7. Introduction

This appendix provides additional materials and experimental details. For consistency, we adopt the definitions and notations introduced in Section 3.2.

8. Datasets

CAMELYON16 is a large-scale WSI benchmark developed to evaluate automated methods for detecting metastatic breast cancer in lymph nodes [2]. Designed to challenge detection systems, it remains a standard for assessing histopathology algorithms. The dataset comprises two classes—normal and tumor—with tumor regions typically occupying only a small fraction of positive WSIs, making classification particularly challenging. CAMELYON16 provides an official test split with verified slide-level labels.

SICAP-MIL is a publicly available WSI dataset curated to benchmark MIL-based methods for prostate cancer grading [39]. It contains biopsy slides from 271 patients, with one slide excluded due to poor quality. The slides were scanned at $40\times$ and tiled into overlapping 512×512 patches at $10\times$ resolution. Each slide is globally labeled by expert pathologists with both primary and secondary Gleason grades, indicating dominant tumor patterns. In addition, proportional constraints describe the relative prevalence of each grade, enabling constrained MIL formulations to approach the performance of fully supervised models. All slides are labeled as normal or abnormal, with abnormal cases annotated by Gleason grades (GG3, GG4, GG5). Tumor regions are present in a large portion of these abnormal WSIs.

Dataset distributions are summarized in Table 5.

9. Additional Experimental Setup

CAMELYON16: For this dataset, we partition the Whole Slide Images (WSIs) into a grid of non-overlapping 256×256 tiles at $20\times$ magnification. To isolate tissue regions, background areas are identified and removed following the criteria outlined in [31]. Adhering to the official competition protocol [2], models are trained using a specific subset of 40 WSIs and subsequently evaluated on the entire test set. This setup is designed to simulate practical clinical scenarios where annotated data is scarce, a notable challenge for complex or rare cancer subtypes [48, 58]. The tiling procedure generates a dataset of approximately 2 million patches, corresponding to an average of 7900 patches per slide. For statistical robustness, each model architecture is trained five independent times from random initialization, and we report the mean and standard deviation across all evaluation metrics.

SICAP-MIL: In this dataset, we process the initial 512×512 patches by subdividing each one into four smaller 256×256 patches. This methodology yields a total of approximately $34{,}000$ patches for training and evaluation. On average, each slide contributes about 100 patches to the final dataset. Following the standard procedure detailed in [39], every model is trained independently from scratch in five separate runs. The final results are presented as the mean and standard deviation computed over these five trials for all reported performance metrics.

The full list of augmentations used in ASC and their descriptions is provided in Table 6.

10. Multitask Metrics

Consider a multi-task classification setting with Γ binary classification tasks and n samples. For each sample i, let the ground-truth label vector be $\mathbf{y}_i \in \{0,1\}^\Gamma$ and the model's prediction be $\hat{y}_{i\gamma} \in \{0,1\}$.

Macro-AUC: For a single task γ, the Receiver Operating Characteristic (ROC) curve is obtained by sweeping a decision criterion over the scores $\{\hat{s}_{1\gamma},\dots,\hat{s}_{n\gamma}\}$ and plotting the resulting True-Positive Rate (TPR) against the False-Positive Rate (FPR). The Area Under this ROC curve is

$$\mathrm{AUC}_\gamma \;=\; \int_0^1 \mathrm{TPR}_\gamma(\mathrm{FPR})\, d(\mathrm{FPR}), \tag{6}$$

typically estimated numerically (e.g., trapezoidal rule). Macro-AUC averages these task-wise areas:

$$\text{Macro-AUC} \;=\; \frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}\mathrm{AUC}_\gamma. \tag{7}$$

A model that ranks every positive instance of every task above all negatives achieves Macro-AUC $= 1$ [55]. Because the mean is taken over tasks rather than samples, each task contributes equally, reducing the influence of task-specific class imbalance.

10.1. Hamming Accuracy

Hamming Accuracy measures the proportion of correctly classified label–task pairs:

$$\mathrm{HA} \;=\; \frac{1}{n\Gamma}\sum_{i=1}^{n}\sum_{\gamma=1}^{\Gamma}\mathbf{1}\big(\hat{y}_{i\gamma}=y_{i\gamma}\big). \tag{8}$$

A perfectly accurate model satisfies $\hat{y}_{i\gamma} = y_{i\gamma}$ for every (i,γ), yielding HA $= 1$ [35]. Unlike *subset accuracy*, which requires every label of a sample to be correct simultaneously, Hamming Accuracy rewards each correct label prediction individually, providing a smoother and more informative metric when many labels are present.

11. Augmentations

A summary of our high-magnitude augmentation operations can be seen in Figure 6, illustrated on randomly sampled images.

	SICAP							CAMELYON16		
Set	NC	GS6	GS7	GS8	GS9	GS10	Total	Normal	Tumor	Total
Train	77	10	61	7	25	8	188	23	17	40
Validation	19	2	26	5	10	2	64	33	21	54
Test	17	9	28	13	27	4	98	80	49	129
Total	**113**	**21**	**115**	**25**	**62**	**14**	**350**	**136**	**87**	**223**

Table 5. Comparison of dataset distributions for SICAP and CAMELYON16.

11.1. Magnitude Mapping

Let the actual magnitude range for an augmentation function τ be centered at a_τ with a maximum deviation of b_τ, forming the interval $[a_\tau - b_\tau, a_\tau + b_\tau]$. The value a_τ represents a zero-effect magnitude, corresponding to an identity transformation.

Given the input scale factor $m \in [0, 1]$, the actual magnitude for such symmetric augmentations is sampled as magnitude $= a_\tau + \sigma m b_\tau$, where σ is either fixed to be 1 or -1 (for example, to provide rotation augmentations in a clockwise or counter-clockwise direction) for each set of embeddings.

For augmentations with a one-sided range, such as Solarize, which operate within an interval like $[a_\tau, a_\tau + b_\tau]$, the magnitude is calculated directly as: magnitude $= a_\tau + m b_\tau$.

12. Bayesian Search Details

We employ Bayesian Optimization (BO) to systematically search for the optimal set of maximum augmentation magnitudes, $\{m_\tau^{\max}\}_{\tau \in \mathcal{A}}$. BO is a sample-efficient global optimization strategy ideal for expensive black-box functions. It iteratively builds a probabilistic surrogate model of the objective function and uses an acquisition function to select promising hyperparameter configurations to evaluate next. The overall procedure and its specific components are detailed below.

12.1. Black-Box Function

The optimization is guided by a black-box function g, which represents the entire MIL training and validation pipeline. This function takes a configuration of maximum augmentation magnitudes $\{m_\tau^{\max}\}_{\tau \in \mathcal{A}}$ as input and returns a single scalar value which we aim to minimize: $g : [0, 1]^{|\mathcal{A}|} \to \mathbb{R}$. The output of g is the final validation loss. For each evaluation, the MIL model is trained using the TrivialInterpolate policy, where augmentation magnitudes for each type τ are sampled from the range $[0, m_\tau^{\max}]$. The goal of the optimization is to find the set of magnitudes that minimizes this validation loss.

12.2. Search Space

The search space is defined over the maximum magnitudes $m_\tau^{\max}$ for the $|\mathcal{A}| = 10$ augmentation types used in our framework. To make the search more efficient, each continuous parameter $m_\tau^{\max} \in [0, 1]$ is discretized into a set of 11 uniformly spaced values: $m_\tau^{\max} \in \{0, 0.1, 0.2, \ldots, 1.0\}$. This results in a 10-dimensional discrete search space containing 11^{10} possible configurations of maximum augmentation magnitudes.

12.3. Optimization Details

Acquisition Function. We use the Expected Improvement (EI) [23] acquisition function to guide the selection of new candidates at each iteration. EI is a standard and widely-used acquisition function that provides a strong theoretical foundation for balancing the exploration of new, uncertain regions of the search space with the exploitation of regions already known to yield low loss.

Number of Trials. The optimization is run for a total of 120 trials, which was found to provide a strong balance between performance and computational cost. The process is initialized with 24 trials drawn from a Sobol quasi-random sequence [40] to ensure a broad, low-discrepancy exploration of the search space before the Gaussian Process surrogate model begins to guide subsequent evaluations.

Implementation. The entire Bayesian optimization procedure is implemented using the Ax Platform [4], a modern tool for adaptive experimentation. Ax leverages BoTorch [1] as its underlying library for probabilistic modeling, handling the Gaussian Process surrogate model and the optimization of the EI acquisition function.

13. Additional Ablations

13.1. DINOASC vs DINO

In this section, we evaluate the effect of applying each augmentation individually, i.e., randomly selecting either the strongly augmented embedding of a given augmentation or the original embedding during MIL training. Table 7 shows that each individual augmentation yields notable improve-

Operation Name	Description	Range
ShearX(Y)	Apply a shear transformation along the horizontal (vertical) axis by a factor of *magnitude*.	$[-0.3, 0.3]$
TranslateX(Y)	Shift the image by *magnitude* pixels in the horizontal (vertical) direction.	$[-150, 150]$
Rotate	Rotate the image around its center by *magnitude* degrees.	$[-30, 30]$
Solarize	Flip the values of pixels whose intensity exceeds the threshold *magnitude*.	$[0, 256]$
Contrast	Scale contrast so *magnitude*=0 produces a uniform gray image, and *magnitude*=1 is unchanged.	$[0.1, 1.9]$
Color	Modify color saturation: *magnitude*=0 yields grayscale, *magnitude*=1 leaves the image unchanged.	$[0.1, 1.9]$
Brightness	Adjust overall brightness so *magnitude*=0 is black and *magnitude*=1 is the original image.	$[0.1, 1.9]$
Sharpness	Alter image clarity: *magnitude*=0 gives a heavily blurred image, *magnitude*=1 is unmodified.	$[0.1, 1.9]$

Table 6. Image augmentation operations and their corresponding ranges of magnitudes.

ments when used with DINOASC. However, the table also demonstrates that applying augmentations in isolation does not improve performance over the baseline for either DINOASC or DINO, and further introduces high variance. This can be attributed to two factors: (1) strong augmentations employ maximum magnitudes defined for natural images, which may not be suitable for histopathology, and (2) relying on a single fixed augmentation fails to capture the diversity inherent in histopathology images.

13.2. ASC and SSL Methods

The ASC loss and our Augmentation via Interpolation technique are not designed to replace specific SSL methods, but rather to imbue the embedding space with AugSev Consistency. This property enables the application of augmentations with arbitrary magnitudes during MIL training, aiming to increase sample diversity. To demonstrate this, we use SOTA methods such as Masked Autoencoders (MAE) [15] as a case study. As shown in Table 4, applying our technique to standard MAE embeddings yields only marginal improvements. This limited performance stems from the lack of explicit AugSev Consistency in standard embeddings; without it, substituting m-augmented embeddings with m-interpolated ones introduces semantic noise rather than realistic diversity. This indicates that our interpolation technique is most effective specifically when AugSev consistency is enforced in the embedding space.

Algorithm 1: Sampling in Trivial Interpolate

Input: Original embedding e for a patch (or bag of patches), Precomputed strong embeddings $\{e_\tau\}_{\tau \in \mathcal{A}}$ (at magnitude 1.0), Upper bounds $\{m_\tau^{\max}\}_{\tau \in \mathcal{A}}$.

Output: Augmented embedding e' (or bag of augmented embeddings).

if *with probability* $\frac{|A|}{|A|+1}$ **then**
  Sample τ uniformly from $\mathcal{A}$
  Sample $m \sim \mathcal{U}(0, m_\tau^{\max})$
  Set $e' = (1 - m) \cdot e + m \cdot e_\tau$
else
  Set $e' = e$ `/* No augmentation */`
end
return e'

Augmentation	DINOASC		DINO	
	AUC	ACC	AUC	ACC
No Augmentation	$0.922_{.031}$	$0.837_{.048}$	$0.893_{.042}$	$0.833_{.059}$
TranslateX	$0.817_{.172}$	$0.778_{.090}$	$0.621_{.212}$	$0.656_{.117}$
TranslateY	$0.813_{.171}$	$0.773_{.088}$	$0.620_{.212}$	$0.654_{.117}$
Solarize	$0.787_{.205}$	$0.783_{.119}$	$0.604_{.205}$	$0.646_{.114}$
Color	$0.815_{.171}$	$0.757_{.152}$	$0.678_{.166}$	$0.651_{.110}$
Contrast	$0.800_{.193}$	$0.765_{.174}$	$0.582_{.180}$	$0.649_{.080}$
Sharpness	$0.839_{.161}$	$0.770_{.125}$	$0.617_{.143}$	$0.667_{.054}$
Brightness	$0.831_{.196}$	$0.793_{.170}$	$0.486_{.079}$	$0.589_{.021}$
Rotate	$0.809_{.235}$	$0.767_{.203}$	$0.585_{.272}$	$0.705_{.071}$
ShearX	$0.837_{.214}$	$0.778_{.210}$	$0.688_{.074}$	$0.677_{.047}$
ShearY	$0.825_{.184}$	$0.783_{.199}$	$0.693_{.068}$	$0.670_{.059}$

Table 7. Performance of various augmentations using the DINOASC and DINO backbones. All experiments are conducted on the CAMELYON16 dataset with DSMIL as the MIL-pooling architecture. Results are reported as Mean_{Std}.

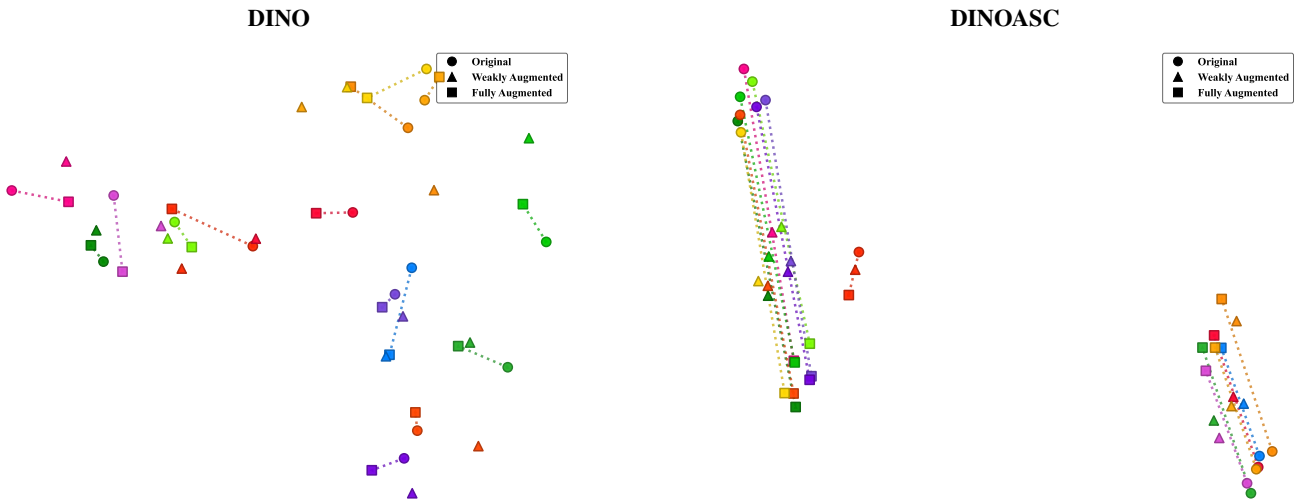

Figure 5. t-SNE projections of embeddings for contrast augmentation. Each plot visualizes the original embedding, its strongly augmented counterpart, and the linear interpolation connecting them. DINOASC produces weakly augmented embeddings that align closely with this interpolation path, indicating improved semantic consistency compared to vanilla DINO [3].

Method	AUC	ACC
MAE No Augmentation	$0.805_{.011}$	$0.822_{.029}$
MAE Search	$0.828_{.031}$	$0.831_{.047}$
DINOASC Search	$0.941_{.016}$	$0.919_{.026}$

Table 8. Comparison of MAE and DINOASC on CAMELYON16 using DSMIL as the MIL-pooling. Reported values are mean and standard deviation in the format Mean_{Std}.

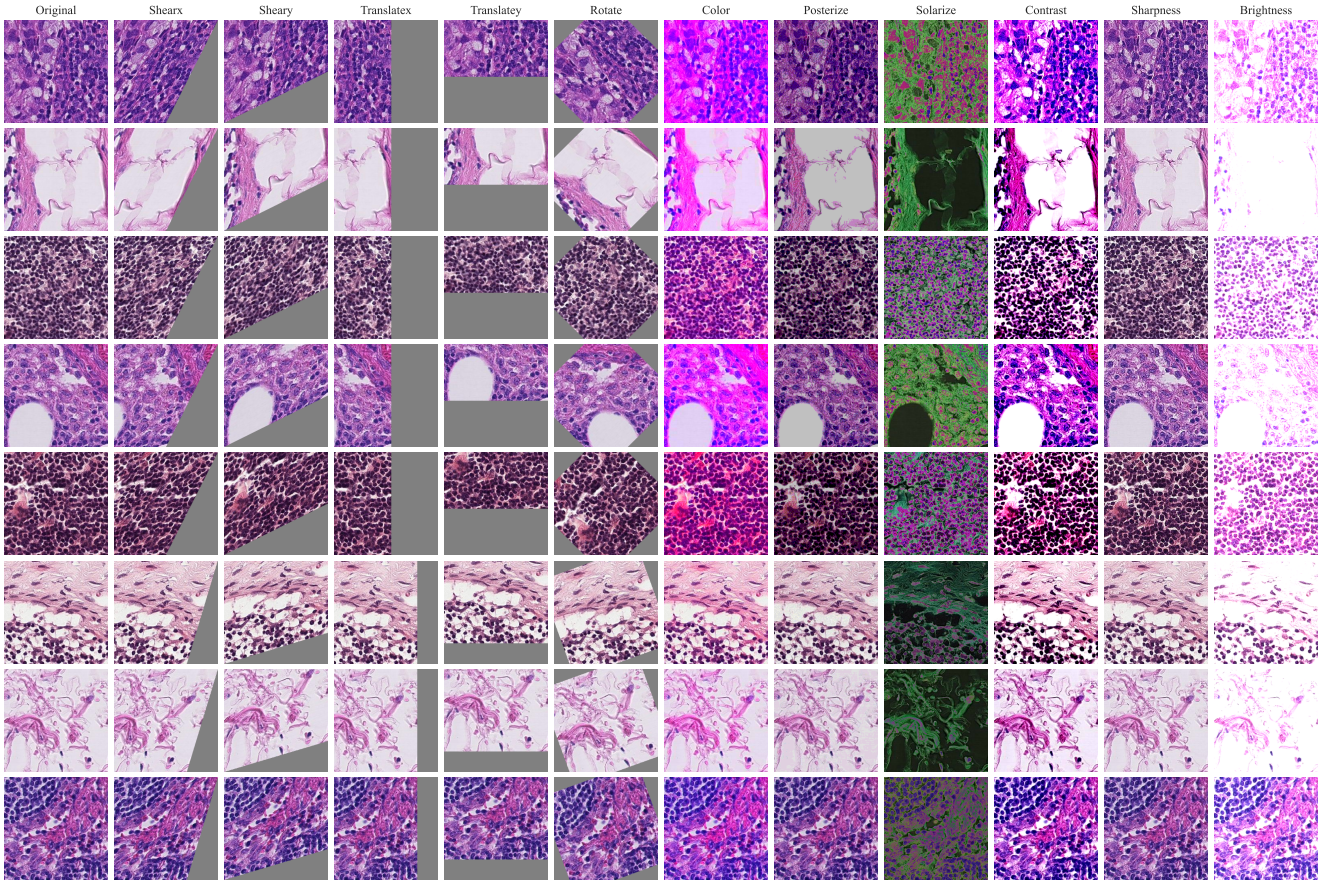

Figure 6. Visualization of high-intensity augmentations applied to eight random samples. Each row corresponds to an input image, and each column shows the result of a different augmentation.

Algorithm 2: Bayesian Search on Augmentation Upper Bounds

Input: Augmentation functions set $\mathcal{A}$, MIL model g, Discretized search space: $m_\tau^{\max}$ for each $\tau \in \mathcal{A}$, Total trials: N, Initial random trials: N_{init}, Acquisition function: Expected Improvement.

Output: Optimal set $\{m_\tau^{\max *}\}_{\tau \in \mathcal{A}}$.

Initialize surrogate model (Gaussian Process) with empty observations

Sample initial configurations $\{\{m_\tau^{\max}\}_{\tau \in \mathcal{A}}\}_{i=1}^{N_{\text{init}}}$ via Sobol

for *each initial configuration* **do**

    Evaluate $g(\{m_\tau^{\max}\}_{\tau \in \mathcal{A}})$ using TrivialInterpolate in MIL pipeline

    Update surrogate model with observation

end

for *trial* $i = N_{init} + 1$ *to* N **do**

    Select next configuration $\{m_\tau^{\max}\}_{\tau \in \mathcal{A}}$ by optimizing EI

    Evaluate $g(\{m_\tau^{\max}\}_{\tau \in \mathcal{A}})$ using TrivialInterpolate in MIL pipeline

    Update surrogate model with observation

end

return *Best found configuration* $\{m_\tau^{\max *}\}_{\tau \in \mathcal{A}}$