

Reducing Variability of Multiple Instance Learning Methods for Digital Pathology

Ali Mammadov^{1,2}, Loïc Le Folgoc¹, Guillaume Hocquet², and Pietro Gori¹

¹ LTCI, Télécom Paris, Institut Polytechnique de Paris, France

² Paris Saint-Joseph Hospital, France
`ali.mammadov@ip-paris.fr`

Abstract. Digital pathology has revolutionized the field by enabling the digitization of tissue samples into whole slide images (WSIs). However, the high resolution and large size of WSIs present significant challenges when it comes to applying Deep Learning models. As a solution, WSIs are often divided into smaller patches with a global label (*i.e.*, *diagnostic*) per slide, instead of a (too) costly pixel-wise annotation. By treating each slide as a bag of patches, Multiple Instance Learning (MIL) methods have emerged as a suitable solution for WSI classification. A major drawback of MIL methods is their high variability in performance across different runs, which can reach up to 10-15 AUC points on the test set, making it difficult to compare different MIL methods reliably. This variability mainly comes from three factors: i) weight initialization, ii) batch (shuffling) ordering, iii) and learning rate. To address that, we introduce a *Multi-Fidelity, Model Fusion* strategy for MIL methods. We first train multiple models for a few epochs and average the most stable and promising ones based on validation scores. This approach can be applied to any existing MIL model to reduce performance variability. It also simplifies hyperparameter tuning and improves reproducibility while maintaining computational efficiency. We extensively validate our approach on WSI classification tasks using 2 different datasets, 3 initialization strategies and 5 MIL methods, for a total of more than 2000 experiments.

Keywords: Multiple Instance Learning · Variadion Reduction · Whole Slide Image Classification.

1 Introduction

Recent advances in Digital Pathology have made automated disease diagnosis using Deep Learning (DL) very popular. In these applications, a pathological slide is converted into a Whole Slide Image (WSI) in a pyramidal format, where each layer represents a different magnification. Because these images are very large, conventional DL methods are not practical. Instead, the Multiple-Instance Learning (MIL) framework is used for WSI classification.

In MIL, each slide is divided into small, non-overlapping patches using a sliding-window approach. These patches form a "bag" of instances. Unlike standard

supervised learning, only the slide-level (bag-level) labels are available. This approach eliminates the need for expensive manual pixel-level annotations. A bag is labeled as negative if all patches are negative, and as positive if at least one patch is positive, which fits well with the fact that tumor regions often cover only part of the slide. First, semantically rich features are extracted from these patches using pre-trained encoders (from ImageNet or in a self-supervised way) or foundation models. After feature extraction, either a patch-level classifier is trained and its scores are aggregated, or an aggregator is trained to create a slide-level representation that is used for the final prediction.

One ongoing challenge in deep learning is reproducibility. The performance of models often varies between runs, making it hard to compare models and tune hyperparameters. This problem appears in many DL fields [11, 22], such as natural language processing [1], generative adversarial networks [20], deep reinforcement learning [12], and image recognition [2].

This issue also affects digital pathology for WSI classification with MIL. Recent works [5, 6, 14, 16, 18, 21, 24, 27, 29, 30] report high standard deviations between different runs of the same model. Furthermore, in all these works, the difference between the best performing method (usually the proposed one) and the second best performing one is very small, usually around 1-2 AUC, and much smaller than the variability of each method. This represents a significant problem for assessing whether a method actually outperforms the other methods or whether the reported differences are merely due to chance (also called "cherry picking"). In our experiments on two datasets with several MIL methods, we observed differences of up to 10–15 AUC points between runs. We found that this variation is mainly due to three factors: model initialization, the order of data presentation during training, and the way model weights are updated. We simplify these factors as: the initialization seed, shuffle seed, and learning rate. Finding the perfect combination of these parameters is computationally expensive. Figure 1-Top shows the test AUC scores for 5 MIL methods on two different datasets (BRACS [3] and Camelyon [9]). For each MIL method, we tried 12 different combinations of shuffle and initialization seeds (black points), tuning the learning rate on the validation set.

Related Works. One of the earliest approaches to build more robust models is ensemble modeling. The idea is straightforward: instead of training a single model, multiple models are trained, and during inference, their predictions are averaged. Another simple solution is to just pick the model with the best Validation score. However, these approaches have a major drawback: high computational cost, as they require fully training several models. To address this, Wortsman et al. [25] introduced Model Soups (or Model Averaging), a method that averages the weights of multiple models, trained from the same parameter initialization, allowing a single forward pass while maintaining the benefits of ensembling. This approach improves performance and robustness without increasing inference costs. However, it may arise another issue when averaging weights with opposite signs, as they can cancel each other out, leading to inac-

tive neurons. *TIES-Merging* [26] solves this by averaging only the weights with matching signs and setting small conflicting values to zero.

Contributions. Inspired by model averaging [25, 26] and multi-fidelity hyperparameter optimization [8, 10], we propose a new method to reduce performance variability in MIL-based whole slide image classification. In our work, we use the idea of *Model Soups* and *TIES-Merging*, where we average the weights of the best models. For choosing the best models we follow the idea of multi-fidelity hyperparameter optimization. Instead of fully training each model, we train it for a few epochs to quickly estimate its performance. By combining these two approaches, our goal is to smooth out differences caused by random initialization, data shuffling, and training updates, thus reducing performance variability. In our method, we first train M models for K epochs (usually $M = 10$ and $K = 5$), then select the top T models (usually $T = 3$) based on early validation AUC scores and eventually average their weights. We show that this simple procedure reduces the performance variability thus increasing reproducibility and trustworthiness. Meanwhile, our method keeps the computational burden at a reasonable rate, increasing the total number of training epochs of only $K * M = 50$, which usually represents 25% or 50% of the total number of training epochs. Additionally, this method is generic and can be applied to any existing MIL method.

2 Method

MIL Formulation. Each slide is modeled as a labeled bag containing unlabeled patches. Consider dataset $\mathcal{S}$ containing N slides, represented as $\mathcal{S} = \{X_1, X_2, \dots, X_N\}$ associated with labels denoted by $Y = \{y_1, y_2, \dots, y_N\}$. Each individual slide X_i is composed of a collection of patches, denoted as $X_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,P_i}\}$, extracted exclusively from the foreground tissue regions of the slide where the value of P_i varies based on the size of the slide. Note that there are no labels for patches ($x_{i,j}$), only slide-level labels are provided, therefore this is considered as a weak supervision. The WSI classification pipeline is structured into multiple phases. The initial phase is pre-training, during which the backbone model $\mathbf{f}_\phi$ is pre-trained on the patches of the training slides with the given self-supervised learning method. Then, for every slide i , features are extracted from patches j , assembled within each bag, and used as input for the MIL aggregator network $\mathbf{g}_{\theta_g}$. This network aggregates the features to generate a bag representation of the slide i , which is then forwarded to the classifier $\mathbf{c}_{\theta_c}$ for predicting the class based on the task. It can be formulated as:

$$h_{i,j} = \mathbf{f}_\phi(x_{i,j}); \quad H_i = \mathbf{g}_{\theta_g}(h_{i,1}, h_{i,2}, \dots, h_{i,P}); \quad C_i = \mathbf{c}_{\theta_c}(H_i) \quad (1)$$

In this work, we ignore the variability of the encoder $\mathbf{f}_\phi$, considering it already pre-trained and frozen, focusing only on the other two networks, namely $\mathbf{g}_{\theta_g}$ and $\mathbf{c}_{\theta_c}$, which present a variability that depends on their gradient-based optimization process. The values of the final parameters $\theta = \{\theta_g, \theta_c\}$ (i.e., at the end of the training) depend on the initialization and on the optimization process,

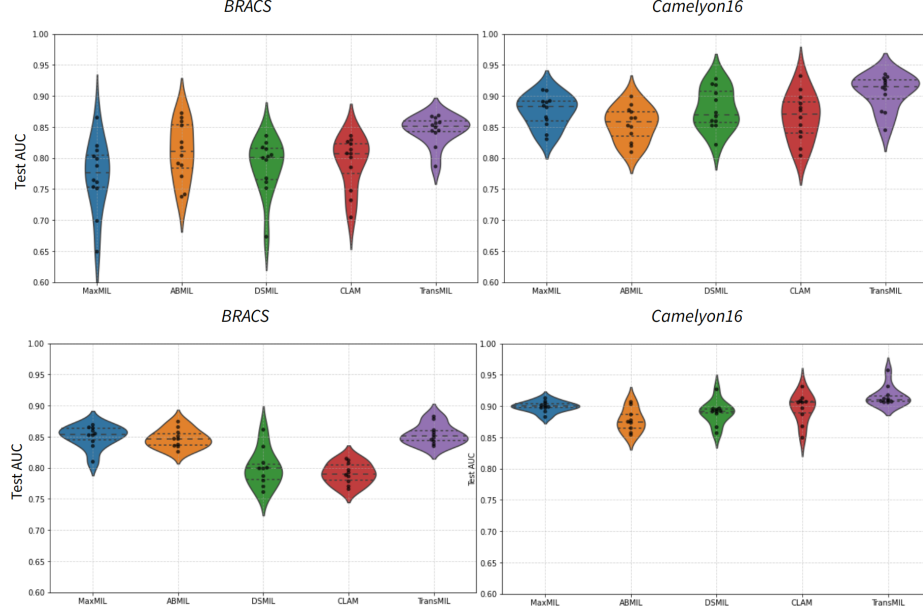


Fig. 1. Top: Violin plots on 2 different datasets. Each dot represents the test AUC of a model trained with a random shuffle and initialization seed and with learning rate tuned on the validation set. **Bottom:** we apply the proposed method (using Soup for MaxMIL and ABMIL and Ties for DSMIL, CLAM and TransMIL) using the *same* initialization seeds as in the Top figure and $M = 10$, $K = 5$ and $T = 3$. The proposed method clearly reduces variability between different runs while preserving the top performance.

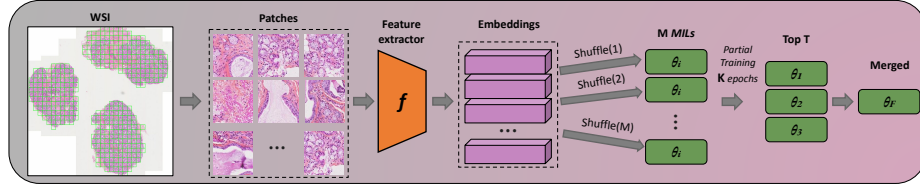


Fig. 2. Pipeline of the proposed Multi-Fidelity Model Averaging method for Whole Slide Image Classification.

whose most important hyper-parameters are: initialization seed, shuffling seed and learning rate. By changing one of them, results may drastically vary.

Model Overview. In our work, we propose combining Multi-Fidelity with Model Averaging to mitigate variability and increase robustness (see Fig.2). Each slide is first cut into patches, whose features are extracted using f_ϕ . Then, M **identical** models are initialized with the same initialization seed, and each model is trained with a different and random shuffling seed for a small number of K

epochs. The learning rate can be tuned using the validation set or chosen based on prior knowledge from existing literature, without further tuning. After the partial training, the top T (e.g., 3) models are selected based on their validation AUC scores. Next, the weights of these selected models are aggregated, and the resulting, combined model is fully trained.

Model aggregation. We propose using two simple merging methods: uniform *SOUP* [25] and *TIES*-Merging [26]. Let $\theta_1, \theta_2, \dots, \theta_M$ be the weights of the M partially trained models. Uniform SOUP is simply defined as the average across models: $\theta_{\text{uniform}} = \frac{1}{M} \sum_{i=1}^M \theta_i$. TIES-Merging works by first calculating the difference between each partially trained model and the initial model. It then trims these differences to keep only the most important changes, discarding the small ones. Next, for each parameter, it chooses the dominant sign among the models and it averages only the values that agree with the dominant sign.

3 Results and Discussion

3.1 Implementation and Data information

Datasets and Data Splits. We conduct experiments on two datasets. Camelyon16 [9] is a 2-class dataset for the detection of metastases in breast cancer. It comprises 400 slides, with 239 normal tissue slides and 160 tumor slides. BReAst Carcinoma Subtyping (BRACS) [3] is a 3-class imbalanced dataset for breast carcinoma subtyping, containing 547 whole-slide images (WSIs): 265 benign tumor cases, 89 atypical tumor cases, and 193 malignant tumor cases. For both datasets, we use the official data splits.

Evaluation Metric. Our main evaluation metric is the AUC score, which is resilient to class imbalance effects. We select the best-performing models on the validation set and report their AUC scores on the test sets.

Pre-processing. Following CLAM’s pre-processing pipeline [19], we cut all WSIs into 256×256 non-overlapping patches extracted solely from foreground tissue regions at x10 magnification. This results in approximately 0.6 million patches for Camelyon16 and 1.4 million patches for BRACS.

Feature Extraction. We extract features using self-supervised learning-based pre-trained backbones. For Camelyon16, we pre-train a ResNet18 (11.7M parameters) with Barlow-Twins [28]. For BRACS, we also use ResNet18 but pre-train it with DINO [4], to ensure that variations in results are not dependent on the pre-training method and since these two methods demonstrated state-of-the-art results [15]. All pre-training is conducted with the *solo-learn* library [7] for 200 epochs, with SSL hyper-parameters kept as in the original papers.

Training and Evaluation. We adopt DSMIL’s code [17] as the base for our training and evaluation pipeline, and we include five state-of-the-art (SOTA) MIL methods in our study: MaxMIL (baseline), ABMIL [13], DSMIL [17], CLAM [19], and TransMIL [23]. We use a cosine annealing scheduler, the Adam optimizer with a weight decay of 0.00001, and a batch size equal to one slide (i.e., one bag), as it is commonly done in the literature. The number of epochs is fixed to 100 for all methods to ensure a fair comparison. About the weight decay,

we found its influence to be negligible compared to LR. For further details on the hyper-parameters, please refer to the released code https://github.com/mammadov7/mil_merging.

3.2 Experiments

Variability Analysis. We evaluate our methods (Soup and Ties) on two datasets (Camelyon16 and BRACS), across five MIL models. We compare them with four other methods: *Baseline*, *LR tuned*, *Ensemble* and *Best on Val*. Each method is evaluated using ten different initialization seeds, thus obtaining ten different AUC scores on the test sets. The variability of the performances is evaluated using four metrics: minimum AUC, maximum AUC, mean AUC, and standard deviation.

Here, we give a brief description of each method, given an initialization seed: *Baseline* represents a single model that is trained for 100 epochs with a LR chosen based on prior knowledge from the literature (*i.e.*, LR=0.01 [29]). For *LR tuned* we perform a grid search over 6 learning rates ($\{0.1, 0.05, 0.01, 0.005, 0.001, 0.0001\}$). Each model is trained for 100 epochs, and we pick the best one on the validation set. The total number of training epochs is 600. *Soup3* and *Ties3* are our proposed methods with parameters $M = 10$, $K = 5$ and $T = 3$, which require $M \times K + 100 = 150$ epochs of training. *Ensemble* is an average of predictions from 10 fully trained models (each trained for 100 epochs with a random shuffling seed), resulting in 1000 total epochs. In *Best on VAL*, we select the model with the highest validation score among the 10 fully trained models and report its test performance (also 1000 total epochs). We tried using fewer epochs (3 or 5) and even an early-stopping rule. This decreased computational cost but it increased variability and degraded performance. By fully training for 100 epochs, we aimed to establish fair comparisons showing the best possible results (in terms of variability and performance) for the proposed baselines.

Ablation Study. We have conducted two ablation studies. In first one, we evaluate the influence of the hyper-parameters K , T , and initialization type using the MaxMIL method and $M = 10$. We compare three highly-used and well-known initialization strategies: i) *Uniform* initialization, where the initial weights are drawn from a uniform distribution with mean $\mu = 0$ and standard deviation $\sigma = 1$, ii) *Xavier*, weights are sampled from a normal distribution with $\mu = 0$ and $\sigma = \sqrt{\frac{2}{fan_{in} + fan_{out}}}$, where fan_{in}/fan_{out} are number of input/output signals, and iii) *Switch* initialization is done by drawing initial weights from a truncated normal distribution with $\mu = 0$ and $\sigma = \sqrt{s/n}$, where s is a scale hyper-parameter and n is the number of input units in the weight tensor. In the second study, we investigate the effect of the number of models to merge (T) on the performance of *Soup* and *Ties* methods across both datasets and all MILs, using $M = 10$ and $K = 100$ (thus each model is trained for 100 epochs before aggregation). Here, for each MIL, we change the value of T from 2 to 10 reporting the average AUC score on the test set of 10 runs.

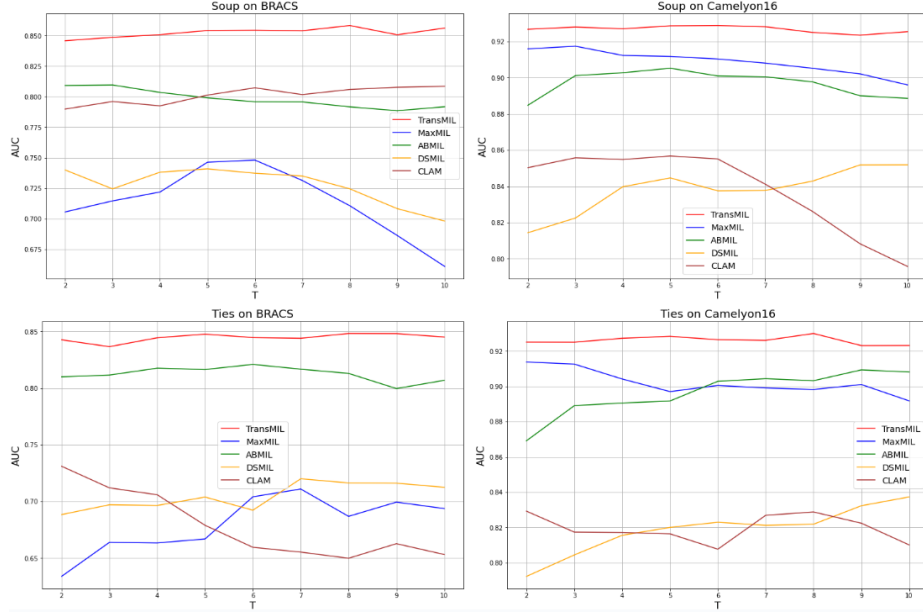


Fig. 3. Ablation Study on the effect of the T to the average AUC score across 5 MILs on Camelyon16 and BRACS datasets

3.3 Discussion

The proposed methods achieve better results in Table 2 than **Baseline** and **LR tuned** based on all metrics across all MILs and on both datasets. Only for BRACS, the Mean and Max of CLAM from *Baseline* and *LR tuned* is slightly better (0.1 AUC point), but it is important to consider that *LR tuned* requires 4 times more training epochs. Furthermore, our proposed methods obtain more stable results across all MIL methods and datasets, having the smallest **STD**. It's also important to notice that the proposed methods have a similar or better performance than **Ensemble** and **Best on VAL** models, but they require almost 7 times less training epochs and they are 10 times faster at inference. Eventually, it's worth mentioning that *Soup3/Ties3* improve the minimum results, which means that multi-fidelity based training helps to avoid local minima during training, and the best-performing methods for both dataset are our proposed methods with a maximum AUC of 95.7 points for Camelyon16 and 88.2 points for BRACS.

From the Ablation study (Table 1), we can see that results using $K = 10$ are slightly better than $K = 5$ or $K = 3$. However, this comes at a cost of increasing the number of training epochs. To keep the computational burden low, while preserving a good performance, we chose the combination $K = 5$ and $T = 3$, which gives almost always the best or second best performances on the Val and Test set (using $M = 10$). This means that aggregating 3 models out of 10 seems

to be a good compromise between stability and performance. This is why we chose $T = 3$ in Table 2.

The ablation study presented in Fig. 3 shows that increasing the number of models to merge does not significantly improve the average performance. Best results are obtained, on average, with a value between $T = 3$ and $T = 5$. $T = 3$ seems thus a good choice.

Table 1. Ablation study on: 1) number of epochs K , 2) number of aggregated models T and 3) initialization type using MaxMIL and $M = 10$.

Method	Uniform			Xavier			Switch			Uniform			Xavier			Switch											
	$K = 3 \text{ epochs}$									$K = 5 \text{ epochs}$									$K = 10 \text{ epochs}$								
	Val	Test	Val	Test	Val	Test	Val	Test	Val	Test	Val	Test	Val	Test	Val	Test	Val	Test									
<i>Soup3 (T=3)</i>	98.2	89.6	98.6	89.1	97.6	89.1	98.4	92.5	98.2	91.8	98.0	89.8	98.6	93.3	98.8	91.7	99.6	91.7									
<i>Soup5 (T=5)</i>	98.0	89.0	98.4	88.8	97.6	90.3	98.0	89.5	97.8	89.8	96.8	89.6	98.0	90.0	97.6	89.3	98.4	89.3									
<i>Soup (T=10)</i>	98.0	91.5	99.8	92.3	98.6	88.9	98.0	89.2	99.8	91.7	99.0	90.4	98.6	93.7	98.6	91.6	98.8	90.7									

Table 2. Variability analysis. Each method is repeated 10 times using 10 different initialization seeds. The 10 AUC on the test set are then used to compute the variability measures (Min, Max, Mean and STD). For each method and a given initialization seed, we report the number of models M (i.e., different shuffling seeds), the number of epochs K of initial training and the total number of training epochs Ep .

Camelyon16 Dataset																			
Method	M	K	Ep	MaxMIL				ABMIL				DSMIL				CLAM			
				Min	Max	Mean	STD	Min	Max	Mean	STD	Min	Max	Mean	STD	Min	Max	Mean	STD
<i>Baseline</i>	1	-	100	75.8	90.7	88.0	4.3	76.4	90.7	83.7	3.9	79.7	91.6	86.2	3.6	70.2	92.7	83.8	5.9
<i>LR tuned</i>	6	-	600	83.0	91.0	87.6	2.5	81.0	89.9	85.5	2.7	82.1	92.8	87.9	3.2	80.4	93.2	86.7	3.7
<i>Soup3 (ours)</i>	10	5	150	88.3	91.3	89.9	0.8	85.4	90.7	87.8	1.7	84.3	91.8	87.8	2.6	84.8	93.5	89.5	2.6
<i>Ties3 (ours)</i>	10	5	150	86.3	91.7	89.8	1.6	79.0	91.5	85.5	3.6	85.7	92.7	89.0	1.8	84.9	93.1	89.7	2.2
<i>Ensemble</i>	10	-	1000	88.0	92.7	90.3	1.3	86.6	93.2	91.1	2.1	79.1	88.4	83.9	3.1	84.9	94.3	87.8	2.7
<i>Best on VAL</i>	10	-	1000	88.3	92.9	91.0	1.4	79.9	92.2	85.6	4.2	82.6	92.8	88.5	3.5	77.0	89.3	83.5	4.2
BRACS Dataset																			
<i>Baseline</i>	1	-	100	50.8	74.3	58.7	7.3	73.5	83.3	78.4	3.1	66.1	81.3	73.8	4.7	71.2	83.5	79.6	3.1
<i>LR tuned</i>	6	-	600	64.9	86.5	77.2	5.5	73.8	87.2	81.0	4.4	67.3	83.6	78.7	4.2	70.4	83.6	79.3	4.1
<i>Soup3(ours)</i>	10	5	150	81.0	86.9	85.0	1.7	81.3	86.6	84.6	1.5	76.1	86.1	80.0	2.8	73.3	83.5	79.5	2.8
<i>Ties3(ours)</i>	10	5	150	81.0	86.9	84.8	2.0	82.6	87.4	84.7	1.4	74.9	81.4	78.7	1.8	76.6	81.5	79.1	1.6
<i>Ensemble</i>	10	-	1000	65.6	84.1	76.4	5.8	78.5	83.2	81.1	1.6	71.6	84.9	77.4	3.7	77.9	82.0	80.2	1.3
<i>Best on VAL</i>	10	-	1000	70.7	86.6	79.1	5.2	76.4	85.2	79.8	2.5	71.3	80.8	76.2	2.7	74.0	82.1	78.2	2.7

4 Conclusion

MIL methods suffer from high variability in performance across different runs, which can hamper reproducibility and trustworthiness when comparing different

methods. To address this issue, we introduced a simple strategy based on model averaging and multi-fidelity optimization. Our experiments demonstrated that the proposed method reduces performance variability across runs while preserving top performance and maintaining a sustainable computational burden. As demonstrated in Soup [24] (Fig. L.1, L.2), techniques like SWA or EMA are complementary to our approach and could be integrated to further boost performance. However, they come with additional computational overhead. Integrating these methods is a potential direction for future improvement.

Acknowledgments. This paper has been supported by the French National Research Agency (ANR-20-THIA-0012) and by the Hi!PARIS Center on Data Analytics and Artificial Intelligence. Furthermore, this work was performed using HPC resources from GENCI-IDRIS (Grant 2023-AD011013982R1).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Belz, A., Agarwal, S., Shimorina, A., Reiter, E.: A systematic review of reproducibility research in natural language processing. In: Merlo, P., Tiedemann, J., Tsarfaty, R. (eds.) *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. pp. 381–393. Association for Computational Linguistics, Online (Apr 2021)
2. Bouthillier, X., Laurent, C., Vincent, P.: Unreproducible research is reproducible. In: *International Conference on Machine Learning*. pp. 725–734. PMLR (2019)
3. Brancati, N., Anniciello, A.M., Pati, P., Riccio, D., Scognamiglio, G., Jaume, G., De Pietro, G., Di Bonito, M., Foncubierta, A., Botti, G., Gabrani, M., Feroce, F., Frucci, M.: BRACS: A Dataset for BReAst Carcinoma Subtyping in H&E Histology Images. *Database: The Journal of Biological Databases and Curation* **2022**, baac093 (Oct 2022)
4. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: *Proceedings of the IEEE/CVF international conference on computer vision*. pp. 9650–9660 (2021)
5. Chan, T.H., Cendra, F.J., Ma, L., Yin, G., Yu, L.: Histopathology whole slide image analysis with heterogeneous graph representation learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023)
6. Chen, R.J., Chen, C., Li, Y., Chen, T.Y., Trister, A.D., Krishnan, R.G., Mahmood, F.: Scaling vision transformers to gigapixel images via hierarchical self-supervised learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 16144–16155 (2022)
7. Costa, V.G.T.d., Fini, E., Nabi, M., Sebe, N., Ricci, E.: solo-learn: A Library of Self-supervised Methods for Visual Representation Learning. *Journal of Machine Learning Research* **23**(56), 1–6 (2022)
8. Egele, R., Guyon, I., Sun, Y., Balaprakash, P.: Is one epoch all you need for multi-fidelity hyperparameter optimization? In: *31st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2023*

9. Ehteshami Bejnordi, B., Veta, M., Johannes van Diest, P., van Ginneken, B., Karssemeijer, N., Litjens, G., van der Laak, J.A.W.M., the CAMELYON16 Consortium: Diagnostic Assessment of Deep Learning Algorithms for Detection of Lymph Node Metastases in Women With Breast Cancer. *JAMA* **318**(22) (2017)
10. Fernández-Godino, M.G.: Review of multi-fidelity models. *Advances in Computational Science and Engineering* **1**(4), 351–400 (2023)
11. Gundersen, O.E., Coakley, K., Kirkpatrick, C., Gil, Y.: Sources of irreproducibility in machine learning: A review. *arXiv preprint arXiv:2204.07610* (2022)
12. Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., Meger, D.: Deep reinforcement learning that matters. In: *Proceedings of the AAAI conference on artificial intelligence*. vol. 32 (2018)
13. Ilse, M., Tomczak, J., Welling, M.: Attention-based Deep Multiple Instance Learning. In: *Proceedings of the 35th International Conference on Machine Learning*. pp. 2127–2136. PMLR (Jul 2018), iSSN: 2640-3498
14. Jaume, G., Vaidya, A., Zhang, A., H. Song, A., J. Chen, R., Sahai, S., Mo, D., Madrigal, E., Phi Le, L., Mahmood, F.: Multistain pretraining for slide representation learning in pathology. In: *European Conference on Computer Vision*. pp. 19–37. Springer (2024)
15. Kang, M., Song, H., Park, S., Yoo, D., Pereira, S.: Benchmarking Self-Supervised Learning on Diverse Pathology Datasets. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 3344–3354 (2023)
16. Lazard, T., Lerousseau, M., Decencière, E., Walter, T.: Giga-ssl: Self-supervised learning for gigapixel images. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 4305–4314 (2023)
17. Li, B., Li, Y., Eliceiri, K.W.: 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)
18. Li, J., Chen, Y., Chu, H., Sun, Q., Guan, T., Han, A., He, Y.: Dynamic graph representation with knowledge-aware attention for histopathology whole slide image analysis. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 11323–11332 (2024)
19. Lu, M.Y., Williamson, D.F., Chen, T.Y., Chen, R.J., Barbieri, M., Mahmood, F.: Data-efficient and weakly supervised computational pathology on whole-slide images. *Nature biomedical engineering* **5**(6), 555–570 (2021), publisher: Nature Publishing Group UK London
20. Lucic, M., Kurach, K., Michalski, M., Gelly, S., Bousquet, O.: Are gans created equal? a large-scale study. *Advances in neural information processing systems* **31** (2018)
21. Mammadov, A., Folgoc, L.L., Adam, J., Buronfosse, A., Hayem, G., Hocquet, G., Gori, P.: Self-supervision enhances instance-based multiple instance learning methods in digital pathology: a benchmark study. *Journal of Medical Imaging* **12**(6), 061404 (2025)
22. Pineau, J., Vincent-Lamarre, P., Sinha, K., Larivière, V., Beygelzimer, A., d’Alché Buc, F., Fox, E., Larochelle, H.: Improving reproducibility in machine learning research (a report from the neurips 2019 reproducibility program). *Journal of machine learning research* **22**(164), 1–20 (2021)
23. Shao, Z., Bian, H., Chen, Y., Wang, Y., Zhang, J., Ji, X., others: Transmil: Transformer based correlated multiple instance learning for whole slide image classification. *Advances in neural information processing systems* **34**, 2136–2147 (2021)

24. Wang, Z., Ma, J., Gao, Q., Bain, C., Imoto, S., Liò, P., Cai, H., Chen, H., Song, J.: Dual-stream multi-dependency graph neural network enables precise cancer survival analysis. *Medical Image Analysis* **97**, 103252 (2024)
25. Wortsman, M., Ilharco, G., Gadre, S.Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A.S., Namkoong, H., Farhadi, A., Carmon, Y., Kornblith, S., et al.: Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In: *International conference on machine learning*. pp. 23965–23998. PMLR (2022)
26. Yadav, P., Tam, D., Choshen, L., Raffel, C.A., Bansal, M.: Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems* **36** (2024)
27. Yang, S., Wang, Y., Chen, H.: Mambamil: Enhancing long sequence modeling with sequence reordering in computational pathology. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. pp. 296–306. Springer (2024)
28. Zbontar, J., Jing, L., Misra, I., LeCun, Y., Deny, S.: Barlow twins: Self-supervised learning via redundancy reduction. In: *International Conference on Machine Learning*. pp. 12310–12320. PMLR (2021)
29. Zhang, Y., Li, H., Sun, Y., Zheng, S., Zhu, C., Yang, L.: Attention-challenging multiple instance learning for whole slide image classification. In: *European Conference on Computer Vision*. pp. 125–143. Springer (2024)
30. Zheng, Y., Gindra, R.H., Green, E.J., Burks, E.J., Betke, M., Beane, J.E., Kolachalama, V.B.: A graph-transformer for whole slide image classification. *IEEE transactions on medical imaging* **41**(11), 3003–3015 (2022)