ADAPTMIL: DOMAIN ADAPTATION FOR MULTIPLE INSTANCE LEARNING IN DIGITAL PATHOLOGY

Anonymous authors
Paper under double-blind review

ABSTRACT

Pathology foundation models (PFMs) support a wide range of histopathology tasks on whole-slide images through multiple instance learning (MIL). Despite recent progress, these models remain sensitive to cross-hospital distribution shifts: performance often degrades when a task head (e.g., classification) is trained on data from one hospital and then used on data from another hospital. To address this challenge, we propose AdaptMIL, a training strategy for integrating domain adaptation into MIL training. AdaptMIL is motivated by our finding that directly applying domain adaptation objectives during MIL training can interfere with MIL aggregation learning, often resulting in performance degradation over standard training. AdaptMIL resolves this by decoupling representation-level domain alignment learning from MIL aggregation learning through a two-stage training procedure. We show that AdaptMIL improves cross-hospital performance and outperforms alternative adaptation methods across experiments spanning multiple WSI datasets, tasks, PFMs, MIL architectures, and domain adaptation objectives. AdaptMIL's code will be released upon publication.

1 INTRODUCTION

The emergence of modern pathology foundation models (PFMs) has driven substantial progress in digital pathology in recent years (Chen et al., 2024; Lu et al., 2024; Filiot et al., 2025). These models take digitized tissue images as input and produce general-purpose representations that can support a wide range of downstream histopathology tasks, including cancer detection, tumor subtyping, biomarker prediction, and survival analysis (Neidlinger et al., 2026).

Unlike conventional image encoders that process standard-sized images, PFMs are used to process extremely large images, commonly referred to as whole-slide images (WSIs), which can contain billions of pixels. This requires a specialized processing pipeline: a WSI is first divided into smaller patches, each patch is independently encoded by a PFM into a patch-level representation, and finally, the patch representations are aggregated into a single slide-level embedding using a weakly supervised multiple instance learning (MIL) module, most commonly trained from scratch per task (Wang et al., 2024; Shao et al., 2025).

Despite their progress, PFM-based MIL pipelines may still struggle in real-world deployment conditions. A major challenge is their limited robustness to variability across hospitals from which tissue samples were collected, arising from differences in tissue processing protocols (Kömen et al., 2026). As a result, the performance of task heads (e.g., classifiers) built on PFM representations degrades when the MIL module and task head are trained on data from one hospital and used on data from another (Fernandez-Romero et al., 2026; Umer et al., 2026). This poses a practical limitation, since labeled data from a specific *target hospital* is often unavailable due to costly and time-consuming expert annotation, whereas unlabeled data from that hospital may be readily available. Meanwhile, labeled data from other *source hospitals* may already be available, a setting known as *unsupervised domain adaptation* (UDA) (Wilson & Cook, 2020).

A direct approach is to fine-tune the pretrained PFM toward the target hospital domain. This can be done using domain adaptation objectives in the UDA setting, and requires labels for individual image patches from source-domain WSIs. Although this approach has been explored before (Vardi et al., 2026), its applicability is limited by the scarcity of patch-level annotations in pathology. By

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

contrast, slide-level annotations are more readily available and can be used for end-to-end PFM fine-tuning. However, this requires propagating gradients from the slide-level objective through the MIL module and back through the PFM across hundreds or thousands of patches per slide, substantially increasing the computational cost compared with training the MIL module alone (Campanella et al., 2025; Kumar & Vanderbilt, 2026).

In this work, we pursue a simpler alternative: we keep the pretrained PFM frozen and perform adaptation within the MIL training pipeline, using only slide-level supervision. We find that a standard approach to unsupervised domain adaptation, which combines a task loss on the labeled source domain and a domain alignment objective to align the source and target domains, can fail when applied to MIL pipelines. In particular, we find that domain alignment and learned MIL aggregation can interfere with one another: rather than learning more domain-invariant representations, the MIL module reduces the domain discrepancy by altering its aggregation pattern. This can result in performance below source-only training.

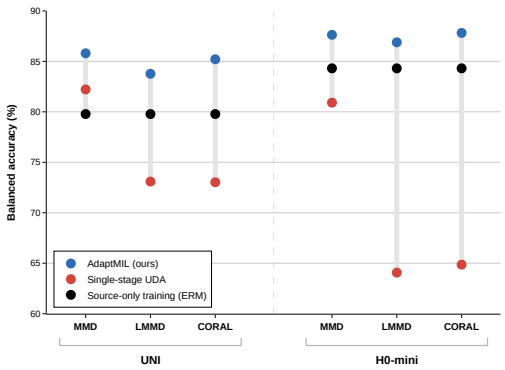


Figure 1: AdaptMIL improves cross-hospital performance across PFMs such as UNI and H0-mini and across domain adaptation objectives. It is motivated by the observation that directly applying standard UDA during MIL training often degrades performance.

To address this, we propose **AdaptMIL**, a two-stage MIL training strategy for domain adaptation that decouples domain alignment from aggregation learning. The first stage trains a patch-level projection layer to make representations more domain-aligned and uses fixed mean pooling to aggregate them into slide-level representations. Source–target domain alignment is performed on these slide-level representations together with a source task objective that preserves task-discriminative information. In the second stage, the MIL aggregation is trained using the source task objective on top of the adapted projection layer learned in the first stage. By performing domain alignment before learning the aggregation mechanism, AdaptMIL prevents the MIL module from altering its aggregation pattern to satisfy the alignment objective, resulting in gains over source-only training (Fig. 1).

Across experiments on multi-hospital datasets, we show that AdaptMIL outperforms source-only MIL training and alternative adaptation methods. We demonstrate these gains across multiple PFMs, slide-level tasks, domain alignment objectives, and MIL aggregation methods. Our code will be released upon publication.

Our contributions are:

- We explore UDA in slide-level pathology MIL pipelines and identify a failure mode for jointly learning domain alignment and MIL aggregation.
- We introduce AdaptMIL, a two-stage training strategy for MIL that decouples domain alignment from aggregation learning and can be applied across different MIL architectures.
- We evaluate AdaptMIL across multiple pathology tasks, WSI datasets, PFMs, MIL architectures, and domain alignment objectives, and demonstrate improved cross-hospital performance over source-only training and other adaptation approaches.

2 Related Work

2.1 Pathology Foundation Models and MIL

Pathology foundation models (PFMs) encode hematoxylin and eosin (H&E)-stained histopathology images into general-purpose embeddings that support a wide range of downstream tasks (Chen et al., 2024; Lu et al., 2024; Filiot et al., 2025; Vorontsov et al., 2024). These representations are learned through large-scale pretraining, using objectives such as self-supervision (Chen et al., 2024;

Vorontsov et al., 2024) and vision-language contrastive learning (Lu et al., 2024). In pathology practice, inference is performed on whole-slide images (WSIs), which often reach gigapixel scale (e.g., $100{,}000 \times 100{,}000$ pixels). The predominant pipeline therefore segments the WSI, divides it into patches, encodes the patches independently using a PFM, and then aggregates the patch embeddings into a slide-level embedding using a multiple instance learning (MIL) module (Wang et al., 2024).

MIL addresses learning from bags of instances for which supervision is available only at the bag level (Dietterich et al., 1997). In digital pathology, a WSI is treated as a bag of patches (Campanella et al., 2019): while a WSI-level label may indicate, for example, the presence of cancer, labels for the individual patches are generally unavailable. MIL methods therefore aggregate patch embeddings into a single slide-level representation from which the downstream task can be predicted.

A wide range of MIL methods have been proposed, varying along several dimensions, such as their aggregation mechanisms, architectures, and whether the aggregation is learnable or parameter-free. An influential architecture is ABMIL (Ilse et al., 2018), which learns a network that assigns attention weights to instances and aggregates them through a weighted average. Subsequent methods introduce more sophisticated aggregation mechanisms. For example, DSMIL (Li et al., 2021) identifies a critical instance and uses it to guide attention over the remaining instances, while TransMIL (Shao et al., 2021) employs transformer-based attention to model dependencies between patches. A common practice is to train MIL models from scratch for each downstream task, while more recent work has explored pretraining slide-level MIL models (Ding et al., 2025; Shao et al., 2025).

2.2 Cross-hospital robustness

Despite impressive performance, slide-level PFM pipelines may face limitations in real-world settings. A major challenge emerges in multi-hospital settings due to variation in H&E appearance, driven by differences in tissue processing, staining, and digitization protocols (Howard et al., 2021; Thiringer et al., 2026).

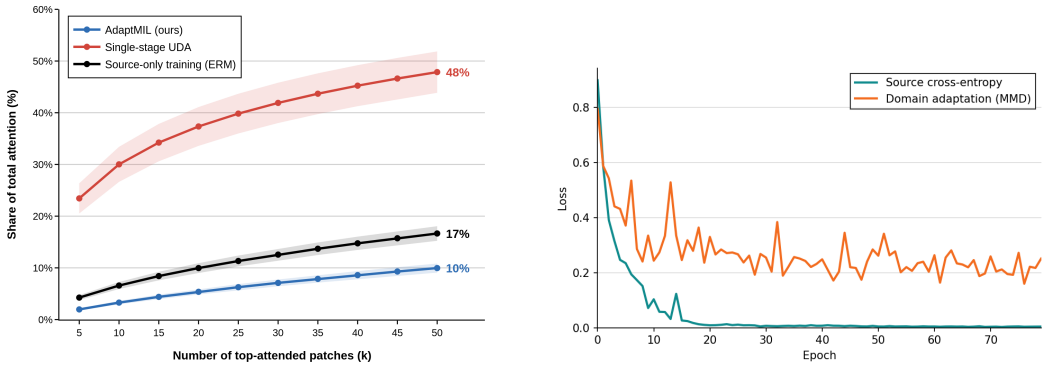
Prior work has examined cross-hospital robustness at the patch level, showing that PFM embeddings encode hospital-specific signatures and exhibit related distribution shifts (Kömen et al., 2026). Proposed approaches to mitigate these shifts include test-time stain normalization of target patches (Macenko et al., 2009; Reinhard et al., 2002), training patch-level models with domain adaptation objectives (Vray et al., 2024), fine-tuning PFMs for domain alignment (Vardi et al., 2026), and learning transformations on top of frozen PFM features (Kömen et al., 2026).

Cross-hospital performance degradation has also been observed in slide-level tasks (Fernandez-Romero et al., 2026; Umer et al., 2026). Several patch-level solutions can be transferred to slide-level prediction: stain normalization can be applied across all patches, while adapted PFMs can be used within MIL pipelines. However, direct slide-level adaptation remains comparatively underexplored. HASD (Liu et al., 2025) takes this direction and proposes a pathology-specific hierarchical adaptation framework with several specialized objectives. We also perform adaptation at the slide-aggregation level, but rather than introducing a specialized adaptation objective, we propose a training strategy that enables established domain adaptation objectives to be integrated into MIL.

2.3 Domain adaptation

Domain adaptation (Wang & Deng, 2018) aims to transfer a model trained on data from a source domain to a different target domain. Unsupervised domain adaptation (UDA) (Wilson & Cook, 2020) is a common setting, where labeled source data is available, while only unlabeled data is available in the target domain.

A central approach to UDA is domain-invariant feature learning, which aims to reduce differences between source and target representation distributions. Invariant representations can be learned through discrepancy minimization methods, which minimize a differentiable distance metric between the two distributions. Prominent examples include Maximum Mean Discrepancy (MMD) (Long et al., 2015), local MMD (LMMD) (Zhu et al., 2020), and CORAL (Sun & Saenko, 2016). Other domain-invariant methods use different mechanisms, such as domain-adversarial neural networks (DANN) (Ganin et al., 2016), which learns domain-invariant features by making representations from the source and target domains difficult to distinguish.



(a) (b)

Figure 2: Analysis of the interaction between domain adaptation and ABMIL aggregation. (a) Standard UDA (single-stage) concentrates attention on a smaller subset of patches than source-only training. AdaptMIL reduces and even reverses this effect, resulting in a more distributed attention pattern. (b) In a representative single-stage UDA run, both source cross-entropy and MMD losses decrease despite degraded performance (the MMD plateau is expected). Results are from NSCLC subtyping with UNI; attention in (a) is averaged over held-out target slides.

Although widely used across computer vision and machine learning, the interaction between domain adaptation and MIL remains less explored in slide-level pathology. In this work, we investigate this interaction and develop a training strategy for domain adaptation within MIL pipelines.

3 METHOD

3.1 PROBLEM SETUP AND DOMAIN ADAPTATION FOR MIL

In UDA, we are given n_s labeled samples from a source domain $\{(x_i^s, y_i^s)\}_{i=1}^{n_s}$, together with n_t unlabeled samples from a target domain, $\{x_j^t\}_{j=1}^{n_t}$. In the cross-hospital setting, each sample x is a WSI, the labels y_i correspond to biological classes (e.g., cancer status), and each domain corresponds to a different hospital. The goal is to learn a MIL module and a task head (e.g., a classifier) that generalize to the target hospital without having access to its labels.

Domain-invariant UDA methods jointly optimize a supervised task loss on labeled source samples and a domain adaptation objective that encourages alignment between source and target representations (Wilson & Cook, 2020). Let $\boldsymbol{z}_i^s = f_\theta(x_i^s)$ and $\boldsymbol{z}_j^t = f_\theta(x_j^t)$ denote the source and target representations, respectively, and let g_ϕ be the task head. The UDA training objective is

$$\frac{1}{n_s}\sum_{i=1}^{n_s}\mathcal{L}_{\text{task}}\left(g_\phi(\boldsymbol{z}_i^s), y_i^s\right) + \lambda\,\mathcal{L}_{\text{DA}}\left(\{\boldsymbol{z}_i^s\}_{i=1}^{n_s}, \{\boldsymbol{z}_j^t\}_{j=1}^{n_t}\right), \tag{1}$$

where $\mathcal{L}_{\text{task}}$ is the supervised task loss, $\mathcal{L}_{\text{DA}}$ is the domain adaptation loss, and λ controls the tradeoff between them. The task head and loss can correspond to any supervised prediction task, e.g., for classification, g_ϕ is a classifier and $\mathcal{L}_{\text{task}}$ is the cross-entropy loss.

This setting can be adapted to slide-level MIL, where each sample is a bag of patches rather than a single image. Let $x_i^s = \{p_{i,1}^s, \ldots, p_{i,m_i^s}^s\}$ denote a source WSI decomposed into m_i^s patches. Each patch is encoded by a PFM encoder f_{PFM}, yielding the bag of patch representations $z_i^s = \{f_{\text{PFM}}(p_{i,1}^s), \ldots, f_{\text{PFM}}(p_{i,m_i^s}^s)\}$. The target WSI x_j^t and bag z_j^t are defined analogously. A learnable MIL aggregation module h_{MIL} then maps z_i^s into a slide representation, on which the task head operates. Applying the UDA formulation gives

$$\frac{1}{n_s}\sum_{i=1}^{n_s}\mathcal{L}_{\text{task}}\left(g_\phi(h_{\text{MIL}}(z_i^s)), y_i^s\right) + \lambda\,\mathcal{L}_{\text{DA}}\left(\{h_{\text{MIL}}(z_i^s)\}_{i=1}^{n_s}, \{h_{\text{MIL}}(z_j^t)\}_{j=1}^{n_t}\right), \tag{2}$$

where domain adaptation is performed on the slide representations produced by the MIL module.

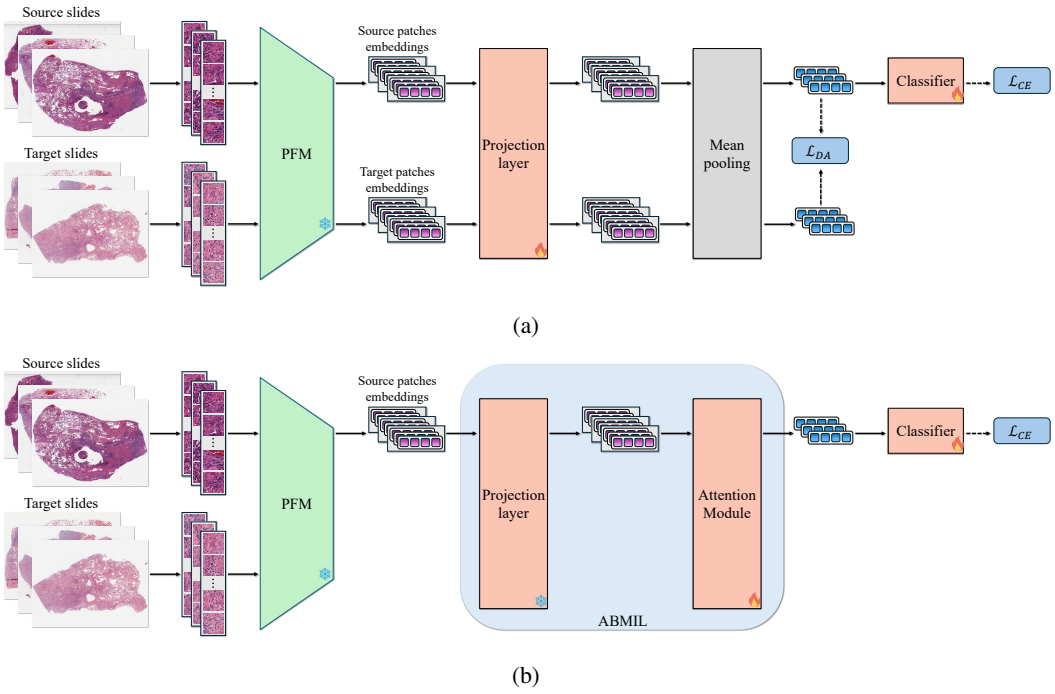


Figure 3: AdaptMIL two-stage training with ABMIL for classification. (a) In Stage 1, a projection layer is trained to map patch embeddings into a more domain-aligned space. The projected embeddings are aggregated using mean pooling, and training uses both a cross-entropy (CE) loss on labeled source slides and a domain adaptation (DA) loss between source and target slide embeddings. (b) In Stage 2, mean pooling is replaced by ABMIL's learnable attention module. The projection layer is frozen, and the attention module and classifier are trained using only labeled source slides.

As in standard domain-invariant UDA, the feature extractor (the PFM) can remain frozen while a learnable projection transforms its representations into a more domain-aligned space. In many MIL architectures such a projection is already included prior to aggregation (Shao et al., 2021; Ilse et al., 2018; Lu et al., 2021), eliminating the need for an additional projection layer. The MIL module and task head are thus trained jointly, with the only difference from standard source-only MIL training (using only labeled source data and the task loss) being the addition of the domain adaptation loss.

3.2 Failure mode of domain adaptation and MIL training

We find that the application of UDA to MIL training (Eq. (2)) is mostly ineffective, performing worse than source-only training in most settings and providing only limited gains otherwise (see Fig. 1). Analyzing this reveals a surprising effect: joint MIL and domain adaptation training substantially changes the learned attention pattern compared to standard MIL training, concentrating attention on a smaller subset of patches. We demonstrate this effect in Fig. 2a: Under UDA training, the top-k attended patches in ABMIL account for a substantially larger fraction of the total attention (e.g., $2.8\times$ for $k = 50$). This shows that the adaptation objective affects not only the learned projection layer, as might be expected, but also how patch information is aggregated by the MIL module.

Examining the training dynamics shows that both the task and domain adaptation losses decrease as expected (Fig. 2b), indicating that the optimization objectives are satisfied. Together with the degradation in performance, this suggests that the MIL model takes advantage of a shortcut to reduce the domain gap. That is, rather than learning genuinely more domain-robust representations, it can satisfy both objectives by changing how patch information is aggregated.

We hypothesize that this shortcut is facilitated by the morphological redundancy across patches in WSIs (Song et al., 2024), which may provide the MIL model with multiple ways to satisfy the task

objective. Adding the domain adaptation constraint may therefore push the model toward a different aggregation solution that, in this case, generalizes worse.

We emphasize that more concentrated attention is not inherently undesirable, as its effect depends on the regions receiving high attention and the quality of the representations. Our analysis therefore does not imply that concentrated attention alone causes performance degradation. Rather, it shows that domain adaptation can substantially alter the learned aggregation pattern, even though the intended effect is to only make representations more domain-invariant, and that these changes are associated with degraded performance.

3.3 ADAPTMIL

Our analysis shows that applying domain adaptation objectives interferes with MIL aggregation learning. This motivated us to try and decouple the two, namely separate domain-invariant representation learning from learning the aggregation mechanism. We propose **AdaptMIL**, a two-stage training strategy that first learns domain-aligned representations and then learns the MIL aggregation over the adapted representations. AdaptMIL is formulated here using ABMIL (Ilse et al., 2018) but can be applied to different MIL architectures. See Fig. 3 for an illustration of AdaptMIL.

ABMIL first transforms each patch representation through a learned projection layer P, and then aggregates the projected representations with an attention module Attn. Let $z_{i,k} = f_{\mathrm{PFM}}(p_{i,k})$ denote the representation of patch k from slide i. ABMIL is given by $h_{\mathrm{MIL}}(z_i) = \mathrm{Attn}(P(z_i))$, where P is applied independently to each patch representation, i.e., $P(z_i) = \{P(z_{i,1}), \ldots, P(z_{i,m_i})\}$.

In the first stage, AdaptMIL learns a projection layer P that maps patch representations into a more domain-invariant space, and does not use the attention module. To obtain slide-level representations for the task and domain alignment objectives without the attention module, the projected representations are aggregated using parameter-free mean pooling. The resulting Stage 1 objective is:

$$\begin{aligned}\mathcal{L}_{\mathrm{stage1}} = \frac{1}{n_s}\sum_{i=1}^{n_s} \mathcal{L}_{\mathrm{task}}(g_\phi(f_{\mathrm{mean}}(P(z_i^s))),\, y_i^s) \\ + \lambda\, \mathcal{L}_{\mathrm{DA}}\left(\{f_{\mathrm{mean}}(P(z_i^s))\}_{i=1}^{n_s}, \{f_{\mathrm{mean}}(P(z_j^t))\}_{j=1}^{n_t}\right),\end{aligned} \tag{3}$$

where $f_{\mathrm{mean}}(P(z_i)) = \frac{1}{m_i}\sum_{k=1}^{m_i} P(z_{i,k})$. Mean pooling is a simple but competitive aggregation strategy in computational pathology (Shao et al., 2025). Importantly, because it is parameter-free, it cannot be altered by the domain adaptation objective.

In the second stage, AdaptMIL learns the MIL aggregation over the adapted representations. It keeps the projection layer P learned in Stage 1 frozen and replaces mean pooling with ABMIL's attention module, recovering the full ABMIL architecture. We train this stage using only source samples and the task loss. This allows the learnable attention module to focus on task-discriminative information without being influenced by the domain adaptation objective. The resulting Stage 2 objective is:

$$\mathcal{L}_{\mathrm{stage2}} = \frac{1}{n_s}\sum_{i=1}^{n_s} \mathcal{L}_{\mathrm{task}}(g_\phi(h_{\mathrm{MIL}}(z_i^s)),\, y_i^s)\,. \tag{4}$$

As shown in Fig. 1, the decoupled two-stage training of AdaptMIL improves over both source-only training and standard UDA. Fig. 2a shows that AdaptMIL does not exhibit the increased attention concentration observed with standard UDA and instead produces more distributed attention.

4 EXPERIMENTS

4.1 TASKS AND DATASETS

We evaluate AdaptMIL on four different tasks. The first is non-small cell lung cancer (NSCLC) subtyping between lung adenocarcinoma and squamous cell carcinoma, using 489 WSIs from TCGA (Komura et al., 2022) across six hospitals, and 150 WSIs from an in-house hospital dataset. The second task is renal cell carcinoma (RCC) subtyping between clear cell carcinoma and papillary cell carcinoma, using 307 WSIs from TCGA across three hospitals. The third task is cancer detection, using 404 WSIs from CAMELYON17 (Bandi et al., 2018) across five hospitals. The fourth task is survival prediction, presented in Sec. D.3 of the supplementary material (Supp.).

Table 1: Cross-hospital NSCLC TCGA subtyping results for AdaptMIL with ABMIL. The metric is balanced accuracy (%). Values in parentheses denote the difference from ERM.

Method	**UNI**	**CONCH-1.5**	**H0-mini**
In-domain training	86.98 (+7.21)	88.88 (+2.93)	91.38 (+7.07)
Source-only (ERM)	79.78	85.95	84.31
Macenko	77.72 (-2.06)	85.32 (-0.63)	82.67 (-1.64)
Patch-level fine-tuning	84.25 (+4.47)	87.11 (+1.16)	–
Single-stage UDA (MMD)	82.22 (+2.45)	86.75 (+0.80)	80.91 (-3.40)
AdaptMIL-MMD (ours)	**85.80** (+6.02)	**88.37** (+2.42)	87.63 (+3.32)
Single-stage UDA (LMMD)	73.09 (-6.69)	76.25 (-9.69)	64.07 (-20.24)
AdaptMIL-LMMD (ours)	83.77 (+3.99)	88.17 (+2.23)	86.89 (+2.58)
Single-stage UDA (CORAL)	73.02 (-6.76)	82.15 (-3.80)	64.86 (-19.45)
AdaptMIL-CORAL (ours)	85.21 (+5.43)	88.13 (+2.19)	**87.82** (+3.51)
Single-stage UDA (DANN)	79.37 (-0.41)	85.17 (-0.78)	84.49 (+0.18)
AdaptMIL-DANN (ours)	81.38 (+1.60)	86.54 (+0.60)	86.22 (+1.90)

4.2 Evaluation

For each subtyping or detection dataset, we evaluate all source-target hospital pairs and report the average metric across them. For example, the TCGA NSCLC dataset contains six hospitals, resulting in 30 source-target pairs. For the in-house lung dataset, we evaluate both directions with each of the six TCGA hospitals, resulting in 12 pairs. Each source-target pair is evaluated using 3-fold cross-validation, with evaluation performed on held-out target WSIs that are not used during training. Our main metric for subtyping and detection is balanced accuracy.

4.3 Models and training

We evaluate AdaptMIL across different PFMs, domain adaptation objectives, and MIL architectures. For patch-level feature extraction, we use three PFMs: UNI (Chen et al., 2024), CONCH-1.5 (Lu et al., 2024), and H0-mini (Filiot et al., 2025). Tissue segmentation, patch extraction, and feature extraction are performed using the TRIDENT package (Zhang et al., 2025).

We evaluate four domain adaptation objectives: three discrepancy-based losses, MMD (Long et al., 2015), LMMD (Zhu et al., 2020), and CORAL (Sun & Saenko, 2016), and the adversarial DANN objective (Ganin et al., 2016). We further apply AdaptMIL to three MIL architectures: ABMIL (Ilse et al., 2018), DSMIL (Li et al., 2021), and TITAN (Ding et al., 2025). The implementation for ABMIL is described in Sec. 3.3. The implementations for DSMIL and TITAN follow the same principle with architecture-specific adjustments. All AdaptMIL models are trained for 80 epochs, with 40 epochs for each stage. Further implementation details are provided in Sec. A of the Supp.

4.4 Method comparisons

We compare AdaptMIL against several baselines. First, we compare against source-only training with standard MIL. This baseline, commonly referred to as empirical risk minimization (ERM) (Vapnik, 1999), represents the out-of-distribution performance that domain adaptation aims to improve. We also compare to standard UDA applied to MIL (defined in Eq. (2)), using each of the four domain adaptation objectives. We refer to this baseline as single-stage UDA.

We additionally compare against Macenko stain normalization (Macenko et al., 2009). In this setting, the MIL model is trained only on source data, as in ERM, but evaluation is performed on target images whose appearance is normalized to a source reference. Finally, we compare against a domain adaptation approach that fine-tunes the PFM using patch-level supervision (Vardi et al., 2026). We note that, to the best of our knowledge, other pathology-specific domain adaptation methods do not have publicly available implementations, and therefore we could not compare against them.

Table 2: Cross-hospital classification results for AdaptMIL across different tasks. The metric is balanced accuracy (%). All results use AdaptMIL based on ABMIL with MMD. IH-NS denotes the in-house NSCLC task, and Det. denotes cancer detection.

Method	UNI			CONCH-1.5			H0-mini		
	IH-NS	Det.	RCC	IH-NS	Det.	RCC	IH-NS	Det.	RCC
In-domain training	89.49 (+8.87)	82.25 (+9.43)	96.45 (+4.67)	89.44 (+5.70)	98.41 (+1.99)	93.79 (+0.79)	93.02 (+12.03)	84.93 (+3.75)	94.94 (+2.10)
Source-only (ERM)	80.62	72.82	91.78	83.74	**96.43**	93.01	81.00	81.18	92.83
Single-stage UDA	**85.20** (+4.59)	72.31 (-0.51)	90.75 (-1.03)	85.58 (+1.84)	86.11 (-10.31)	92.53 (-0.47)	76.44 (-4.56)	70.28 (-10.90)	88.93 (-3.90)
AdaptMIL (ours)	82.30 (+1.68)	**78.87** (+6.05)	**95.09** (+3.31)	**86.72** (+2.98)	95.59 (-0.84)	**94.10** (+1.09)	**86.92** (+5.92)	**89.30** (+8.12)	**93.32** (+0.48)

Table 3: Cross-hospital NSCLC TCGA subtyping results across different MIL architectures. The metric is balanced accuracy (%).

Method	ABMIL (UNI)	DSMIL (UNI)	TITAN
In-domain training	86.98 (+7.21)	83.12 (+8.78)	92.32 (+1.65)
Source-only (ERM)	79.78	74.34	90.68
Single-stage UDA	82.22 (+2.45)	81.29 (+6.95)	89.50 (-1.18)
AdaptMIL (ours)	**85.80** (+6.02)	**82.10** (+7.76)	**90.88** (+0.20)

Table 4: Ablation studies for AdaptMIL.

Method	Balanced Acc.
Source-only (ERM)	79.78
Single-stage + mean-pool adap. head	84.87 (+5.09)
AdaptMIL w/ Stage 1 only	82.17 (+2.39)
AdaptMIL w/ adaptation in Stage 2	83.23 (+3.45)
AdaptMIL w/o projection freeze	85.56 (+5.79)
AdaptMIL w/o adaptation loss	81.38 (+1.60)
AdaptMIL (full)	**85.80** (+6.02)

4.5 RESULTS

Tab. 1 presents the evaluation of AdaptMIL with ABMIL across different PFMs and domain adaptation objectives on the NSCLC subtyping task. AdaptMIL improves over ERM across all PFMs and objectives, with gains up to $+6.02\%$ in balanced accuracy. MMD is the best-performing objective for two of the three PFMs, while LMMD and CORAL also yield notable gains, and DANN provides smaller improvements. Similar trends are observed for ROC-AUC and F1 (see Tab. 6 in Supp.).

We additionally report in-domain performance of an ABMIL model that is trained and evaluated on samples from the same hospital. The performance drop from in-domain to source-only ERM highlights the degradation caused by the cross-hospital distribution shift. AdaptMIL substantially reduces this gap. For example, with UNI and CONCH-1.5, AdaptMIL recovers more than 80% of the gap.

AdaptMIL outperforms the other adaptation methods. As observed in Sec. 3.2, directly applying standard single-stage UDA during MIL training often fails to improve over ERM and can substantially degrade performance. For example, LMMD with H0-mini reduces balanced accuracy by 20.24%. Macenko normalization also fails to improve performance, consistent with previous reports (Huang et al., 2025; Lin et al., 2025). Patch-level fine-tuning of PFMs is the baseline that provides the largest gains over ERM, but it remains inferior to AdaptMIL. Although effective, patch-level fine-tuning requires both patch supervision and re-extraction of patch features for every source–target adaptation setting. In contrast, AdaptMIL requires only slide-level supervision and operates on pre-extracted PFM features, allowing the same features to be reused across adaptation settings.

We evaluate AdaptMIL across additional datasets and classification tasks in Tab. 2. AdaptMIL improves over source-only ERM in eight of the nine settings, with gains of up to $+8.12\%$. The only exception is cancer detection with CONCH-1.5, where ERM already achieves very strong performance. In contrast, single-stage UDA degrades performance in seven of the nine settings.

Finally, we evaluate AdaptMIL with MIL architectures beyond ABMIL in Tab. 3. AdaptMIL improves performance with DSMIL, although standard single-stage UDA is also effective. We further obtain a gain when adapting the pretrained TITAN slide encoder. These results indicate that AdaptMIL is not specific to ABMIL and can be applied to other MIL architectures.

4.6 Ablations

We perform ablation studies to examine the impact of AdaptMIL's components, with results for AdaptMIL-MMD with ABMIL and UNI in Tab. 4. We first consider an alternative way to decouple representation alignment from aggregation learning. Instead of two-stage training, we train ABMIL throughout with the task objective, while applying the adaptation objective to a separate mean-pooling head placed after the projection layer. This prevents adaptation gradients from reaching the attention module, while still adapting the projection layer. This alternative method performs well (row 2 in Tab. 4), but remains inferior to AdaptMIL, suggesting an additional benefit from explicitly separating alignment and aggregation learning into distinct training stages.

We next examine whether the second stage of AdaptMIL is necessary. Specifically, we use only Stage 1, where the learned projection is followed by mean pooling and trained with both the task and adaptation objectives. This performs worse than full AdaptMIL (row 3), confirming the benefit of using ABMIL, consistent with prior work favoring ABMIL over mean pooling (Shao et al., 2025).

We additionally evaluate two design choices in Stage 2. Keeping the adaptation loss active in Stage 2 degrades performance (row 4), supporting our hypothesis that adaptation gradients interfere with aggregation learning. Not freezing the projection layer in Stage 2 slightly reduces performance (row 5), possibly because the projection becomes more source-oriented.

Finally, we test whether the two-stage training schedule itself contributes to the improvement by removing the adaptation objective (row 6). This variant still improves over ERM, suggesting that separating representation and aggregation learning provides some benefit on its own. However, it remains substantially below AdaptMIL, demonstrating the importance of the alignment in Stage 1.

5 Conclusion

In this paper, we introduce AdaptMIL, a training strategy for performing domain adaptation within MIL architectures. Motivated by the observation that applying standard UDA to MIL can interfere with aggregation learning and degrade performance, AdaptMIL decouples domain alignment and aggregation learning into two training stages. Evaluated in cross-hospital adaptation settings with different tasks, PFMs, MIL architectures, and domain adaptation objectives, AdaptMIL consistently improves over source-only training and outperforms alternative adaptation methods.

Our work has several limitations. We primarily evaluate MIL models trained from scratch, which is the most common setting and offers flexibility in the choice of the PFM (Shao et al., 2025). Another approach is to use pretrained MIL models, which avoid training from scratch but are tied to specific PFM features. Our TITAN experiments show that AdaptMIL can be applied in this setting, although broader evaluation remains for future work. A further limitation is that AdaptMIL depends on the quality of the PFM representations and may not be able to compensate for strong representation biases. Combining AdaptMIL with end-to-end PFM fine-tuning (Kumar & Vanderbilt, 2026) is therefore an interesting direction for future work. Finally, our survival prediction results (in Supp.) show that AdaptMIL improves in this setting, while standard UDA can sometimes be effective, motivating further study of when decoupling domain alignment from MIL aggregation is necessary.

We focus on cross-hospital domain adaptation in digital pathology, but AdaptMIL may also apply to other MIL settings affected by a domain shift. Examples include shifts between frozen and formalin-fixed paraffin-embedded tissues in pathology (Ozyoruk et al., 2022), as well as MIL applications in other domains such as medical imaging (Cheplygina et al., 2019) and remote sensing (Paheding et al., 2024).

AI use statement

We used generative AI tools during the research process and preparation of this paper:

1. Implementation: We used generative AI to assist with writing and modifying code, typically by providing specific implementation tasks (e.g., implementing a particular feature). All code contributing to the method and reported results was manually reviewed by the authors, including through debugging and verification of correctness.
2. Research ideation and methodology: We used generative AI to discuss research ideas, propose and refine hypotheses, and consider methodological and experimental choices.
3. Literature review: We used dedicated AI literature review tools to help identify relevant papers. We also manually searched for relevant literature, including through citation chains and independent searches. We did not rely solely on AI-generated summaries. Each paper cited in this work was manually checked by the authors for its relevance. We did not use AI tools to generate bibliography entries.
4. Analysis and interpretation: We used generative AI to assist with analyzing experimental results and discussing possible interpretations. All reported results, numerical values, and claims were manually checked by the authors.
5. Writing and presentation: We used generative AI to help polish and restructure text, assist with LaTeX, and generate code used to create some figures. Generative AI was not used to write complete sections of the paper. All AI-assisted text and figure-generation code were reviewed and revised by the authors.

We take responsibility for the final content of this work, including its text, claims, implementation, analyses, and figures.

Reproducibility statement

We will release our code upon publication. We have also made an effort to provide the information needed to reproduce our method and experiments in the paper. Sec. A in the supplementary material provides implementation details, including the public libraries used and hyperparameter choices. We also provide detailed information about the datasets in use in Sec. B of the supplementary material. For the four public datasets used in this work, we will release the exact lists of WSIs included in our experiments together with our code. The in-house NSCLC dataset is not planned for release.

References

Peter Bandi, Oscar Geessink, Quirine Manson, Marcory Van Dijk, Maschenka Balkenhol, Meyke Hermsen, Babak Ehteshami Bejnordi, Byungjae Lee, Kyunghyun Paeng, Aoxiao Zhong, et al. From detection of individual metastases to classification of lymph node status at the patient level: the CAMELYON17 challenge. *IEEE transactions on medical imaging*, 38(2):550–560, 2018.

Carlo Alberto Barbano and André Pedersen. Eidoslab/torchstain: v1.2.0-stable, August 2022.

Gabriele Campanella, Matthew G Hanna, Luke Geneslaw, Allen Miraflor, Vitor Werneck Krauss Silva, Klaus J Busam, Edi Brogi, Victor E Reuter, David S Klimstra, and Thomas J Fuchs. Clinical-grade computational pathology using weakly supervised deep learning on whole slide images. *Nature medicine*, 25(8):1301–1309, 2019.

Gabriele Campanella, Neeraj Kumar, Swaraj Nanda, Siddharth Singi, Eugene Fluder, Ricky Kwan, Silke Muehlstedt, Nicole Pfarr, Peter J Schüffler, Ida Häggström, et al. Real-world deployment of a fine-tuned pathology foundation model for lung cancer biomarker detection. *Nature Medicine*, 31(9):3002–3010, 2025.

Richard J Chen, Tong Ding, Ming Y Lu, Drew FK Williamson, Guillaume Jaume, Andrew H Song, Bowen Chen, Andrew Zhang, Daniel Shao, Muhammad Shaban, et al. Towards a general-purpose foundation model for computational pathology. *Nature medicine*, 30(3):850–862, 2024.

Veronika Cheplygina, Marleen De Bruijne, and Josien PW Pluim. Not-so-supervised: A survey of semi-supervised, multi-instance, and transfer learning in medical image analysis. *Medical image analysis*, 54:280–296, 2019.

Thomas G. Dietterich, Richard H. Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1):31–71, 1997. ISSN 0004-3702.

Tong Ding, Sophia J Wagner, Andrew H Song, Richard J Chen, Ming Y Lu, Andrew Zhang, Anurag J Vaidya, Guillaume Jaume, Muhammad Shaban, Ahrong Kim, et al. A multimodal whole-slide foundation model for pathology. *Nature medicine*, 31(11):3749–3761, 2025.

Jesus Fernandez-Romero, Pablo Ramos-Berciano, Manuel Perez-Perez, David Benavides, Antonio Robles-Frias, Jorge Garcia-Gutierrez, and Laura Macias-Garcia. Domain generalisation challenges in breast cancer molecular classification using foundation models: a cross-cohort exploratory study. *Medical & Biological Engineering & Computing*, 64:2321–2331, 2026.

Alexandre Filiot, Nicolas Dop, Oussama Tchita, Auriane Riou, Rémy Dubois, Thomas Peeters, Daria Valter, Marin Scalbert, Charlie Saillard, Geneviève Robin, et al. Distilling foundation models for robust and efficient models in digital pathology. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 162–172. Springer, 2025.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35, 2016.

Frederick M Howard, James Dolezal, Sara Kochanny, Jefree Schulte, Heather Chen, Lara Heij, Dezheng Huo, Rita Nanda, Olufunmilayo I Olopade, Jakob N Kather, et al. The impact of site-specific digital histology signatures on deep learning model accuracy and bias. *Nature communications*, 12(1):4423, 2021.

Yanyan Huang, Weiqin Zhao, Zhengyu Zhang, Yihang Chen, Yu Fu, Feng Wu, Yuming Jiang, Li Liang, Shujun Wang, and Lequan Yu. Knowledge-guided adaptation of pathology foundation models effectively improves cross-domain generalization and demographic fairness. *Nature communications*, 16(1):11485, 2025.

Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2127–2136. PMLR, 10–15 Jul 2018.

Junguang Jiang, Baixu Chen, Bo Fu, and Mingsheng Long. Transfer learning library. `https://github.com/thuml/Transfer-Learning-Library`, 2020. GitHub repository.

Jonah Kömen, Edwin D de Jong, Julius Hense, Hannah Marienwald, Jonas Dippel, Philip Naumann, Eric Marcus, Lukas Ruff, Maximilian Alber, Jonas Teuwen, et al. Towards robust foundation models for digital pathology. *Nature Communications*, 17(1):5218, 2026.

Daisuke Komura, Akihiro Kawabe, Keisuke Fukuta, Kyohei Sano, Toshikazu Umezaki, Hirotomo Koda, Ryohei Suzuki, Ken Tominaga, Mieko Ochi, Hiroki Konishi, Fumiya Masakado, Noriyuki Saito, Yasuyoshi Sato, Takumi Onoyama, Shu Nishida, Genta Furuya, Hiroto Katoh, Hiroharu Yamashita, Kazuhiro Kakimi, Yasuyuki Seto, Tetsuo Ushiku, Masashi Fukayama, and Shumpei Ishikawa. Universal encoding of pan-cancer histology by deep texture representations. *Cell Reports*, 38(9):110424, 2022.

Neeraj Kumar and Chad Vanderbilt. Single GPU task adaptation of pathology foundation models for whole slide image analysis. *Advances in Neural Information Processing Systems*, 38:11323–11350, 2026.

Bin Li, Yin Li, and Kevin W Eliceiri. Dual-stream multiple instance learning network for whole slide image classification with self-supervised contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14318–14328, 2021.

W Lin, S Liu, R Zhu, and L Wang. Unveiling institution-specific bias in pathology foundation models: Detriments, causes, and potential solutions, 2025.

Jingsong Liu, Han Li, Chen Yang, Michael Deutges, Ario Sadafi, Xin You, Katharina Breininger, Nassir Navab, and Peter J Schüffler. HASD: Hierarchical adaptation for pathology slide-level domain-shift. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 332–342. Springer, 2025.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pp. 97–105. PMLR, 2015.

Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2017.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2019.

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.

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.

Marc Macenko, Marc Niethammer, James S Marron, David Borland, John T Woosley, Xiaojun Guan, Charles Schmitt, and Nancy E Thomas. A method for normalizing histology slides for quantitative analysis. In *2009 IEEE international symposium on biomedical imaging: from nano to macro*, pp. 1107–1110. IEEE, 2009.

Peter Neidlinger, Omar SM El Nahhas, Hannah Sophie Muti, Tim Lenz, Michael Hoffmeister, Hermann Brenner, Marko Van Treeck, Rupert Langer, Bastian Dislich, Hans Michael Behrens, et al. Benchmarking foundation models as feature extractors for weakly supervised computational pathology. *Nature biomedical engineering*, pp. 1113–1123, 2026.

Kutsev Bengisu Ozyoruk, Sermet Can, Berkan Darbaz, Kayhan Başak, Derya Demir, Guliz Irem Gokceler, Gurdeniz Serin, Uguray Payam Hacisalihoglu, Emirhan Kurtuluş, Ming Y Lu, et al. A deep-learning model for transforming the style of tissue images from cryosectioned to formalin-fixed and paraffin-embedded. *Nature Biomedical Engineering*, 6(12):1407–1419, 2022.

Sidike Paheding, Ashraf Saleem, Mohammad Faridul Haque Siddiqui, Nathir Rawashdeh, Almabrok Essa, and Abel A Reyes. Advancing horizons in remote sensing: a comprehensive survey of deep learning models and applications in image classification and beyond. *Neural Computing and Applications*, 36(27):16727–16767, 2024.

Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Computer graphics and applications*, 21(5):34–41, 2002.

Daniel Shao, Richard J. Chen, Andrew H. Song, Joel Runevic, Ming Y. Lu, Tong Ding, and Faisal Mahmood. Do multiple instance learning models transfer? In *Forty-second International Conference on Machine Learning*, 2025.

Zhuchen Shao, Hao Bian, Yang Chen, Yifeng Wang, Jian Zhang, Xiangyang Ji, et al. TransMIL: Transformer based correlated multiple instance learning for whole slide image classification. *Advances in Neural Information Processing Systems*, 34:2136–2147, 2021.

Andrew H. Song, Richard J. Chen, Tong Ding, Drew F.K. Williamson, Guillaume Jaume, and Faisal Mahmood. Morphological prototyping for unsupervised slide representation learning in computational pathology. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11566–11578, June 2024.

Baochen Sun and Kate Saenko. Deep CORAL: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pp. 443–450. Springer, 2016.

Erik Thiringer, Fredrik K. Gustafsson, Kajsa Ledesma Eriksson, and Mattias Rantalainen. Scanner-induced domain shifts undermine the robustness of pathology foundation models, 2026.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 2962–2971. Ieee, 2017.

Rao Muhammad Umer, Daniel Sens, Jonathan Noll, Sohom Dey, Christian Matek, Lukas Wolfseher, Rainer Spang, Ralf Huss, Johannes Raffler, Sarah Reinke, Ario Sadafi, Wolfram Klapper, Katja Steiger, Kristina Schwamborn, and Carsten Marr. A multicenter benchmark of multiple instance learning models for lymphoma subtyping from HE-stained whole slide images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 6825–6834, June 2026.

Vladimir N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

Ben Vardi, Dana Schonberger, Yuval Friedmann, Zohar Yakhini, Iris Barshack, Alexander Loebel, and Ariel Shamir. Discrepancy minimization improves cross-hospital robustness in digital pathology, 2026.

Eugene Vorontsov, Alican Bozkurt, Adam Casson, George Shaikovski, Michal Zelechowski, Kristen Severson, Eric Zimmermann, James Hall, Neil Tenenholtz, Nicolo Fusi, et al. A foundation model for clinical-grade computational pathology and rare cancers detection. *Nature medicine*, 30(10): 2924–2935, 2024.

Guillaume Vray, Devavrat Tomar, Behzad Bozorgtabar, and Jean-Philippe Thiran. Distill-SODA: Distilling self-supervised vision transformer for source-free open-set domain adaptation in computational pathology. *IEEE transactions on medical imaging*, 43(5):2021–2032, 2024.

Jindong Wang et al. Everything about transfer learning and domain adaptation. `http://transferlearning.xyz`, 2018.

Jun Wang, Yu Mao, Nan Guan, and Chun Jason Xue. Advances in multiple instance learning for whole slide image analysis: Techniques, challenges, and future directions. *arXiv preprint arXiv:2408.09476*, 2024.

Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312: 135–153, October 2018. ISSN 0925-2312.

Garrett Wilson and Diane J Cook. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(5):1–46, 2020.

Andrew Zhang, Guillaume Jaume, Anurag Vaidya, Tong Ding, and Faisal Mahmood. Accelerating data processing and benchmarking of AI models for pathology, 2025.

Yongchun Zhu, Fuzhen Zhuang, Jindong Wang, Guolin Ke, Jingwu Chen, Jiang Bian, Hui Xiong, and Qing He. Deep subdomain adaptation network for image classification. *IEEE transactions on neural networks and learning systems*, 32(4):1713–1722, 2020.

Supplementary Material

A Implementation details

We provide the implementation details used across the evaluated PFMs and MIL architectures. We use the AdamW optimizer (Loshchilov & Hutter, 2019) with a cosine learning rate schedule (Loshchilov & Hutter, 2017), starting at 10^{-4} and decaying to zero. Each AdaptMIL stage uses its own cosine schedule (the schedule is restarted at the beginning of Stage 2). The classifier is the only module trained in both stages and is carried over from Stage 1 to Stage 2.

We use a batch size of 16, with each batch balanced across source and target domains (i.e., 8 source samples and 8 target samples). We additionally balance the labeled source samples within each batch by class, using an equal number of examples from each class. The adaptation coefficient λ is 2.5 for MMD, 1.0 for LMMD, 15 for CORAL, and 1.0 for DANN (also see Sec. D.4).

For the domain-adaptation objectives, we build on public implementations. We use multi-kernel MMD, CORAL, and DANN from TLlib (Jiang et al., 2020), and LMMD from Transfer Learning Repo (Wang et al., 2018). All experiments are conducted in mixed precision on a single NVIDIA GeForce RTX 5090 GPU.

AdaptMIL for ABMIL. Our ABMIL implementation is based on MIL-Lab (Shao et al., 2025). The slide embedding dimension and internal attention dimension are set to match the dimensionality of each PFM's patch representations.

AdaptMIL for DSMIL. We use the multi-class DSMIL formulation (Li et al., 2021), with the implementation from MIL-Lab (Shao et al., 2025). Unlike ABMIL, DSMIL produces a class-specific slide representation for each class rather than a single slide embedding. For domain adaptation, we concatenate the two class-specific representations and apply the adaptation objective to the resulting embedding. The remaining implementation details follow ABMIL, as DSMIL similarly applies a patch-level projection followed by attention-based aggregation.

AdaptMIL for TITAN. TITAN (Ding et al., 2025) consists of a pre-trained transformer that takes CONCH-1.5 (Lu et al., 2024) patch embeddings as input and produces contextualized patch representations, followed by a pooling module that aggregates them into a single slide embedding. A natural AdaptMIL decomposition would treat the transformer as the projection component and the pooling module as the aggregation component. However, due to memory constraints and TITAN's limited batch-size support, we keep the transformer frozen and insert a trainable linear projection between the transformer and the pooler. The projection is trained in Stage 1. In Stage 2, we keep it frozen and fine-tune the pooling module.

Baselines. We describe the implementation details of the different baselines. For a fair comparison, single-stage UDA uses the same training configuration as AdaptMIL, except that training is performed in a single stage. We train it for 80 epochs, matching the total number of epochs used across the two AdaptMIL stages.

For Macenko stain normalization (Macenko et al., 2009), we use the implementation from torchstain (Barbano & Pedersen, 2022). For each source hospital, we manually select a reference patch containing a substantial amount of tissue.

B Datasets

We evaluate on four tasks across four public datasets: TCGA (Komura et al., 2022) NSCLC subtyping, TCGA RCC subtyping, CAMELYON17 (Bandi et al., 2018) cancer detection, and TCGA KIRC survival prediction. Tab. 5 summarizes the distribution of samples across hospitals for these datasets. For CAMELYON17, we use only negative slides and slides containing macrometastases. For the in-house NSCLC dataset, we use 150 slides, balanced between lung adenocarcinoma and lung squamous cell carcinoma. For all datasets, we have assured no patient leakage between train and test sets.

Table 5: Dataset composition for (a) TCGA NSCLC subtyping, (b) CAMELYON17 cancer detection, (c) TCGA RCC subtyping, and (d) TCGA KIRC survival prediction. LUAD: lung adenocarcinoma; LUSC: lung squamous cell carcinoma; ccRCC: clear cell renal cell carcinoma; pRCC: papillary renal cell carcinoma; IGC: International Genomics Consortium; RUMC: Radboud University Medical Center; CWZ: Canisius-Wilhelmina Hospital; UMCU: University Medical Center Utrecht; RST: Rijnstate Hospital; LPON: Laboratorium Pathologie Oost-Nederland; UNC: University of North Carolina; MSKCC: Memorial Sloan Kettering Cancer Center; UOP: University of Pittsburgh; MDA: MD Anderson.

(a) TCGA NSCLC subtyping.

Hospital	LUAD	LUSC	Total
Asterand	34	50	84
Christiana Healthcare	47	20	67
Roswell Park	12	25	37
Univ. of Pittsburgh	22	13	35
Johns Hopkins	84	50	134
IGC	91	41	132
Total	290	199	489

(b) CAMELYON17 cancer detection.

Hospital	Negative	Macro	Total
RUMC	64	15	79
CWZ	58	12	70
UMCU	75	14	89
RST	60	19	79
LPON	61	26	87
Total	318	86	404

(c) TCGA RCC subtyping.

Hospital	ccRCC	pRCC	Total
IGC	53	31	84
UNC	33	16	49
MSKCC	139	35	174
Total	225	82	307

(d) TCGA KIRC survival prediction.

Hospital	Patients	Events	Censored
UOP	104	69	35
MSKCC	139	38	101
MDA	69	26	43
Total	312	133	179

We note that some hospital datasets are relatively small and may exhibit substantial class imbalance which reflects a realistic cross-hospital setting. Upon publication, we will release the exact slide lists used from all public datasets together with the code to facilitate reproducibility.

C Statistical tests

We provide details on the statistical tests reported in Sec. D for balanced accuracy and top-50 attention mass. Statistical significance is assessed using a two-sided Wilcoxon signed-rank test over the 30 source–target hospital combinations in the TCGA NSCLC dataset. For each source–target combination, we first average the relevant metric across the three cross-validation folds and then compute the difference between the compared methods (e.g., AdaptMIL vs. source-only ERM or single-stage UDA vs. source-only ERM). We note that the Wilcoxon signed-rank test assumes independence across these differences, whereas in our setting the same hospital appears in multiple source–target combinations. We therefore interpret the statistical results with this limitation in mind.

D Additional results

D.1 Full NSCLC results

We provide full results for the TCGA NSCLC subtyping experiments presented in Tab. 1 of the main paper. We report macro F1 and ROC-AUC results in Tab. 6. The trends are consistent with the balanced accuracy results: AdaptMIL improves performance with MMD, LMMD, CORAL, and DANN. The gains with DANN are generally smaller, consistent with known optimization challenges of gradient-reversal-based adversarial adaptation (Tzeng et al., 2017).

We additionally evaluate whether the balanced accuracy improvements of AdaptMIL over ERM are statistically significant, using a two-sided Wilcoxon signed-rank test with $\alpha = 0.05$. For MMD, LMMD, and CORAL, the improvements are statistically significant with UNI and H0-mini. With

Table 6: Cross-hospital TCGA NSCLC subtyping results for AdaptMIL with ABMIL, for (a) Macro F1 and (b) ROC-AUC. Metrics are reported on a 0–100 scale.

(a) Macro F1.

Method	UNI	CONCH-1.5	H0-mini
In-domain training	87.12 (+10.71)	88.89 (+5.05)	91.35 (+8.51)
Source-only (ERM)	76.41	83.83	82.84
Macenko	72.85 (-3.57)	83.22 (-0.61)	79.77 (-3.07)
Single-stage UDA (MMD)	78.95 (+2.54)	83.92 (+0.08)	78.43 (-4.41)
AdaptMIL-MMD (ours)	**83.54** (+7.12)	86.35 (+2.52)	86.27 (+3.43)
Single-stage UDA (LMMD)	69.36 (-7.05)	73.13 (-10.70)	60.67 (-22.17)
AdaptMIL-LMMD (ours)	80.86 (+4.44)	86.33 (+2.49)	85.81 (+2.97)
Single-stage UDA (CORAL)	70.29 (-6.12)	79.49 (-4.34)	61.10 (-21.74)
AdaptMIL-CORAL (ours)	83.14 (+6.73)	**86.42** (+2.58)	**86.53** (+3.69)
Single-stage UDA (DANN)	75.94 (-0.47)	82.98 (-0.85)	83.01 (+0.17)
AdaptMIL-DANN (ours)	78.62 (+2.20)	84.92 (+1.08)	85.24 (+2.40)

(b) ROC-AUC.

Method	UNI	CONCH-1.5	H0-mini
In-domain training	94.89 (+3.09)	95.44 (+0.31)	96.88 (+3.30)
Source-only (ERM)	91.80	95.13	93.57
Macenko	91.94 (+0.14)	94.82 (-0.32)	93.53 (-0.05)
Single-stage UDA (MMD)	89.93 (-1.87)	95.21 (+0.08)	88.12 (-5.46)
AdaptMIL-MMD (ours)	**94.38** (+2.59)	96.29 (+1.16)	**95.05** (+1.47)
Single-stage UDA (LMMD)	81.26 (-10.54)	84.19 (-10.94)	69.88 (-23.69)
AdaptMIL-LMMD (ours)	93.93 (+2.13)	95.90 (+0.77)	94.95 (+1.37)
Single-stage UDA (CORAL)	80.27 (-11.52)	89.67 (-5.46)	71.75 (-21.82)
AdaptMIL-CORAL (ours)	93.96 (+2.16)	**96.41** (+1.27)	94.74 (+1.16)
Single-stage UDA (DANN)	91.65 (-0.15)	94.29 (-0.84)	93.67 (+0.10)
AdaptMIL-DANN (ours)	94.12 (+2.32)	96.31 (+1.18)	94.65 (+1.08)

CONCH-1.5, AdaptMIL improves the average performance, but the improvements are not statistically significant. While these tests provide evidence of consistent gains across source–target combinations, AdaptMIL does not guarantee improvement for every individual combination. For example, with MMD, AdaptMIL matches or outperforms ERM in 86.7% of the combinations with UNI, but in only 63.3% of the combinations with CONCH-1.5.

D.2 Additional attention and optimization analysis

To further support the attention analysis in Sec. 3.2 and Fig. 2a, we perform two-sided Wilcoxon signed-rank tests ($\alpha = 0.05$) on the top-50 attention mass in the TCGA NSCLC dataset. Across all 12 PFM and adaptation method combinations, single-stage UDA yields a higher average top-50 attention mass than ERM on held-out target slides, with the difference being statistically significant in 11 of the 12 settings.

Conversely, AdaptMIL produces a more distributed attention pattern than ERM in all 12 settings, with statistically significant differences in 11 of the 12 settings. We hypothesize that this occurs because Stage 1 learns the projection using mean pooling, which aggregates patch representations uniformly, and this affects the attention patterns subsequently learned in Stage 2.

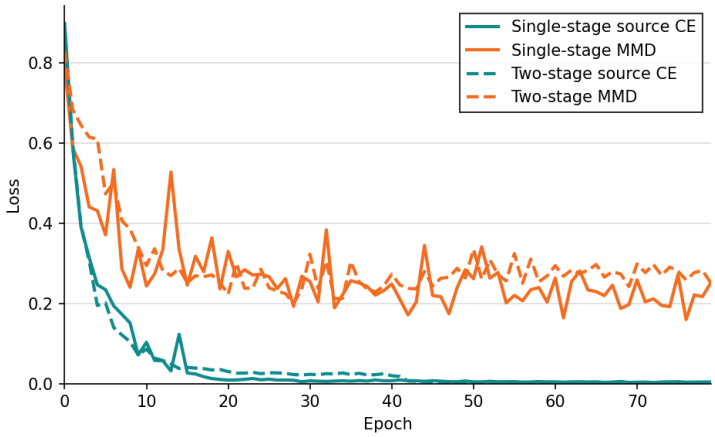


Figure 4: Representative training dynamics of single-stage UDA and AdaptMIL, showing similar behavior of the source cross-entropy and MMD losses during training. For AdaptMIL, the MMD loss is not optimized during Stage 2 (after epoch 40) and is shown only for monitoring.

Lastly, for completeness, we show that the source cross-entropy and domain adaptation losses exhibit similar dynamics in single-stage UDA and AdaptMIL. Fig. 4 extends Fig. 2b by also showing the AdaptMIL losses, which follow a similar pattern. Despite the similar optimization behavior, single-stage UDA performs 3.21% below ERM in this example, whereas AdaptMIL improves over ERM by 17.65%, further supporting our hypothesis that single-stage UDA exploits a shortcut.

D.3 Survival prediction

In the main paper, we evaluate AdaptMIL on three classification tasks. Here, we additionally consider a non-classification task: survival prediction, which ranks patients according to their disease-specific survival risk. We use 312 patients from the TCGA kidney renal clear cell carcinoma (KIRC) cohort across three hospitals. Evaluation is performed using leave-one-out hospital evaluation with 5-fold cross-validation, and performance is measured using the concordance index (C-Index).

Implementation uses the same ABMIL-based training framework, replacing the classification head and cross-entropy loss with a single-output risk head that predicts each patient's log relative hazard. The model is trained with the negative Cox partial log-likelihood.

Tab. 7 presents the survival prediction results. AdaptMIL improves over ERM for all evaluated PFMs. However, unlike the classification tasks, single-stage UDA is also effective and even outperforms AdaptMIL, although the margins are mostly small.

We observe an increase in attention concentration under single-stage UDA, as shown in Fig. 2a for NSCLC subtyping and in the other classification tasks. However, the increase in top-50 attention mass is much smaller for survival prediction. For example, with UNI and MMD it increases by only $1.2\times$, compared with $2.8\times$ for NSCLC subtyping. This suggests that the interaction between domain adaptation and MIL aggregation may depend on the downstream task. Nevertheless, AdaptMIL still improves over source-only training across all evaluated PFMs, suggesting a more robust choice when the effectiveness of single-stage UDA is uncertain.

D.4 Additional ablations

We evaluate the sensitivity of AdaptMIL to the domain adaptation coefficient λ in Tab. 8. AdaptMIL is relatively insensitive to the choice of λ across MMD, LMMD, and CORAL. For example, across different MMD coefficients, the difference between the best and worst balanced accuracy is only 0.96%.

Table 7: Cross-hospital survival prediction results for AdaptMIL-MMD with ABMIL. The metric is C-index.

Method	UNI	CONCH-1.5	H0-mini
In-domain training	0.6303 (+0.0173)	0.6185 (-0.0162)	0.6352 (+0.0105)
Source-only (ERM)	0.6130	0.6346	0.6247
Single-stage UDA	**0.6406** (+0.0276)	**0.6494** (+0.0147)	**0.6520** (+0.0272)
AdaptMIL (ours)	0.6372 (+0.0242)	0.6490 (+0.0144)	0.6377 (+0.0130)

Table 8: Sensitivity of AdaptMIL to the domain adaptation coefficient λ on NSCLC subtyping with UNI and ABMIL.

(a) MMD.

Objective	λ	Balanced Acc.	Δ
MMD	0.5	84.84	+5.07
	1.0	85.22	+5.44
	1.5	85.48	+5.70
	2.0	85.67	+5.89
	2.5	**85.80**	**+6.02**
	3.0	**85.80**	**+6.02**
	3.5	85.57	+5.79
	4.0	85.74	+5.96
	4.5	85.70	+5.92
	5.0	85.64	+5.87
	10.0	85.46	+5.68
	15.0	85.41	+5.63
	20.0	85.42	+5.64

(b) LMMD and CORAL.

Objective	λ	Balanced Acc.	Δ
LMMD	0.5	83.09	+3.31
	1.0	83.77	+3.99
	1.5	**84.16**	**+4.39**
	2.0	83.77	+4.00
	2.5	83.21	+3.43
CORAL	1.0	82.94	+3.16
	1.5	83.66	+3.88
	2.5	83.94	+4.16
	5.0	84.74	+4.96
	10.0	85.13	+5.36
	15.0	**85.21**	**+5.43**
	20.0	85.14	+5.36

D.5 Runtime

We measure the training time of AdaptMIL-MMD with UNI. Training a single source–target model takes 27.6 seconds on average for a TCGA NSCLC fold, compared with 47.6 seconds for single-stage UDA and 21.7 seconds for source-only ERM. AdaptMIL is therefore approximately $1.7\times$ faster than single-stage UDA. This is because Stage 1 uses the computationally cheaper mean-pooling aggregation, and Stage 2 does not compute the domain adaptation objective. In contrast, single-stage UDA uses both learned MIL aggregation and the adaptation objective throughout training.