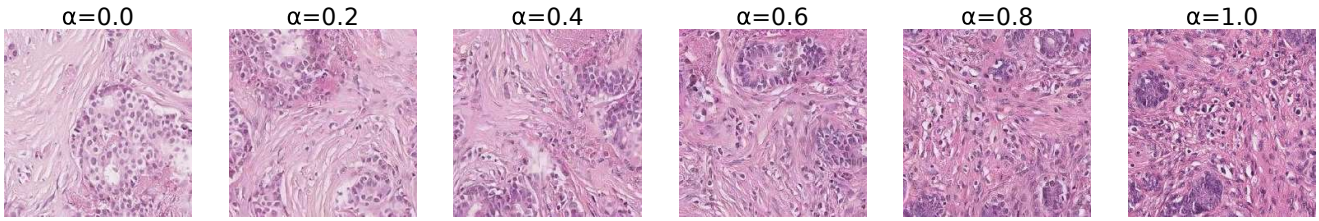



(c) : This image shows the transition of a large benign gland to a tile with prostate stroma. The amount of benign epithelium diminishes gradually.

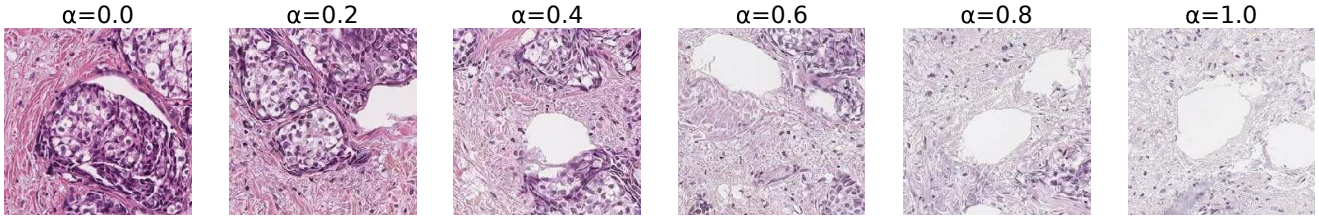



(d) : This image shows the transition of a loose extracellular matrix containing a few fibroblasts to a dense cellular stroma with fragments of benign glands. The most apparent fragment of a gland appears in α=0.6.

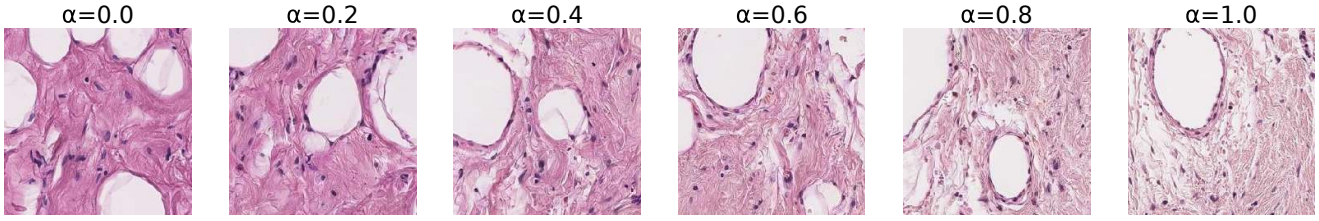
Figure 12. Interpolated augmentations at various interpolating ratios on PANDA. $\alpha$=0.0 and $\alpha$=1.0 denote the two real images used as sources for interpolation. We interpolate the embeddings of the two source images, and then condition the E-LDM using the interpolated embedding to synthesize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



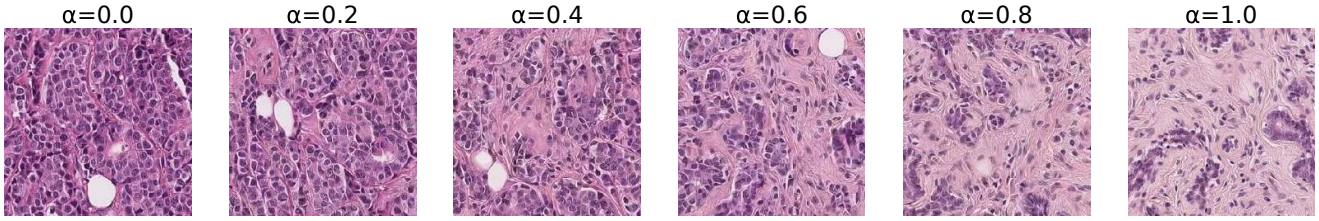

(a) : This image shows the transition of a patch containing well-demarcated epithelial glands with adjacent acellular stroma to a patch with cellular, inflamed stroma surrounding small glandular structures.




(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 13. Interpolated augmentations at various interpolating ratios on BRIGHT. $\alpha$=0.0 and $\alpha$=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.