IN-CONTEXT MULTIPLE INSTANCE LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multiple Instance Learning (MIL) addresses problems where supervision is available at the level of bags of instances and has been successfully applied in fields ranging from computational pathology to satellite imagery. Nevertheless, existing algorithms struggle in the low-label regime that characterizes many real-world applications. Flexible models overfit and rigid ones fail to adapt to the task at hand. We show that pretraining an in-context learner with a Perceiver-style architecture on synthetic data yields a model that can solve new tasks from a handful of labeled bags. At inference time, classification happens in a single forward pass and requires no gradient updates. We propose and investigate synthetic data generators for bag-structured data and find that they capture complementary inductive biases. A model pretrained on a mixture of these generators inherits their per-task strengths and achieves the best average performance across twelve MIL benchmarks, outperforming supervised baselines that require task-specific training.

1 INTRODUCTION

Multiple Instance Learning (MIL) is a widely used framework for problems where labels are available only at the level of collections (*bags*) of instances rather than for individual ones. Applications of MIL span computational pathology, drug activity prediction, satellite imagery, and text classification (Otsu et al., 2023; Waqas et al., 2024; Idaji et al., 2026; Jeong et al., 2026). A persistent challenge in many of these domains is label scarcity, because MIL datasets often contain only a few dozen labeled bags (Dietterich et al., 1997; Ehteshami Bejnordi et al., 2017). Supervised training of a robust aggregation function on such limited data is fragile as flexible models such as attention-based MIL (Ilse et al., 2018) and TransMIL (Shao et al., 2021) may overfit, while restrictive models encode strong inductive biases that may not match the task at hand (Andrews et al., 2002).

Supervised MIL pretraining (Shao et al., 2025) can mitigate this for applications with access to large labeled source corpora with similar examples, but many applications lack such resources and transfer is not always guaranteed. Self-supervised methods that learn bag-level representations (Xu et al., 2024; Ding et al., 2025) remove the need for pretraining labels but must compress instance information into a fixed representation without knowledge of the downstream task, potentially discarding features that are critical for a specific problem. Thus, so far no method enables learning from a small number of labeled bags at inference time without also requiring a large task-relevant pretraining corpus.

To close this gap, we introduce *In-Context Multiple Instance Learning* (ICMIL). Specifically, we build on the Prior-data Fitted Network (PFN) paradigm (Müller et al., 2022) and train a model on diverse synthetic bag-structured data to approximate the posterior predictive distribution over bag labels. Intuitively, by training across many simulated MIL tasks, the model learns what plausible bag classifiers look like, so at test time it can infer the labeling rule and use it to predict labels for unseen bags. This shifts the problem from fitting an aggregator on a handful of bags to designing good data generators (priors) over bag-structured tasks that span the relevant hypothesis space. At inference time, the model is given a real MIL dataset as context and classifies the test bags in a single forward pass, without gradient updates or task-specific training. The trained model approximates Bayesian model averaging over hypotheses consistent with the labeled context and avoids the variance that comes with fitting a single aggregator on scarce labels (Geman et al., 1992). We illustrate this favorable bias-variance trade-off in Figure 1 and show that our ICMIL achieves both higher median AUROC and lower variance across training-set resamples than supervised baselines in the low-sample regime.

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.

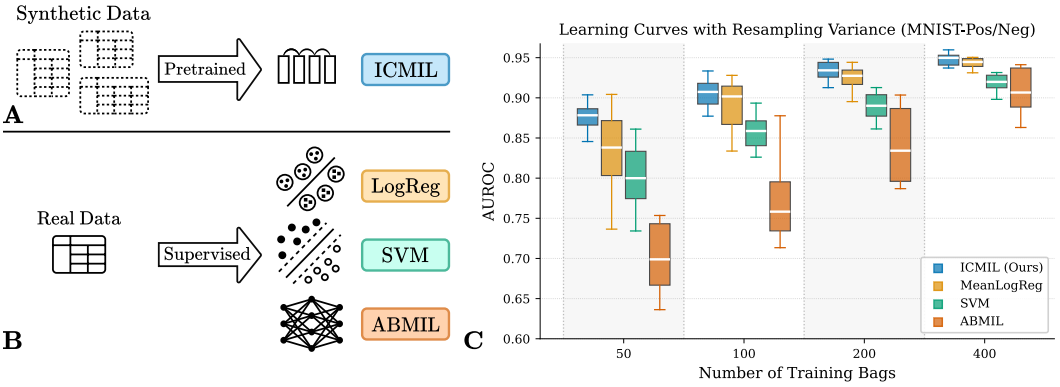

Figure 1: (**A**) ICMIL is pretrained on synthetic bag-structured data and classifies new tasks in a single forward pass without gradient updates or hyperparameter tuning. (**B**) Supervised methods (MeanLogReg, SVM, ABMIL) are retrained and tuned from scratch on each downstream dataset. (**C**) Resampling variance on MNIST-Pos/Neg, a binary bag classification task where the label depends on the balance of positive and negative digit classes. Boxplots show the distribution of AUROC over 20 independently resampled training sets with a fixed test set. ICMIL achieves the highest median AUROC and the lowest variance across training set draws, with the advantage most pronounced in the low-bag regime.

While PFNs have been successfully applied to flat tabular classification (Hollmann et al., 2023; Qu et al., 2025) and regression (Hollmann et al., 2025), adapting them to bag-structured data requires designing both architectures suited to in-context learning over hierarchical sets and synthetic priors that capture MIL reasoning patterns. On the architectural side, challenges are that (i) attention over all instances across all bags scales quadratically in both, (ii) bag-level compression should be task-aware and take bag labels into account, and (iii) the model should be permutation-invariant within bags while preserving bag identity across them. In this paper, we propose to address these with a Perceiver-style architecture in which learnable bag tokens iteratively gather information from their instances and exchange it with representations and labels of other bags.

On the prior side, it is unclear what a good data generator for MIL problems looks like. Standard MIL architectures assume that bag labels factorize through a permutation-invariant decomposition over instances, and the natural question is whether this assumption should also be reflected in the prior. We design two families of generators that take opposite stances. Factorized priors encode the classical decomposition, while joint priors drop it in favor of a single generator over the full bag that permits richer dependence structures among instances. We study whether either family suffices on its own or whether a mixture yields a single model that performs robustly across MIL regimes.

We summarize our contributions as follows:

- We introduce *In-Context Multiple Instance Learning* (ICMIL) and identify key architectural challenges that arise when dealing with bag-structured data. We propose a Perceiver-style architecture for hierarchical set inputs as a solution.
- We propose data generators that encode different MIL assumptions and find that they yield models with complementary strengths. Each model excels on a distinct subset of real benchmarks.
- We show that ICMIL, a model trained on a mixture of synthetic priors, achieves the best average AUROC and rank across twelve MIL benchmarks in the low-label regime. It outperforms supervised baselines despite using no gradient updates or hyperparameter tuning.

2 Background

2.1 Multiple Instance Learning

In MIL, each data point is a *bag* $B = \{x_1, \ldots, x_I\}$ of I instances $x_i \in \mathbb{R}^F$, annotated with a single bag-level label y, which may represent a classification, regression, or time-to-event target.

The number of instances I may vary across bags. The original MIL formulation (Dietterich et al., 1997; Maron & Lozano-Pérez, 1997) assumes that instances within a bag are statistically independent and permutation-invariant, and that each instance carries an unknown binary label $y_i \in \{0, 1\}$ determining the bag label via $y = \max_i\{y_i\}$. These assumptions have been relaxed to accommodate a wider range of applications, including more general bag-label functions capturing complex instance-label relationships (Foulds & Frank, 2010; Carbonneau et al., 2018; Hense et al., 2024), or instance dependencies arising from shared generative structure such as spatial proximity in pathological tissue (Zhao et al., 2020; Shao et al., 2021). Traditionally, MIL models rely on pre-defined pooling functions, such as max or mean pooling, to aggregate instance predictions into a bag prediction (Dietterich et al., 1997; Campanella et al., 2019). Ilse et al. (2018) introduced learned instance aggregation via attention, enabling the model to identify instances most predictive for a given task. A plethora of extensions have been proposed, e.g., exploiting self-attention to explicitly model dependencies between instances (Shao et al., 2021). However, these approaches depend on weakly supervised learning from task-specific labeled bags, making them prone to overfitting and shortcut learning (Howard et al., 2021; Kömen et al., 2025; Drexlin et al., 2025; Möllers et al., 2026; Dawood et al., 2026), particularly when training data is scarce. To reduce reliance on labeled training data, several directions for zero- or few-shot MIL have been explored: leveraging vision-language models (Lu et al., 2023; Meseguer et al., 2026), self-supervised pretraining (Xu et al., 2024; Ding et al., 2025), and transfer learning across related tasks (Shao et al., 2025). However, such methods constrain the flexibility of the aggregation mechanism and remain dependent on domain-specific pretraining, limiting their applicability across diverse tasks and domains. In contrast, our ICL approach requires no task-specific labeled training data and generalizes across tasks and application domains without finetuning. By combining cross-bag, cross-instance, and cross-feature attention, ICMIL enables both task-specific instance aggregation and effective few-shot inference.

2.2 Prior-data Fitted Networks

Prior-data Fitted Networks (PFNs) (Müller et al., 2022) amortize Bayesian inference over a prior of data-generating processes via in-context learning. A transformer is pretrained on synthetic datasets sampled from a prior over data-generating processes to predict labels of held-out points given the remaining examples as context. At inference time, a PFN directly outputs predictions for a new dataset in a single forward pass, without gradient-based optimization. Performance depends on the alignment between the prior and the target task, motivating careful prior design. Recently, PFNs have matched or surpassed classical methods on tabular prediction benchmarks (Hollmann et al., 2023; 2025) and have been extended to causal inference (Robertson et al., 2025), time series forecasting (Hoo et al., 2025), Bayesian optimization (Müller et al., 2023) and single-cell perturbation effect estimation (Sextro et al., 2026). Closest to our work, Kopp et al. (2025) adapts TabPFN to multi-instance regression by collapsing each bag into a tabular input through k-means pooling. However, their aggregation is unsupervised and fixed before inference, which limits the bag-label functions it can represent (Appendix C). In contrast, we directly train a PFN over bag-structured inputs with priors and an architecture that can use MIL labels in-context.

3 In-Context Learning for Multiple Instance Problems

We now formulate the in-context learning problem for MIL. Let $p(\mathcal{D})$ be a prior over MIL datasets that allows us to sample a dataset $\mathcal{D} = \{(B_i, y_i)\}_{i=1}^{N+1}$ of bags and labels. The learner q_θ receives the full set of N labeled context bags as direct input and has to infer y_{N+1} for a query bag B_{N+1}. We train q_θ to minimize the expected negative log-likelihood

$$\mathcal{L}(\theta) = \mathbb{E}_{\mathcal{D}\sim p(\mathcal{D})}\left[-\log\, q_\theta(y_{N+1} \mid B_{N+1}, \{(B_i, y_i)\}_{i=1}^{N})\right]. \tag{1}$$

To design a learner that minimizes the objective, one has to take some particular challenges into account that come with the bag-structured nature of MIL data.

Challenge 1: Computational scalability. A dataset of N bags with I instances contains $N \times I$ feature vectors. Naive attention over all instances across all bags scales as $\mathcal{O}(N^2I^2)$ in compute. This quickly becomes prohibitively expensive as MIL datasets frequently have hundreds or thousands of instances per bag.

Challenge 2: Task-dependent instance compression. A naive solution to Challenge 1 would be to first aggregate instance-level features into bag representations, and then perform a full attention over these compressed representations and the corresponding labels. In this case, compression happens before the learner can attend across labels and bags, and cannot use task-specific information to determine which instance-level features to preserve. For example, the same bag of tissue patches could be used for a tumor detection task because of a single atypical cell, or for a subtyping task because of the overall tissue architecture. Any compression layer must therefore be able to incorporate label information across all bags to decide which instance-level features to preserve.

Challenge 3: Permutation invariance while preserving bag identity. Datasets in typical MIL problems are often sets. That is, there is no specific ordering of the instances of a bag and the model should be invariant to any permutation of the instances within bags of a dataset. At the same time, the model needs to recognize which instances belong to the same bag.

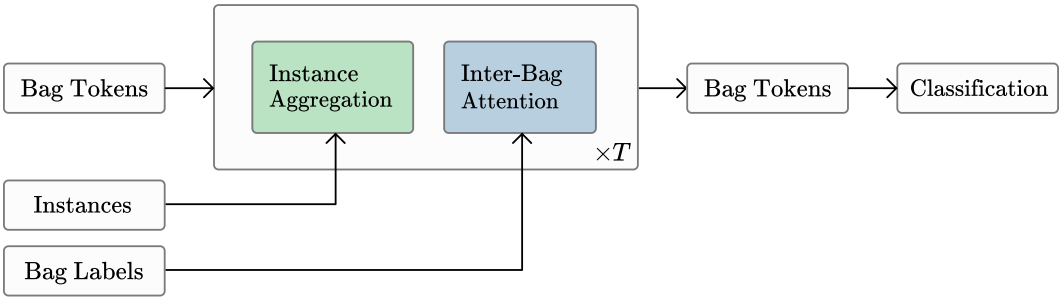

Figure 2: Schematic of the ICMIL architecture, illustrated for T iterations. A learnable bag token $\mathbf{Q}_b \in \mathbb{R}^{G\times E}$ is initialized for each bag. At every iteration, each bag token is first updated by **instance aggregation**, where it cross-attends to its own instance tokens (applied independently per bag). It is then concatenated with the bag-label embedding $\tilde{\mathbf{y}}_b$ and updated by **inter-bag attention**, a column-row self attention over all bags. After T iterations, the label token of each query bag is decoded into the predicted class distribution.

3.1 An Architecture for ICMIL

We now show how we can design a Perceiver-style (Jaegle et al., 2021) architecture that resolves the above challenges. To implement the architecture, we follow Hollmann et al. (2025) and embed each of the F input features into groups of size s, yielding $G = \lceil F/s \rceil$ feature-group embeddings of dimension E per instance. This produces a tensor $\tilde{\mathbf{X}} \in \mathbb{R}^{N\times I\times G\times E}$ of instance embeddings. We further embed the bag labels into $\tilde{\mathbf{y}} \in \mathbb{R}^{N\times E}$, with test bags receiving the mean of the training label embeddings. We then instantiate a learnable token $Q_b \in \mathbb{R}^{G\times E}$ for each bag and update it over T steps of alternating cross attention from a bag vector to its instance vectors and self attention among bag vectors (Figure 2):

Instance aggregation / Cross attention: Bag tokens attend to instance tokens. The G latent bag-level tokens $\mathbf{Q}_{b,g} \in \mathbb{R}^E$ are updated by attending to their respective instance tokens across the corresponding feature group:

$$\mathbf{Z}_{b,g}^{(t)} = \mathbf{Q}_{b,g}^{(t)} + \mathrm{MHA}\Big(\mathbf{Q}_{b,g}^{(t)},\, \tilde{\mathbf{X}}_{b,g},\, \tilde{\mathbf{X}}_{b,g}\Big) \tag{2}$$

where $\mathrm{MHA}(Q, K, V)$ denotes multi-head attention and $\tilde{\mathbf{X}}_{b,g} \in \mathbb{R}^{I\times E}$ are the instance tokens of bag b for feature group g. Since the operation is independent per bag, we can chunk over the bags N and process only one bag at a time, reducing peak memory from $\mathcal{O}(N \cdot I \cdot G)$ to $\mathcal{O}(I \cdot G)$.

Inter-bag attention / Self attention: Bag tokens attend to each other. After instance aggregation, the label embedding $\tilde{\mathbf{y}}_b \in \mathbb{R}^E$ is appended to $\mathbf{Z}_b^{(t)}$, so that each bag is represented by $G + 1$ tokens of size E. Stacked across all bags, this yields a matrix $\hat{\mathbf{Z}}^{(t)} \in \mathbb{R}^{N\times(G+1)\times E}$ that is processed with column-row attention (Hollmann et al., 2025):

$$\mathbf{Q}^{(t+1)} = \mathrm{ColRowAttn}\Big(\hat{\mathbf{Z}}\Big) \tag{3}$$

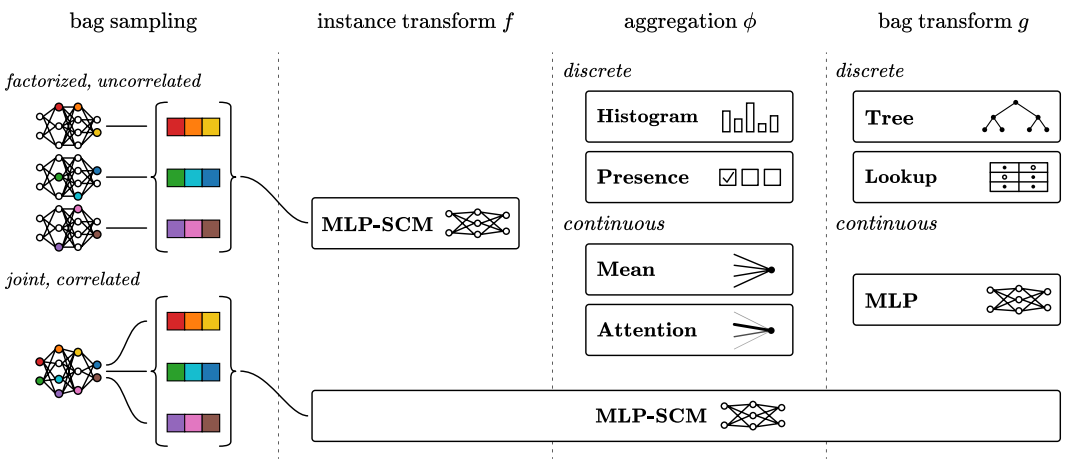

Figure 3: Taxonomy of bag-structured priors for ICMIL. **Bag sampling**: factorized priors draw each instance from an independent SCM (top), producing uncorrelated bags, while joint priors draw all instances from a single SCM (bottom), inducing within-bag feature correlations. **Instance transform** f: induced by the SCM as the mapping from the feature nodes x_i to a designated output node along the causal graph. **Aggregation** ϕ: for factorized priors, the per-instance outputs are compressed by a continuous or discrete permutation-invariant summary. **Bag transform** g: the bag summary is decoded into a label via a tree, a lookup table over discrete class patterns, or an MLP.

where column attention operates over feature groups and labels, and row attention allows for the flow of information between bags. Importantly, test bags can only attend to training context bags, but not to each other. The compute for this step scales as $\mathcal{O}(N^2 + G^2)$, with memory linear in $\mathcal{O}(N \cdot G)$ using flash attention. Removing this step makes in-context learning impossible (Appendix B).

After T iterations, the label token of each query bag is passed through a decoder to produce the predicted class distribution. The architecture directly addresses the three challenges identified above. Importantly, instance aggregation reduces the cost and complexity compared to naive attention over all instances (Challenge 1). Furthermore, the iterative attention mechanism naturally resolves the task-dependent compression problem. The model does not need to compress instances into bag representations in a single label-agnostic step, but can alternate between attending to instances and attending across bags and labels (Challenge 2). Finally, since MHA is permutation-invariant over its inputs, the architecture is invariant to instance ordering within bags while preserving bag identity across bags (Challenge 3). Appendix B validates these choices empirically, showing that both the iterative aggregation and the attention between bags are necessary.

3.2 Synthetic Priors for ICMIL

The effectiveness of ICMIL depends on the alignment between its training prior $p(\mathcal{D})$ and the structure of real MIL tasks. We therefore investigate the effect of various prior designs that capture different distributional properties of bag-structured data. We broadly divide our priors into factorized and joint priors, illustrated in Figure 3.

Factorized priors. These priors follow the dominant MIL modeling approach and assume the bag label is determined by a permutation-invariant function of the instances (Zaheer et al., 2017):

$$y = g\big(\phi\big(\{f(x_i)\}_{i=1}^{I}\big)\big)\,, \tag{4}$$

where $f\colon \mathbb{R}^F \to \mathbb{R}^d$ is an instance-level transform, ϕ is an aggregation that compresses the instance outputs into a bag summary, and g maps that summary to the bag label. For each instance, we use an MLP-SCM (Qu et al., 2025) with randomly parameterized nonlinear relationships to jointly generate both the instance features $x_i \in \mathbb{R}^F$ and the instance-level outputs $f(x_i) \in \mathbb{R}^d$ from the same causal graph. Latent causes are sampled independently per instance, so that $p(x_1, \ldots, x_I \mid y) = \prod_i p(x_i \mid y)$. For ϕ we consider *discrete* and *continuous* summaries. Discrete summaries map the output of f to one of K discrete classes and compute either a histogram of class counts or a presence indicator of which classes appear. Continuous summaries pool the raw instance embeddings via mean pooling

or ABMIL pooling (Ilse et al., 2018). For g we investigate an MLP, a tree, and a lookup table over discrete instance-class patterns.

Joint priors. These priors drop the f-g decomposition and instead define a single function over the flattened bag $[x_1; \ldots; x_I] \in \mathbb{R}^{I \cdot F}$:

$$y = f_{\mathcal{S}}(x_1, \ldots, x_I). \tag{5}$$

We use an MLP-SCM (Qu et al., 2025) to jointly generate the concatenated instance features $[x_1; \ldots; x_I]$ and the bag label y from the same causal graph, with latent causes sampled once per bag. This allows for inter-instance correlations that the factorized prior cannot express, and in general we have $p(x_1, \ldots, x_I \mid y) \neq \prod_i p(x_i \mid y)$.

The two prior families differ in important ways that may influence downstream performance. Factorized priors are structurally aligned with classical MIL assumptions (permutation invariance, instance-level decomposition, witness-style aggregation), while joint priors are more expressive and induce within-bag feature correlations that the factorized family cannot express. We empirically investigate whether structural alignment or expressiveness leads to better downstream performance, and whether the resulting differences in feature correlation structure influence which prior is best suited to a given MIL task.

4 Experiments

We evaluate the proposed in-context learner on twelve MIL benchmarks. We investigate (i) whether the different priors encode complementary inductive biases and MIL regimes, (ii) whether a single model trained on a mixture of priors can inherit their individual strengths, and (iii) whether that model is more sample-efficient than standard MIL baselines in the low-label regime that frequently occurs in practical applications. We make our code available at `https://anonymous.4open.science/r/ICMIL-62E2`.

4.1 Experimental Setup

Training. We pre-generate synthetic datasets from each prior configuration and store them separately. For both the MLP and Tree components of our priors, we adapt the implementation from Qu et al. (2025) and use their default hyperparameter sampling ranges. The generated data is split into context and test bags and retrieved during training. For the prior ablations, we use a reduced setup with models of 6 iterations, 4 attention heads, embedding dimension 128, MLP hidden size 512, and feature group size 1, trained on instances with 25 features. Models are trained for $20{,}000$ steps with batch size 128. The number of bags per dataset is sampled uniformly in $[70, 125]$. Based on the results in this reduced setting, we train a final model with increased embedding dimension and training duration in §4.4. Further training details can be found in Appendix A.

Benchmarks. We evaluate on twelve multiple-instance learning benchmarks. Six benchmarks expose a witness label rule (SMIL, Musk1, Musk2, Letters, HEPMASS, RSNA-ICH) and six rely on interactions across instances (Elephant, Fox, Tiger, TCGA, Adjacent Pairs, Pos/Neg). We subsample large benchmarks (e.g. RSNA-ICH, TCGA LUAD-LUSC) and use them to create tasks with approximately 100 bags that fall into the low-sample regime. We apply PCA to the benchmarks whose feature dimension exceeds 25. The full per-benchmark descriptions are located in Appendix D.

Baselines. We compare our method against eight deliberately heterogeneous MIL baselines. **MeanLogReg:** We fit scikit-learn's `LogisticRegressionCV` on the mean of the features of each bag. We apply 5-fold stratified CV over $C \in \{10^{-2}, 10^{-1}, 1, 10, 10^2\}$. **SVM-Summ:** For each feature we compute six fixed summary statistics (sum, mean, median, min, max, stdv) and concatenate them to form one vector per bag. We then fit an RBF-kernel SVM, with $C \in \{10^{-2}, 10^{-1}, 1, 10, 10^2\}$ tuned by stratified 5-fold cross-validation. **ABMIL:** We train an ABMIL model with embedding dimension 500 and attention dimension 128, selecting the learning rate from $10^{-2}, 5{\times}10^{-3}, 10^{-3}, 5{\times}10^{-4}, 10^{-4}$ and weight decay from $0, 10^{-4}, 5{\times}10^{-4}$ using a single stratified 10% held-out validation split. Each candidate is trained for up to 200 epochs in full-batch mode with Adam and early stopping (patience 20). The configuration with the lowest validation

Prior	Wit/Uncorr	Wit/Corr	Int/Uncorr	Int/Corr	Overall
Joint	2.60	**1.00**	**1.00**	**1.75**	**1.92**
Fact. (cont, MLP)	**1.80**	2.00	2.25	2.12	2.00
Fact. (cont, tree)	3.40	3.00	2.75	2.75	3.04
Fact. (disc, lookup)	2.40	4.00	4.00	3.38	3.12
Fact. (disc, MLP)	4.80	5.00	5.00	5.00	4.92
Fact. (disc, tree)	6.00	6.00	6.00	6.00	6.00

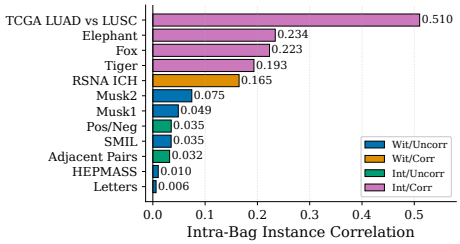

Figure 4: **Left:** Mean rank across MIL benchmarks for models trained on joint and factorized priors, broken down by group and pooled over all datasets (Overall). Lower is better. Wit/Corr contains a single benchmark (RSNA-ICH). **Right:** Within-bag feature correlation (ICC) per benchmark and colored by task groups. ICC quantifies the share of feature variance explained by bag membership. High values indicate strong within-bag correlation.

cross-entropy is used as the final model. **DSMIL:** We train the dual-stream model of Li et al. (2021) with embedding dimension 512 and attention dimension 384, and tune the learning rate and weight decay with the same protocol as for ABMIL. **ACMIL:** We train the gated-attention model of Zhang et al. (2024) with embedding dimension 256, attention dimension 128, 5 attention branches, top-10 masked patches and a mask-drop rate of 0.6, and tune the learning rate and weight decay with the same protocol. **TabPFN-Concat:** We flatten each bag to a single vector by concatenating the instances and pass it through TabPFN-v2 (Hollmann et al., 2025). In cases where the flattened vector exceeds 500 features we truncate to avoid OOM errors. **TabPFN-Subsample:** We randomly subsample whole instances per bag, flatten them and predict a label. We repeat this 10 times and average the logits to obtain the final prediction. **TabPFN-Cluster:** Following Kopp et al. (2025), we cluster all training instances with K-means (K=5). We then assign all instances in the bags a cluster label. For each bag, the instances belonging to the same cluster are summed and scaled by the ratio of total bag size to cluster instance count. This produces one feature vector per bag per cluster. TabPFN-v2 is then called for each cluster vector and bag giving logits for each cluster. The final prediction is the average of the resulting logits across all clusters the bag participates in.

4.2 Different Priors Capture Different MIL Regimes

We train one model per prior with the reduced setup and report per-benchmark AUROC in Table 1. We group the benchmarks by label rule (witness vs. interaction) and within-bag feature correlation, measured by the intra-class correlation coefficient (ICC) of instance features (Figure 4).

The Joint prior achieves the best average rank and is especially strong in settings where within-bag correlation and/or interaction are present. Nevertheless, the factorized priors outperform it by large margins on the uncorrelated witness benchmarks. The Factorized(disc, lookup) prior that was explicitly designed for such tasks has a large advantage on Letters (95.2 ± 0.5), SMIL (74.9 ± 0.3) and HEPMASS (86.1 ± 1.0) where it improves over the Joint prior by up to 4 percentage points. The Factorized(cont, MLP) prior takes the lead on Musk1 (93.9 ± 0.6) and Musk2 (85.4 ± 3.2). The Factorized(disc, tree) prior collapses to near-chance performance across all benchmarks, indicating that this combination is not suitable for real MIL tasks.

These results address the questions raised in §3.2. The more expressive joint prior achieves the best average rank, but the factorized priors that are structurally aligned with the MIL paradigm outperform it on uncorrelated witness benchmarks. Interestingly, the factorized priors do not have an advantage on the correlated witness task RSNA-ICH. This suggests that the correlation structure of the instance features might play a larger role than the label rule in determining the optimal prior. Overall, we observe that greater expressiveness does not always translate into better downstream performance and that aligning priors to target tasks can yield measurable advantages. The fact that no single prior dominates motivates the prior-mixing experiment in the next section.

4.3 Mixing Priors Yields a Robust Generalist

The previous results show that priors tailor to specific downstream tasks. We test whether a single model can inherit these per-task strengths by training on a weighted mixture of the three priors

Table 1: Per-benchmark mean AUROC (%) ± standard error for each model. **Bold** marks the best result per benchmark and underline the second-best (ties marked for both). The two leftmost numeric columns report each model's mean AUROC across all benchmarks and its mean rank (lower is better). For the prior ablations, we use a reduced setup with 20,000 training steps, embedding dimension 128, and MLP hidden size 512, while the scaled setup used for the final ICMIL model doubles each of these.

Model	Avg AUROC	Avg Rank	Musk1	Musk2	HEPMASS	Letters	SMIL	Adj. Pairs	Pos/Neg	RSNA ICH	Elephant	Fox	Tiger	TCGA
Baselines														
MeanLogReg	82.37	7.46	91.6 ±0.5	82.1 ±1.6	86.9 ±0.3	92.3 ±1.0	72.4 ±0.2	73.9 ±0.1	85.4 ±0.5	74.6 ±0.0	93.3 ±0.1	62.5 ±0.6	86.3 ±0.2	87.1 ±0.2
SVM-Summ	79.38	7.08	**94.7** ±0.2	81.3 ±0.7	42.1 ±5.1	93.9 ±0.5	66.5 ±1.9	77.7 ±0.8	84.5 ±0.2	**77.0** ±0.4	92.5 ±0.1	65.7 ±0.7	88.5 ±0.7	88.2 ±0.2
ABMIL	79.97	9.17	85.7 ±3.5	75.4 ±2.8	75.7 ±1.7	97.6 ±0.8	85.4 ±0.5	65.5 ±1.1	74.6 ±1.6	71.2 ±0.3	94.1 ±0.2	59.8 ±1.6	83.9 ±1.5	**90.7** ±1.6
DSMIL	80.33	8.58	86.2 ±1.8	84.0 ±3.9	84.5 ±1.1	97.6 ±1.2	80.0 ±1.7	65.2 ±1.4	69.9 ±1.5	69.5 ±0.7	93.4 ±0.4	61.2 ±0.6	84.7 ±3.0	87.8 ±1.9
ACMIL	83.81	5.88	89.0 ±1.3	83.9 ±2.5	81.1 ±3.9	98.7 ±1.3	**87.7** ±1.0	72.5 ±0.6	80.8 ±0.4	74.8 ±1.0	94.1 ±0.2	64.4 ±1.7	88.0 ±1.2	**90.7** ±0.7
TabPFN-Concat	74.16	11.50	91.1 ±0.0	79.3 ±0.0	66.4 ±0.0	88.8 ±0.0	56.3 ±0.0	57.5 ±0.0	63.7 ±0.0	63.9 ±0.0	91.4 ±0.0	63.9 ±0.0	**89.6** ±0.0	78.0 ±0.0
TabPFN-Subsample	74.82	10.67	92.5 ±0.2	78.6 ±6.2	67.7 ±0.3	88.8 ±0.0	58.2 ±0.2	57.0 ±0.1	65.4 ±0.3	64.1 ±0.0	92.0 ±0.0	65.7 ±0.2	**89.6** ±0.1	78.2 ±0.1
TabPFN-Cluster	81.21	6.67	88.0 ±1.0	**91.3** ±0.4	67.5 ±3.0	**99.2** ±0.8	77.8 ±1.0	71.4 ±0.4	77.0 ±1.8	76.9 ±0.1	**95.6** ±0.2	**66.2** ±0.0	86.5 ±0.3	77.1 ±0.2
Prior Ablations (reduced setup)														
Joint	82.56	5.79	91.0 ±0.3	83.2 ±3.4	81.9 ±1.1	90.9 ±1.0	74.5 ±0.6	**78.3** ±0.6	86.9 ±0.5	74.9 ±0.3	95.0 ±0.4	56.1 ±3.4	89.2 ±0.1	88.8 ±0.2
Fact. (cont, MLP)	82.76	6.50	93.9 ±0.6	85.4 ±3.2	84.3 ±0.7	92.3 ±1.0	71.4 ±0.8	77.9 ±0.1	86.3 ±0.3	73.9 ±0.2	93.7 ±0.3	58.2 ±2.0	88.9 ±0.1	86.9 ±0.4
Fact. (cont, tree)	81.55	8.71	90.3 ±1.0	82.5 ±1.5	84.0 ±0.5	87.5 ±0.3	70.3 ±0.5	77.9 ±0.4	85.1 ±0.4	73.4 ±0.0	93.1 ±0.3	61.0 ±0.4	87.3 ±0.2	86.2 ±0.8
Fact. (disc, lookup)	80.98	9.17	87.0 ±2.4	75.2 ±0.8	86.1 ±1.0	95.2 ±0.5	74.9 ±0.3	74.5 ±0.7	84.9 ±0.3	72.4 ±0.2	92.5 ±0.1	57.3 ±0.5	84.9 ±0.8	86.9 ±0.6
Fact. (disc, MLP)	75.12	13.04	76.7 ±4.4	75.3 ±5.2	80.5 ±5.5	77.9 ±6.5	65.5 ±6.6	71.1 ±3.9	78.7 ±6.9	70.1 ±3.5	87.9 ±4.4	51.6 ±2.4	83.9 ±2.3	82.3 ±4.9
Fact. (disc, tree)	51.27	15.92	40.6 ±2.9	50.4 ±3.8	57.9 ±4.9	54.7 ±2.5	49.0 ±0.8	40.9 ±6.6	51.9 ±3.6	51.3 ±0.7	56.3 ±1.4	49.2 ±1.4	67.3 ±5.1	45.8 ±13.2
Mixed	83.38	5.21	92.5 ±0.5	87.6 ±1.0	**87.2** ±0.5	94.4 ±1.6	74.5 ±0.5	77.3 ±0.3	**87.3** ±0.2	74.8 ±0.0	94.5 ±0.1	53.9 ±2.1	88.2 ±0.6	88.3 ±0.2
Mixed Prior (scaled setup)														
ICMIL (Ours)	**84.17**	**4.67**	93.3 ±0.6	90.8 ±1.1	84.5 ±0.7	95.7 ±0.7	74.4 ±0.3	77.6 ±0.5	87.0 ±0.2	75.0 ±0.1	94.2 ±0.1	60.7 ±0.8	88.9 ±0.6	87.9 ±0.1

with complementary coverage from §4.2. We include JOINT with a weight of 0.70, FACTORIZED(cont, MLP) with 0.15, and FACTORIZED(disc, lookup) with 0.15. We select these weights a priori and report results for alternative weightings in Appendix B. In this way, we hope to preserve the joint prior's strong overall performance while improving on the uncorrelated witness tasks where performance seems to be largely driven by the factorized priors. We report this model as MIXED in Table 1.

MIXED retains the joint prior's lead on correlated and interaction-driven benchmarks (e.g. Pos/Neg 87.3 ± 0.2, TCGA 88.3 ± 0.2) while recovering most of the gap to the factorized priors on the uncorrelated witness tasks (Letters 94.4 ± 1.6 vs. 95.2 ± 0.5, HEPMASS 87.2 ± 0.5, the best result overall). It achieves better overall performance across all benchmarks than any of the individual priors.

4.4 SCALING YIELDS FURTHER IMPROVEMENT ON SELECTED BENCHMARKS

Building on the MIXED model from §4.3, we ask whether scaling along training duration and capacity yields a pretrained model that outperforms supervised baselines on average in the small-data regime. Specifically, we double the number of training steps to 40,000 at batch size 128, and increase the embedding dimension to 256 and the MLP hidden size to 1054. We refer to the resulting scaled model as ICMIL.

Scaling our setup further improves performance on a subset of the benchmarks. Across the twelve benchmarks, ICMIL achieves the best average AUROC (84.17) and the best average rank (4.67) of any of the models considered in Table 1. The strongest supervised baseline is ACMIL (83.81), which ICMIL leads by 0.36 points while requiring no gradient updates or per-dataset tuning. We observe that the improvement from scaling over the MIXED model is driven by Fox (+6.8), Musk2 (+3.2) and, to a smaller degree, Letters (+1.3) and Musk1 (+0.8) (Table 1). Performance on the remaining benchmarks stays within standard error of MIXED, with the exception of HEPMASS, where we observe a 2.7-point regression.

Comparing ICMIL to the baselines, we find that no single baseline is competitive across all benchmarks. The strongest aggregate baseline is MeanLogReg with an average AUROC of 82.37, followed by TabPFN-Cluster (81.21) and SVM-Summ (79.38). The two flat TabPFN baselines underperform substantially, suggesting that flattening MIL inputs into tabular form discards information that bag-aware architectures can exploit.

ABMIL produces strong results on TCGA (90.7 ± 1.6) and SMIL (85.4 ± 0.5), and is competitive on Letters (97.6 ± 0.8) and Elephant (94.1 ± 0.2). However, it underperforms on benchmarks like Adjacent Pairs (65.5 ± 1.1) and Musk2 (75.4 ± 2.8). ACMIL, which is explicitly designed to counteract attention overfitting, is markedly more robust in this regime (83.81 on average), while DSMIL (80.33) stays close to ABMIL and falls off on Adjacent Pairs (65.2 ± 1.4) and Pos/Neg (69.9 ± 1.5). This illustrates the instability caused by the small number of training bags in our benchmarks (typically around 100), which makes hyperparameter selection and early stopping highly unreliable. ICMIL avoids this failure mode because it requires no hyperparameter selection or task-specific training. This also has practical runtime consequences, as the per-split hyperparameter search of supervised baselines such as ABMIL dominates total wall-clock time, while ICMIL's runtime is competitive with the linear and kernel baselines (Figure 5). That said, ICMIL does not dominate on every benchmark. On SMIL, TCGA, and Letters, the baselines still hold a notable lead (e.g. ACMIL on SMIL by $+13.3$ and TabPFN-Cluster on Letters by $+3.5$). This suggests room for improvement, for instance through richer priors, longer pretraining, or finetuning.

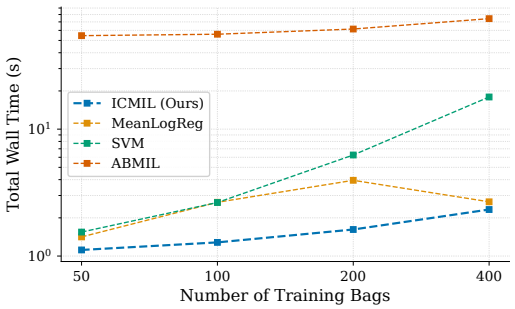

Figure 5: Total wall-clock time on the Pos/Neg benchmark for ICMIL and three supervised MIL baselines. ABMIL is substantially slower due to its per-split hyperparameter search. Once trained, ICMIL classifies all query bags in a single forward pass and is competitive with the simpler linear and kernel baselines.

5 Limitations

A few aspects of our setup leave room for further investigation. Finetuning ICMIL on the training partitions of the benchmark datasets to unlock further performance gains remains a promising direction. Furthermore, our benchmarks reduce the feature dimension to 25 via PCA, leaving higher-dimensional foundation-model embeddings as future work. This gap should be addressable by training on higher-dimensional synthetic data and scaling model capacity and training duration (Kolberg et al., 2025). At the same time, our scaling experiment suggests that gains from added capacity and training time are not fully uniform across benchmarks and hint at interactions between prior mixture, model size and training duration that warrant closer study.

6 Conclusion

We introduced *In-Context Multiple Instance Learning* (ICMIL), a framework for solving MIL problems via in-context learning over bag-structured data. To support this, we proposed an architecture that addresses the scalability, task-dependent compression, and permutation-invariance challenges that arise with hierarchical set inputs, and designed a family of synthetic priors that encode complementary inductive biases over MIL tasks. A model trained on a mixture of these priors achieves the best average AUROC and the best average rank across twelve MIL benchmarks in the low-label regime, ahead of both classical and recent supervised MIL methods.

Beyond the scaling directions outlined in the previous section, several orthogonal avenues are likely to yield further gains. On the prior side, our results show that the choice of synthetic data generator matters considerably and that we can fruitfully combine priors that encode complementary inductive biases. Therefore, designing priors aligned with specific application domains (Sextro et al., 2026), such as spatial priors for computational pathology or sequence-aware priors for time-resolved tasks (Hoo et al., 2025), is another promising direction.

Furthermore, inspired by recent work on tabular PFNs (Hollmann et al., 2025; Qu et al., 2025), post-training on real-world MIL corpora could further close the synthetic-to-real gap (Garg et al., 2025), and richer architectural variants remain to be explored. We hope ICMIL serves as a starting point for bringing the benefits of in-context learning to bag-structured prediction problems, and more broadly to settings where labels are scarce and supervision is weak.

AI USE STATEMENT

In this work, we used generative AI tools to implement methods and to support qualitative and thematic data analysis. We have not used generative AI tools to generate synthetic data sets, help develop theoretical models or conceptual frameworks, propose or refine hypotheses, design or provide feedback on research methodology or experiments, clean and reformat datasets, or interpret results. Formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, and assisting with translation are not applicable to this work. Additionally, we used generative AI tools to create or edit software code, create or modify scientific figures or images, and edit the paper to improve readability. We have reviewed all AI-assisted work. AI-generated code was verified and tested for correctness by two authors. AI-generated figures were checked against the underlying data and all final interpretations are the author's own. We take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We release code for data generation, training, and evaluation of the ICMIL models at `https://anonymous.4open.science/r/ICMIL-62E2`, together with the checkpoints of all models trained for this manuscript, the exact data splits used for benchmarking, and the code and configurations for training and evaluating all baseline methods.

REFERENCES

Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. Support vector machines for multiple-instance learning. In *Proceedings of the 16th International Conference on Neural Information Processing Systems*, NIPS'02, pp. 577–584, 2002.

Pierre Baldi, Kyle Cranmer, Taylor Faucett, Peter Sadowski, and Daniel Whiteson. Parameterized neural networks for high-energy physics. *The European Physical Journal C*, 76(5):235, 2016.

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.

Marc-André Carbonneau, Veronika Cheplygina, Eric Granger, and Ghyslain Gagnon. Multiple instance learning: A survey of problem characteristics and applications. *Pattern Recognition*, 77: 329–353, 2018.

Francisco M Castro-Macías, Francisco J Sáez-Maldonado, Pablo Morales-Álvarez, and Rafael Molina. Torchmil: A pytorch-based library for deep multiple instance learning. *Neurocomputing*, 680:133286, 2026.

Muhammad Dawood, Kim Branson, Sabine Tejpar, Nasir Rajpoot, and Fayyaz ul Amir Afsar Minhas. Confounding factors and biases abound when predicting molecular biomarkers from histological images. *Nature Biomedical Engineering*, pp. 1–15, 2026.

Aaron Defazio, Xingyu Yang, Harsh Mehta, Konstantin Mishchenko, Ahmed Khaled, and Ashok Cutkosky. The road less scheduled. *arXiv arXiv:2405.15682*, 2024.

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.

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, Drew F. K. Williamson, Harry Robertson, Bowen Chen, Cristina Almagro-Pérez, Paul Doucet, Sharifa Sahai, Chengkuan Chen, Christina S. Chen, Daisuke Komura, Akihiro Kawabe, Mieko Ochi, Shinya Sato, Tomoyuki Yokose, Yohei Miyagi, Shumpei Ishikawa, Georg Gerber, Tingying Peng, Long Phi Le, and Faisal Mahmood. A multimodal whole-slide foundation model for pathology. *Nature Medicine*, 31(11): 3749–3761, 2025.

David Jacob Drexlin, Jonas Dippel, Julius Hense, Niklas Prenißl, Grégoire Montavon, Frederick Klauschen, and Klaus-Robert Müller. Medi: Metadata-guided diffusion models for mitigating biases in tumor classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 379–388, 2025.

Babak Ehteshami Bejnordi, Mitko Veta, Paul Diest, Bram Ginneken, Nico Karssemeijer, Geert Litjens, Jeroen van der Laak, Meyke Hermsen, Quirine Manson, Maschenka Balkenhol, Oscar Geessink, Nikolaos Stathonikos, Marcory van Dijk, Peter Bult, Francisco Beca, Andrew Beck, Dayong Wang, Aditya Khosla, Rishab Gargeya, and Rui Venâncio. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *JAMA*, 318:2199–2210, 2017.

Adam E. Flanders, Luciano M. Prevedello, George Shih, Safwan S. Halabi, Jayashree Kalpathy-Cramer, Robyn Ball, John T. Mongan, Anouk Stein, Felipe C. Kitamura, Matthew P. Lungren, Gagandeep Choudhary, Lesley Cala, Luiz Coelho, Monique Mogensen, Fanny Morón, Elka Miller, Ichiro Ikuta, Vahe Zohrabian, Olivia McDonnell, Christie Lincoln, Lubdha Shah, David Joyner, Amit Agarwal, Ryan K. Lee, and Jaya Nath. Construction of a machine learning dataset through collaboration: The RSNA 2019 brain CT hemorrhage challenge. *Radiology: Artificial Intelligence*, 2(3):e190211, 2020.

James Foulds and Eibe Frank. A review of multi-instance learning assumptions. *The Knowledge Engineering Review*, 25(1):1–25, 2010.

Peter W. Frey and David J. Slate. Letter recognition using Holland-style adaptive classifiers. *Machine Learning*, 6(2):161–182, 1991.

Anurag Garg, Muhammad Ali, Noah Hollmann, Lennart Purucker, Samuel Müller, and Frank Hutter. Real-TabPFN: Improving Tabular Foundation Models via Continued Pre-training With Real-World Data. In *Proceedings of the 1st ICML Workshop on Foundation Models for Structured Data*, 2025.

Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, 1992.

Julius Hense, Mina Jamshidi Idaji, Oliver Eberle, Thomas Schnake, Jonas Dippel, Laure Ciernik, Oliver Buchstab, Andreas Mock, Frederick Klauschen, and Klaus Robert Muller. xMIL: Insightful explanations for multiple instance learning in histopathology. In *Advances in Neural Information Processing Systems*, pp. 8300–8328, 2024.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *The Eleventh International Conference on Learning Representations*, 2023.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

Shi Bin Hoo, Samuel Müller, David Salinas, and Frank Hutter. From tables to time: Extending TabPFN-v2 to time series forecasting. *arXiv arXiv:2501.02945*, 2025.

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

Mina Jamshidi Idaji, Julius Hense, Tom Neuhäuser, Augustin Krause, Yanqing Luo, Oliver Eberle, Thomas Schnake, Laure Ciernik, Farnoush Rezaei Jafari, Reza Vahidimajd, Jonas Dippel, Christoph Walz, Frederick Klauschen, Andreas Mock, and Klaus-Robert Müller. Beyond attention heatmaps: How to get better explanations for multiple instance learning models in histopathology. *Medical Image Analysis*, pp. 104148, 2026.

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, 2018.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4651–4664, 2021.

Kyeonghun Jeong, Jinwook Choi, and Kwangsoo Kim. scMILD: Single-cell multiple instance learning for sample classification and associated subpopulation discovery. *iScience*, 29(4), 2026.

Christopher Kolberg, Katharina Eggensperger, and Nico Pfeifer. TabPFN-Wide: Continued pretraining for extreme feature counts. In *EurIPS 2025 Workshop: AI for Tabular Data*, 2025.

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. *arXiv preprint arXiv:2507.17845*, 2025.

Nikolaus Kopp, Alexander Fuchs, Markus Feuerstein, Phillip Paller, and Franz Pernkopf. Utilizing TabPFN for multi-instance data with scarce labels. In *EurIPS 2025 Workshop: AI for Tabular Data*, 2025.

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 (CVPR)*, pp. 14318–14328, 2021.

Ming Y. Lu, Bowen Chen, Andrew Zhang, Drew F. K. Williamson, Richard J. Chen, Tong Ding, Long Phi Le, Yung-Sung Chuang, and Faisal Mahmood. Visual language pretrained multiple instance zero-shot transfer for histopathology images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19764–19775, 2023.

Oded Maron and Tomás Lozano-Pérez. A framework for multiple-instance learning. In *Advances in Neural Information Processing Systems*, volume 10, pp. 570–576, 1997.

Pablo Meseguer, Rocío del Amor, and Valery Naranjo. MIL-Adapter: Coupling multiple instance learning and vision-language adapters for few-shot slide-level classification. *Medical Image Analysis*, 110:103964, 2026.

Alexander Möllers, Julius Hense, Florian Schulz, Timo Milbich, Maximilian Alber, and Lukas Ruff. Mind the gap: Continuous magnification sampling for pathology foundation models. *arXiv preprint arXiv:2601.02198*, 2026.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference. In *International Conference on Learning Representations*, pp. 81861–81875, 2022.

Samuel Müller, Matthias Feurer, Noah Hollmann, and Frank Hutter. PFNs4BO: In-context learning for Bayesian optimization. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 25444–25470, 2023.

Mitsuyoshi Otsu, Sho Nakamura, Shigeru Tomita, Tomoyuki Suhama, Yasunobu Shimazaki, and Katsuya Nishimura. Multi-resolution domain adaptation via multiple instance learning for improving the recognition accuracy of japanese oak wilt in low-resolution satellite imagery. In *SPIE Future Sensing Technologies 2023*, volume 12327, pp. 1232716, 2023.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: an imperative style, high-performance deep learning library. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pp. 8026–8037, 2019.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12 (85):2825–2830, 2011.

Alexander Pfefferle, Johannes Hog, Lennart Purucker, and Frank Hutter. nanoTabPFN: A Lightweight and Educational Reimplementation of TabPFN. In *EurIPS 2025 Workshop: AI for Tabular Data*, 2025.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICL: A tabular foundation model for in-context learning on large data. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 50817–50847, 2025.

Jake Robertson, Arik Reuter, Siyuan Guo, Noah Hollmann, Frank Hutter, and Bernhard Schölkopf. Do-PFN: In-Context Learning for Causal Effect Estimation. In *Advances in Neural Information Processing Systems*, volume 38, pp. 174811–174848, 2025.

Marvin Sextro, Weronika Kłos, and Gabriel Dernbach. MapPFN: Learning causal perturbation maps in context. In *ICLR 2026 Workshop on Generative AI in Genomics (Gen^2)*, 2026.

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 *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 54219–54238, 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. In *Advances in Neural Information Processing Systems*, volume 34, pp. 2136–2147, 2021.

Ryan Soklaski, Justin Goodwin, Olivia Brown, Michael Yee, and Jason Matterer. Tools and practices for responsible AI engineering. *arXiv preprint arXiv:2201.05647*, 2022.

Muhammad Waqas, Syed Umaid Ahmed, Muhammad Atif Tahir, Jia Wu, and Rizwan Qureshi. Exploring multiple instance learning (MIL): A brief survey. *Expert Systems with Applications*, 250:123893, 2024.

Hanwen Xu, Naoto Usuyama, Jaspreet Bagga, Sheng Zhang, Rajesh Rao, Tristan Naumann, Cliff Wong, Zelalem Gero, Javier González, Yu Gu, Yanbo Xu, Mu Wei, Wenhui Wang, Shuming Ma, Furu Wei, Jianwei Yang, Chunyuan Li, Jianfeng Gao, Jaylen Rosemon, Tucker Bower, Soohee Lee, Roshanthi Weerasinghe, Bill J. Wright, Ari Robicsek, Brian Piening, Carlo Bifulco, Sheng Wang, and Hoifung Poon. A whole-slide foundation model for digital pathology from real-world data. *Nature*, 630(8015):181–188, 2024.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander Smola. Deep Sets. In *Advances in Neural Information Processing Systems*, volume 30, pp. 3394–3404, 2017.

Yunlong Zhang, Honglin Li, Yuxuan Sun, Sunyi Zheng, Chenglu Zhu, and Lin Yang. Attention-challenging multiple instance learning for whole slide image classification. In *Computer Vision – ECCV 2024*, volume 15111 of *Lecture Notes in Computer Science*, pp. 125–143, 2024.

Yu Zhao, Fan Yang, Yuqi Fang, Hailing Liu, Niyun Zhou, Jun Zhang, Jiarui Sun, Sen Yang, Bjoern Menze, Xinjuan Fan, et al. Predicting lymph node metastasis using histopathological images based on multiple instance learning with deep graph convolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4837–4846, 2020.

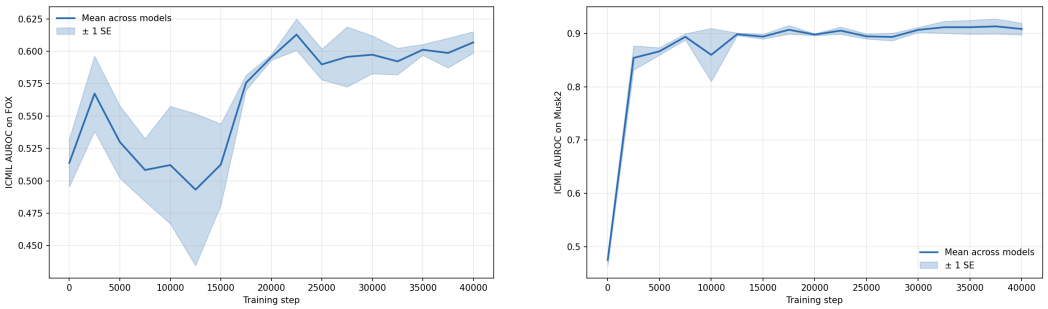

Figure 6: Learning curves on Fox (left) and Musk2 (right), mean ± SE across three runs. Performance keeps improving with longer training, motivating the extended schedule used for ICMIL.

APPENDIX

A TRAINING DETAILS

Synthetic datasets. Synthetic datasets are sampled from a fixed pre-generated pool. Across 20,000 steps with batch size 128, the model is exposed to approximately 2.56M independently sampled synthetic datasets, doubling to 5.12M for the scaled ICMIL model. No synthetic dataset is seen twice during training. Each batch is homogeneous in the sense that all 128 datasets are drawn from the same prior configuration. Across all priors, we apply TabICL's regression-to-classification ("reg2cls") transform (Qu et al., 2025) to convert the SCM's continuous output into a discrete bag label.

Curriculum. To speed up training, we apply a three-stage curriculum over bag size and instance class count. In the first 500 steps, bag sizes are restricted to $[2, 8]$ with at most 6 instance classes. From step 500 to 7,500, bag sizes grow to $[4, 15]$ with up to 12 instance classes. For the remainder of training (steps 7,500–20,000), bag sizes reach $[6, 20]$ with up to 20 instance classes.

Architecture. The reduced model used for the prior ablations (§4.2, §4.3) has T=6 iterations, 4 attention heads, embedding dimension E=128, MLP hidden size 512, and feature group size s=1. The scaled ICMIL model (§4.4) keeps the same number of iterations and attention heads, increases E to 256 and the MLP hidden size to 1054. All other architectural settings (column-row attention, chunked instance aggregation, label embedding) are identical.

Optimization. We optimize the negative log-likelihood on the query bags with the schedule-free AdamW optimizer (Defazio et al., 2024). For the reduced setup used in the prior ablations (§4.2, §4.3) we use learning rate 1×10^{-3} with no warmup. For the scaled ICMIL model (§4.4) we use learning rate 5×10^{-4} with 2,500 warmup steps to stabilize training. We train ICMIL for 40,000 steps and show exemplary learning curves for the Fox and Musk2 benchmarks in Figure 6. All models are trained on a single NVIDIA A100 GPU.

A.1 IMPLEMENTATION

We use `PyTorch` (Paszke et al., 2019) to implement our experiments and configure them via `hydra-zen` (Soklaski et al., 2022). Our synthetic priors build on the MLP-SCM and Tree-SCM generators from `TabICL` (Qu et al., 2025), including its hyperparameter samplers and its regression-to-classification (`reg2cls`) transform. The ICMIL Perceiver-style architecture and training loop are implemented on top of the `nanoTabPFN` reference implementation (Pfefferle et al., 2025), and we optimize with schedule-free AdamW from `schedule-free` (Defazio et al., 2024). For baselines, we use `ABMIL` (Ilse et al., 2018) as re-implemented in `MIL-Lab` (Shao et al., 2025), and the `TabPFN-v2` model from `TabPFN` (Hollmann et al., 2025), all classical baselines (logistic regression, SVM, PCA, stratified cross-validation) use `scikit-learn` (Pedregosa et al., 2011). Benchmark loaders for RSNA-ICH are taken from `torchmil` (Castro-Macías et al., 2026).

We run our experiments on a high-performance cluster. ICMIL pretraining runs are executed on a single NVIDIA A100 with 80 GB of VRAM. Training the reduced models used in the prior ablations (§4.2, §4.3) takes approximately 12 h per run, while the scaled ICMIL model (§4.4) trains for roughly 24 h. Evaluation runs for the supervised baselines (ABMIL, DSMIL, ACMIL, MeanLogReg, SVM-Summ, TabPFN-Concat, TabPFN-Subsample) are executed on less powerful GPUs of the same cluster.

B Ablations

We isolate the contribution of the components that our architecture introduces for bag-structured inputs (§3.1), namely the iterative aggregation of instances into bag tokens and the attention between bags, and additionally probe the embedding dimension and the weighting of the prior mixture. All ablations use the reduced setup of Appendix A and train on the MIXED prior for 20,000 steps with the same seed. We vary one component at a time and report mean AUROC across the twelve benchmarks. Rows marked *(paper)* correspond to the configuration used in the main text.

Iterative aggregation and bag-token updates. At $T{=}1$ the model cannot refine the bag token after the labels are seen and underperforms drastically (65.5, Table 2). Iterating the aggregation is therefore necessary, with performance rising steeply through $T{=}2$ (79.0) and stabilizing around $T{=}4$ (83.3).

Inter-bag attention. In-context learning is impossible without information exchange between bag representations. Disabling inter-bag attention leaves each bag token with access only to its own instances, and performance collapses to chance (45.8 vs. 83.3, Table 2).

Embedding dimension. Embeddings are required for the attention computation, but their size has little effect between $E{=}48$ and $E{=}128$ (82.4–83.3, Table 2).

Table 2: Ablation of the architectural components, reporting mean AUROC (%) across the twelve benchmarks. Each block varies one component while holding the remainder at the reduced setup.

Ablation	Mean AUROC
Iterations	
$T{=}1$	65.5
$T{=}2$	79.0
$T{=}4$	83.3
$T{=}6$ (paper)	83.3
Inter-bag attention	
Disabled	45.8
Enabled (paper)	83.3
Embedding dimension	
$E{=}48$	83.1
$E{=}64$	82.4
$E{=}96$	83.3
$E{=}128$ (paper)	83.3

Prior mixture weights. The mixture weights are fixed a priori, giving most of the weight to the JOINT prior for its strong overall performance and the remainder to the two factorized priors that cover the uncorrelated witness benchmarks. As a post-hoc check of their sensitivity, we train one model per weighting and report the results in Table 3. Mean AUROC varies by at most 1.5 points across the three weightings and never collapses, so the weights are not a sensitive hyperparameter. The more uniform $0.60/0.20/0.20$ weighting is marginally ahead in this reduced setup, and adopting it at scale might improve ICMIL further.

Table 3: Sensitivity to the prior mixture weights, given as JOINT / FACTORIZED(cont, MLP) / FACTORIZED(disc, lookup) and reporting mean AUROC (%) across the twelve benchmarks.

Mixture weights	Mean AUROC
0.60/0.20/0.20	83.5
0.70/0.15/0.15 (paper)	83.3
0.80/0.10/0.10	82.0

C PREDEFINED VERSUS IN-CONTEXT AGGREGATION

TabPFN-Cluster (Kopp et al., 2025) and ICMIL both build on a pretrained in-context learner, but they differ fundamentally in how instances are aggregated into a bag representation. The aggregation strategy of TabPFN-Cluster is predefined and independent of the task labels. ICMIL instead learns to aggregate instances according to the task at hand, using the labels of the context bags to decide which instance-level information to preserve.

To illustrate the difference, we construct a simple task on which TabPFN-Cluster fails and ICMIL succeeds. We generate 150 training and 60 test bags of 50 instances (25 features) from two Gaussians (means ± 2), clip feature 24 at ± 3.5, and inject witness instances beyond ± 4. A bag is positive only if it contains both witnesses. We run TabPFN-Cluster with $K=2$ clusters, matching the true number, and ICMIL out of the box (Table 4). The predefined per-cluster mean aggregation of TabPFN-Cluster averages away the witness values that define the labels. This predefined aggregation makes it fail on this task, while ICMIL can flexibly approximate the labeling function in-context and succeed.

Table 4: Predefined versus in-context aggregation on the synthetic two-witness task (150 training bags, 60 test bags). AUROC (%) on the test bags.

Method	Aggregation	Uses bag labels for Aggregation	AUROC
TabPFN-Cluster	predefined	no	54.0
ICMIL	learned in-context	yes	**88.0**

D BENCHMARK DETAILS

This appendix gives the full description of the twelve MIL benchmarks summarized in Section 4.2. The paragraphs that follow describe each benchmark. Each benchmark comprises a fixed number of bags: Musk1 has 92 bags and Musk2 has 102, Elephant, Fox, and Tiger contain 200 bags each, and the remaining benchmarks (SMIL, Pos/Neg, Adjacent Pairs, Letters, HEPMASS, RSNA-ICH, and TCGA) are subsampled to 100 bags.

MNIST SMIL. Following the xMIL benchmark (Hense et al., 2024), we draw $N_{\text{bags}}=100$ bags whose size $K \sim \mathcal{U}\{10, 20\}$ is resampled per draw. For every bag we uniformly activate a random subset of the digit classes $\{0, \ldots, 9\}$ and sample instances with replacement from the active set, empty activations are rejected. Each instance is represented by a 512-dim ResNet-18 embedding reduced to $d=25$ dimensions with PCA fit per draw on its training bags. The witness digit is sampled uniformly per draw from $\{0, \ldots, 9\}$, and positive bags contain exactly $w \sim \mathcal{U}\{1, 2\}$ copies of that digit while negative bags contain none. We generate 5 independent draws, each with a fresh label rule and freshly sampled bags, partitioned 90:10 into train and test.

MNIST PosNeg. Identical sampling and features to SMIL, but labels are determined by a *counting* rule: for each draw a pair of disjoint size-3 digit triples (P, N) is sampled from a random permutation of $\{0, \ldots, 9\}$ (e.g. $P=\{4, 6, 8\}$, $N=\{5, 7, 9\}$), and a bag is positive iff $|\{i : x_i \in P\}| > |\{i : x_i \in N\}|$. Unlike SMIL, a single witness is insufficient, the label depends on the balance of positive and negative evidence across the bag.

MNIST AdjPairs. Identical sampling and features to SMIL. For each dataset we sample a set of adjacent digit pairs (for example $\{(1, 2), (3, 4)\}$) and label a bag positive if any sampled adjacent pair is jointly present in the bag. Bag sizes are sampled in $[10, 20]$ and the class balance is sampled in $[0.2, 0.5]$.

UCI Musk1. The classical natural-MIL benchmark (Dietterich et al., 1997): 92 molecules (bags) whose conformations form the instances, with 166 physicochemical features per conformation. We keep the native bag structure, zero-pad all bags to the dataset-wide maximum conformation count, and reduce features to d=25 with PCA. The task is binary musk vs. non-musk classification. We evaluate with 5-fold stratified cross-validation on molecules so every bag tests exactly once.

UCI Musk2. Same sampling, feature representation, and protocol as Musk1, applied to the larger Musk2 collection of 102 molecules (39 musk, 63 non-musk) with 1 to 1,044 conformations per molecule. Bags are zero-padded to the dataset-wide maximum conformation count, features are reduced to d=25 with PCA, and we evaluate with 5-fold stratified cross-validation on molecules.

UCI Letters. We adapt the UCI Letter Recognition dataset (Frey & Slate, 1991) (20,000 instances, 16 features, 26 classes) to MIL by constructing bags of size K=10 with the *witness-rate* recipe of Carbonneau et al. (2018): positive bags contain $\max(1, \lfloor 0.2K \rfloor)$=2 witnesses drawn from the target class (letter `A`) and 8 non-target background instances, negative bags contain only non-target instances. We build N_{bags}=100 bags with balanced 50:50 positive/negative fractions, no PCA (since d=16≤25), and 5 stratified 90:10 splits.

UCI HEPMASS. We use the HEPMASS 1000 variant (Baldi et al., 2016) (27 features, binary signal-vs-background). Bags are constructed with the same witness-rate recipe as Letters, with the signal class as the witness concept: 100 bags of size 10, balanced classes, 2 witnesses per positive bag. Features are reduced to d=25 with PCA. Splits: 5 stratified 90:10.

Elephant/Fox/Tiger. Three natural-image MIL datasets of 200 images each, evaluated separately per animal class. Each image is a bag, the segmented regions are the instances, and each region is described by 230 color and texture features. A bag is positive if at least one region contains the target animal. Bag size varies across images and is zero-padded to the dataset-wide maximum. We evaluate with 5-fold stratified cross-validation on bags so every image tests exactly once.

RSNA-ICH. An intracranial-hemorrhage CT benchmark from the `torchmil` (Castro-Macías et al., 2026) release of the RSNA 2019 Intracranial Hemorrhage Detection dataset (Flanders et al., 2020). Each CT scan is a bag and its axial slices are instances, embedded with a pretrained ResNet-50 backbone (2,048-dim). A bag is positive if at least one slice contains any of the six hemorrhage subtypes. Rather than padding to the global maximum slice count, we follow the multi-draw protocol used for our other torchmil benchmarks: 50 independent draws are sampled from the bundled `splits.csv` train/test partition. For each draw an instance count $n \sim \mathcal{U}\{15, 30\}$ is drawn uniformly, 100 training bags with $\geq n$ slices are sampled without replacement from the train partition and trimmed to n slices each, while every qualifying test bag is kept and trimmed to n. PCA reduces features to d=25, refit per draw on that draw's training bags.

TCGA LUAD vs. LUSC. A binary lung adenocarcinoma (LUAD) vs. lung squamous cell carcinoma (LUSC) task built from TCGA whole-slide embeddings. Each patient contributes one bag, sampled from a single slide with one tile per patient cohort to prevent slide-level leakage, instances are 1,536-dim UNI2 patch features. Class balance is intentionally imbalanced at 80 LUAD / 20 LUSC patients, and each bag is constructed by sampling K=10 tiles uniformly without replacement from the chosen slide (with replacement when a slide has fewer than K tiles). Features are reduced to d=25 with PCA fit per split on the training bags. We use 5 stratified 90:10 train/test splits drawn over the patient pool.

Common interface and preprocessing. All tasks expose the same interface, yielding per split a tuple $(\mathcal{X}_{\text{train}}, y_{\text{train}}, \mathcal{X}_{\text{test}}, y_{\text{test}})$ with bag tensors of shape (N_{bags}, K, d) and bag labels of shape $(N_{\text{bags}},)$, where K is the number of instances per bag (zero-padded when variable) and d is the feature dimensionality. Musk1, Musk2, and Andrews Fox/Tiger/Elephant use 5-fold stratified cross-validation on bags so every bag tests exactly once; RSNA-ICH uses the 50-draw protocol described above, the remaining tasks (MNIST SMIL/PosNeg/AdjPairs, Letters, HEPMASS, TCGA) use 5 stratified 90:10 train/test splits. Variable-size bags are zero-padded to the dataset-wide maximum. Per-instance features are reduced to d=25 with PCA fit per split on the training bags, with the sole exception of Letters where the raw d=16 ≤ 25 leaves PCA inactive.