Scaling Multi-Teacher Distillation for Digital Pathology

Sofiène Boutaj[1,2*], **Pierre Marza**[1,2*], **Varun Belagali**[3], **Dimitris Samaras**[3], **Maria Vakalopoulou**[1,2], **Stergios Christodoulidis**[1,2]

[1] Université Paris-Saclay, CentraleSupélec, Gustave Roussy, INSERM, IHU PRISM, Cancer Data Science Unit, France
[2] Université Paris-Saclay, CentraleSupélec, MICS Laboratory, France
[3] Department of Computer Science, Stony Brook University, USA

Project page Github Models Datasets

Abstract

Multi-teacher distillation allows transferring knowledge from multiple "teacher" networks to a single "student" network. This method is promising in fields such as digital pathology, where many powerful foundation models were proposed recently. However, as we show in this paper, the standard multi-teacher distillation approach reacts poorly to an increase in the number of teachers used to train a student encoder. We demonstrate that learning teacher-specific representations is a key to scaling in MTD. Importantly, different design choices, such as learnable teacher tokens, a tailored attention scheme, additional mixture-of-experts layers and a contrastive loss, are proposed to better learn such teacher-specific representations. We show that our method better scales with respect to the number of considered teachers, allowing us to train compact student encoders, up to $10\times$ more efficient than larger teacher foundation models, while matching their performance and even outperforming them on a large set of tile-level and slide-level tasks. We conduct a thorough empirical validation, evaluating more than 10 foundation models on 39 tasks spanning tile and slide levels. As a result, we release a new collection of strong and efficient foundation models, named **OMNI**, trained from the knowledge of 10 state-of-the-art foundation models. We finally conduct an analysis of the learned teacher-specific representations, highlighting their complementarity and explaining why they can be easily aggregated at downstream time.

1 Introduction

Knowledge distillation [24] has been extensively studied over the last few decades, yet the main goal has stayed the same throughout it: transferring knowledge from a large model, typically referred to as the teacher, to a smaller one, the student. Such knowledge distillation can either be related to a specific downstream task (supervised [24]) or focus on matching a representation distribution (self-supervised [16]). An extension of this line of research is Multi-Teacher Distillation (MTD), where the objective is to distill knowledge from a set of diverse teachers [54, 53, 60, 58, 57] by fusing different representations into a powerful common latent space.

[*] denotes equal contribution.
Corresponding authors: sofiene.boutaj@centralesupelec.fr, pierre.marza@centralesupelec.fr

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

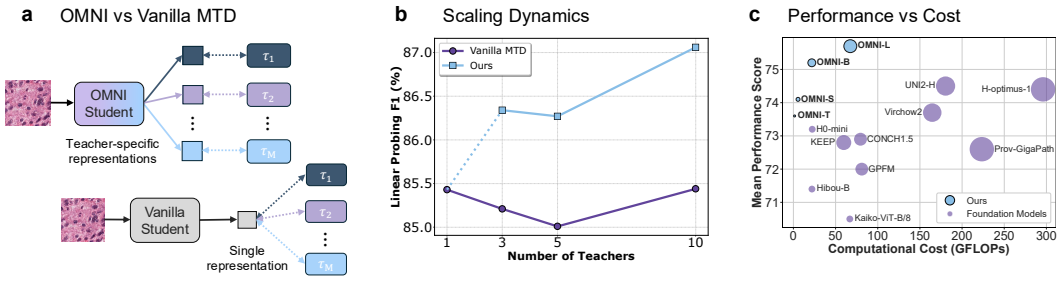


Figure 1: **OMNI**. We propose a new method to scale Multi-Teacher Distillation (MTD) to more teachers. **(a)** Unlike vanilla MTD, we learn to extract teacher-specific representations. **(b)** Downstream performance (tile-level linear probing averaged across 13 tasks) better scales when increasing the diversity of teachers for our method. For a fair evaluation, we rank teachers by their individual performance on the THUNDER benchmark and add them from strongest to weakest. **(c)** Overall we provide a collection of light vision encoder ($7 - 330M$ parameters) models that are competitive with and even outperform all state-of-the-art larger teacher foundation models. In particular, we report here the average of the tile-level and slide-level linear probing performances (see Table 1 for more details) as a function of GFLOPs. Circle sizes indicate the number of parameters of each model.

Importantly, while recent work [53, 60, 58, 57] has improved upon the original methods [54], a central question has not yet been addressed in the MTD literature: *What are the scaling dynamics of MTD with respect to the number of considered teachers?* Indeed, most previous work only leverages a few teachers, often 3 or rarely 5 of them, and does not explore the benefits of adding more teachers.

Such a gap in the literature is mainly visible now that the diversity of existing foundation models has dramatically increased in the past few years. A typical example of this is digital pathology where, similarly to other medical imaging sub-fields, many pre-trained vision encoders were released recently [12, 69, 80, 55, 18, 19, 46, 32, 66, 70, 74, 41, 13, 78, 26, 25, 73, 59]. MTD is particularly appealing in this context and, despite some recent works exploring its application to digital pathology [42, 22, 36, 6], its scaling remains largely underexplored.

In this study, we thus investigate the scaling laws of MTD, with a specific focus on digital pathology foundation models. We discover, as shown in Figure 1b, that increasing the number of teachers for vanilla MTD (baseline) does not improve downstream performance, and even degrades it. We hypothesize that this drop in performance stems from the difficulty of reconstructing a large set of distinct embedding spaces from a single, usually small, common latent space.

Aiming in solving the scaling problem in MTD, we propose a simple yet powerful method based on teacher-specific representations (Figure 1a). Such a design leads to a better performance scaling with respect to the integration of more foundation models as teachers (Figure 1b). We show that our method is efficient at distilling up to 10 pathology foundation models, some of which are up to $1.1B$ parameters, into smaller students ($7M - 330M$ parameters) trained on only $10M$ public TCGA tiles. Moreover, we release a new family of models, named **OMNI**, that outperform state-of-the-art larger foundation models used as teachers while being parameter-efficient (Figure 1c).

Our contributions can be summarized as follows:

- **Learning teacher-specific representations allows scaling in MTD**: We show that vanilla MTD does not directly benefit from broadening the pool of teachers, and overcome this limitation by training our student model to extract teacher-specific representations instead of a single global representation.
- **A novel methodological framework supporting teacher-specific representation learning**: We propose important design choices, such as learnable teacher embeddings, Mixture-of-Expert (MoE) layers, and a specific attention masking to better learn teacher-specific representations.
- **A new collection of efficient and powerful pathology foundation models**: We introduce **OMNI**, a collection of lightweight pathology vision encoders with different sizes (OMNI-T, OMNI-S, OMNI-B, OMNI-L), distilled from 10 of the current best state-of-the-art open-sourced foundation models. While being up to $10\times$ more efficient (GFLOPs) compared

with concurrent models, they are competitive and even outperform all of them on a diverse set of 39 tile-level and slide-level tasks.

2 Related work

Improving the flexibility of the Transformer — The Transformer model [67], or its vision counterpart ViT [14], is able to adapt representations through its core attention mechanism. However, more recent work has proposed methods to improve such a flexibility on several complementary axes. A first direction consists in designing specialized attention schemes. A prominent example arises in multi-modal applications, where specific cross-attention can be applied to enrich one modality with information coming from another one [64, 2, 39, 34]. Optimizing a set of input token embeddings for a specific task or domain is another way to adapt Transformer-based representations. This has been extensively explored when adapting a pre-trained Transformer to a new task: prompt tuning [40, 29, 79] introduces a small set of learnable tokens prepended to the input, providing a lightweight yet effective way to steer the model toward task-specific behavior. Finally, another orthogonal approach is the integration of Mixture-of-Experts (MoE) layers into Transformer-based models [61, 30, 17]. Their main goal is to allow the activation of different experts, i.e. feed-forward networks, to modulate neural activations based on the input.

We follow this direction of making a student ViT-based model more flexible to learn diverse teacher-specific representations, revisiting the above ideas in the context of multi-teacher distillation.

Multi-teacher distillation (MTD) — RADIO [54] was one of the first papers proposing to fuse the knowledge of various pre-trained vision encoders by training a student model to decode their representations simultaneously. Theia [60] applied this concept to robotics, showing the advantage of aggregating diverse teachers, and focused specifically on improving dense spatial representations. Following this, different works addressed arising concerns when considering diverse teachers. First, PHI-S [53] proposed to study how to normalize representations from different teachers using diverse standardization techniques. Then, UNIC [57] studied the impact of regularization techniques such as teacher dropping, and DUNE [58] went one step further by considering very different teachers (2D, 3D, human pose estimation) and showed how improving the capacity of decoder heads was beneficial in such a regime. However, the current MTD frameworks include three to five teachers without studying scaling laws to more teachers.

Unlike previous work, we highlight a lack of clear scaling trends in multi-teacher distillation and propose a method to better benefit from integrating more teachers. This allows to distill knowledge from 10 state-of-the-art foundation model teachers into a compact student vision encoder.

MTD for histopathology — Multi-teacher distillation has recently been studied in the field of digital pathology to leverage existing foundation models as teachers. GPFM [42] proposed integrating MTD with DINOv2-style [51] self-distillation pretraining. Pathryoshka [22] explored MTD with Matryoshka-based [36] nested embeddings, providing adaptable embedding dimensions in the student model. Further, TICON [6] explored MTD by pretraining a slide-encoder on multiple tile-encoder embeddings with an omni-feature masked modeling approach. Current MTD frameworks in this domain have explored pretraining using only up to three teachers and have not studied the scaling trends with respect to the number of considered teachers.

Pathology foundation models — are trained with self-supervised objectives to capture patterns in large-scale unsupervised histopathology datasets. We can differentiate vision-only [12, 69, 80, 55, 18, 19, 46, 32, 66, 70, 74, 6] and vision-language [41, 13, 78, 26, 25, 73, 59] models. The former are trained with DiNO-style losses [11, 51] and the latter following CLIP-like objectives [52, 75]. As shown in previous work benchmarking pathology foundation models [72, 31, 48, 23, 3, 21, 28, 8, 37, 44, 4, 77, 10, 43, 45], these models have different strengths and weaknesses, and their ranking depends on the data domains and tasks to solve. Such a result indicates that they extract diverse and, thus, complementary information, mainly due to their pretraining data.

In this work, we propose to train an efficient vision model from the knowledge of a large set of open-sourced foundation models in order to leverage their diverse and complementary information.

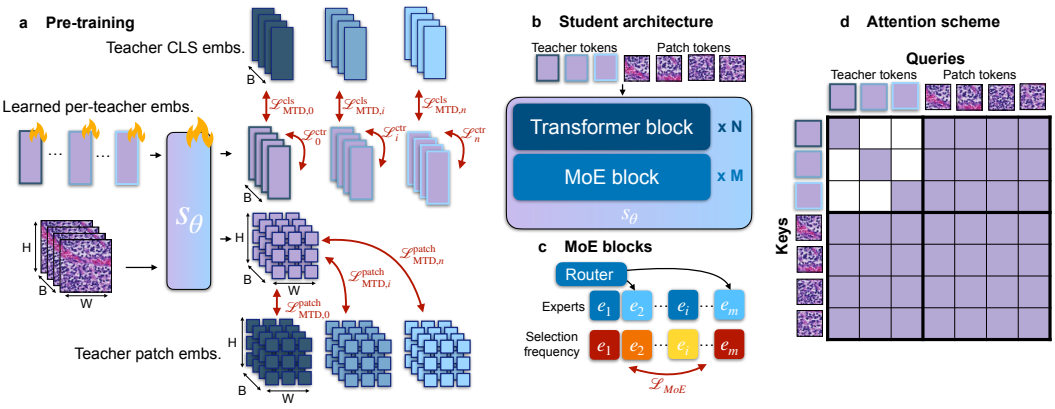


Figure 2: **Method overview**. **(a)** We train teacher-specific embeddings along with the student model with a token loss ($\mathcal{L}^{\text{cls}}_{\text{MTD},i}$ for each teacher T_i), a dense patch loss ($\mathcal{L}^{\text{patch}}_{\text{MTD},i}$) and a contrastive loss ($\mathcal{L}^{\text{ctr}}_i$). **(b)** The student architecture is composed of a sequence of transformer and Mixture-of-Experts (MoE) layers. **(c)** Each MoE layer is based on an expert-selection mechanism where top-2 experts are picked conditioned on the input. An additional regularization loss ($\mathcal{L}_{\text{MoE}}$) ensures a diversity in the selection of the experts. **(d)** All attention operations in the student follow a structured masking. Each teacher token attends to itself and to all patch tokens, but not to other teacher tokens, while patch tokens attend to all tokens.

3 Training a vision encoder from multiple teachers

3.1 Preliminaries

Vision Transformer — Our method specifically addresses the shortcomings of Vision Transformer (ViT) [14] models trained with multi-teacher distillation. A ViT processes an image $x \in \mathbb{R}^{H\times W\times C}$ by first splitting it into N_p non-overlapping patches of size $P \times P$, yielding $N_p = HW/P^2$ patch tokens. Each patch is linearly projected to a D-dimensional embedding, and a single learnable *CLS token* $\mathbf{c} \in \mathbb{R}^D$ is prepended, giving the input sequence,

$$\mathbf{X}_0 = \left[\mathbf{c},\ \mathbf{z}^{\text{patch}}_1,\ \ldots,\ \mathbf{z}^{\text{patch}}_{N_p}\right] + \mathbf{E}^{\text{pos}} \ \in\ \mathbb{R}^{(1+N_p)\times D}, \tag{1}$$

where $\mathbf{E}^{\text{pos}} \in \mathbb{R}^{(1+N_p)\times D}$ are learnable positional embeddings. In the next paragraphs, we will refer to the sequence of patch tokens only as $\mathbf{Z}^{\text{patch}}$.

$\mathbf{X}_0$ is passed through L transformer blocks, each consisting of a multi-head self-attention (MHSA) and a feed-forward network (FFN),

$$\mathbf{X}_\ell = \text{FFN}(\text{MHSA}(\mathbf{X}_{\ell-1})) + \mathbf{X}_{\ell-1}, \qquad \ell = 1,\ldots,L. \tag{2}$$

In standard MHSA, every token attends to every other token, corresponding to a fully dense attention mask $\mathbf{A} = \mathbf{1}_{(1+N_p)\times(1+N_p)}$. After the final block, the CLS token $\mathbf{z}^{\text{cls}} = [\mathbf{X}_L]_0 \in \mathbb{R}^D$ serves as the global image representation used for downstream tasks.

Vanilla [54] — Let us define a ViT-based student model s_θ parametrized by weights θ and a set of M ViT teacher models $\mathcal{T} = \{t_{\psi_i}\}_{i=1\ldots M}$, each parametrized by weights ψ_i. The objective is then to reconstruct, for a given input image, the representations of all teachers from that of the student. To this end, we must also introduce, for each teacher, *(i)* a linear layer $p^{\text{cls}}_{w_i}$ parametrized by weights w_i to project the student CLS token to the teacher CLS embedding space, and *(ii)* a linear layer $p^{\text{patch}}_{\bar{w}_i}$ parametrized by weights $\bar{w}_i$ to project the student patch tokens to the teacher patch embedding space.

Given a training dataset $\mathcal{D} = \{d_j\}_{j=1\ldots N}$ composed of N samples, the vanilla MTD objective is,

$$\hat{\theta} = \arg\min_\theta \sum_{i=1}^{M}\sum_{j=1}^{N} \mathcal{L}^{\text{cls}}_{\text{MTD},i}(\theta,\psi_i,w_i,d_j) + \mathcal{L}^{\text{patch}}_{\text{MTD},i}(\theta,\psi_i,\bar{w}_i,d_j), \tag{3}$$

where losses are themselves computed as follows,

$$\mathcal{L}^{\text{cls}}_{\text{MTD},i}(\theta, \psi_i, w_i, d_j) = \Phi\left[p^{\text{cls}}_{w_i}(s_\theta(d_j;\theta)\mathbb{1}_{cls}; w_i)\ ,\ t_{\psi_i}(d_j;\psi_i)\mathbb{1}_{cls}\right],$$
$$\mathcal{L}^{\text{patch}}_{\text{MTD},i}(\theta, \psi_i, \bar{w}_i, d_j) = \Phi\left[p^{\text{patch}}_{\bar{w}_i}(s_\theta(d_j;\theta)\mathbb{1}_{patch}; \bar{w}_i)\ ,\ t_{\psi_i}(d_j;\psi_i)\mathbb{1}_{patch}\right], \tag{4}$$

with $\mathbb{1}_{cls}$ and $\mathbb{1}_{patch}$ being indicator functions respectively indexing only CLS and patch tokens. $\Phi(\mathbf{x}, \mathbf{y}) = [1 - \cos(\mathbf{x}, \mathbf{y})] + \lambda_{L_1}\text{smooth}_{L_1}(\mathbf{x} - \mathbf{y})$ combines a cosine and smooth L1 losses, as done in previous work [54, 53, 60, 58, 57, 42, 22].

3.2 Teacher tokens and mixture-of-experts blocks

A central challenge in multi-teacher distillation is that teachers might capture different information. A single shared CLS token must simultaneously compress all of these into one representation, which becomes increasingly bottlenecked as M grows (Figure 1b). We address this issue with two complementary design choices: teacher-specific tokens with a structured attention mask, and Mixture-of-Experts (MoE) layers. An overview of the pre-training scheme is presented in Figure 2a.

Teacher-specific tokens — In vanilla MTD, all M teachers must be reconstructed from a single shared CLS token $\mathbf{z}^{\text{cls}}$, forcing it to simultaneously compress potentially different visual abstractions into one vector. We therefore abandon the single-token design and instead assign one independent token per teacher. Formally, we maintain a learnable token matrix $\mathbf{C} \in \mathbb{R}^{M\times D}$ and a corresponding positional embedding matrix $\mathbf{E}^{\text{teach}} \in \mathbb{R}^{M\times D}$. For a given input and a set of M active teachers, we prepend their M specific tokens to the N_p patch tokens, forming a sequence of length $M + N_p$:

$$\mathbf{X} = \left[(\mathbf{C} + \mathbf{E}^{\text{teach}})\ ,\ \mathbf{Z}^{\text{patch}}\right] \in \mathbb{R}^{(M+N_p)\times D}. \tag{5}$$

Each teacher-specific token can thus learn to selectively aggregate the patch-level information most relevant to its assigned teacher.

Structured attention masking — To encourage the learning of a specific aggregation of patch token embeddings from each teacher token, we apply in all the layers the following block-structured attention mask $\mathbf{A} \in \{0, 1\}^{(M+N_p)\times(M+N_p)}$:

$$\mathbf{A} = \begin{pmatrix} \mathbf{I}_M & \mathbf{1}_{M\times N_p} \\ \mathbf{1}_{N_p\times M} & \mathbf{1}_{N_p\times N_p} \end{pmatrix}, \tag{6}$$

where a value of 1 indicates that the query (row) can attend to the key (column), and 0 indicates masking. The three colored blocks encode distinct attention roles: (i) $\mathbf{I}_M$: each token attends only to itself among teacher tokens; (ii) $\mathbf{1}_{M\times N_p}$ and $\mathbf{1}_{N_p\times M}$: every teacher-specific token attends to all patch tokens and vice versa, enabling full spatial aggregation per teacher; (iii) $\mathbf{1}_{N_p\times N_p}$ patch tokens attend freely among themselves, preserving standard self-attention over image content.

The masking of cross-teacher attention (Figure 2d) is an important design choice. By enforcing $\mathbf{I}_M$ in the teacher-to-teacher block, we guarantee that each teacher token is exclusively informed by the patch tokens processed rather than by the already-aggregated views of other teachers. The objective here is to obtain a set of M teacher-specialized representations that share computation at the patch level but diverge at the aggregation level.

MoE blocks — As a complementary mechanism, the last transformer blocks replace the standard dense feed-forward networks with Mixture-of-Experts (MoE) layers, increasing the total model capacity while keeping inference cost constant. Again, to provide more flexibility in the extraction of image representations, for each token, a MoE router adaptively selects a set of relevant experts, i.e., feed-forward networks (Figure 2b, c). To encourage balanced expert utilization, we apply the Switch Transformer load-balancing loss [17]:

$$\mathcal{L}_{\text{MoE}} = E \cdot \sum_{e=1}^{E} f_e \cdot \bar{p}_e, \tag{7}$$

where f_e is the fraction of tokens routed to expert e and $\bar{p}_e$ is the mean routing probability assigned to expert e over the batch. This loss is summed across all MoE blocks and added to the total objective with coefficient α_{MoE}=0.05.

3.3 Training objective

Per-teacher loss — The total per-teacher loss combines CLS-level and patch-level cosine losses with an InfoNCE contrastive term:

$$\mathcal{L}_i = \mathcal{L}^{\text{cls}}_{\text{MTD},i} + \mathcal{L}^{\text{patch}}_{\text{MTD},i} + \lambda_{\text{ctr}}\,\mathcal{L}^{\text{ctr}}_i, \tag{8}$$

with λ_{ctr}=0.1.

InfoNCE contrastive loss — The contrastive term $\mathcal{L}^{\text{ctr}}_i$ is a symmetric InfoNCE loss [50] computed between the projected student teacher-specific embeddings and the corresponding ground-truth teacher CLS embeddings within a batch of size B. Let $\tilde{\mathbf{z}}^{(b)}_i = \ell_2\big(p^{\text{cls}}_i(s_\theta(d_b))\big)$ and $\tilde{\mathbf{y}}^{(b)}_i = \ell_2\big(\mathbf{y}^{\text{cls}}_i(d_b)\big)$ denote ℓ_2-normalized student and teacher embeddings for sample b, where $s_\theta(d_b)$ is the student teacher-specific token for teacher i, p^{cls}_i the corresponding linear projection head, and $\mathbf{y}^{\text{cls}}_i(d_b)$ the CLS embedding of teacher i. The pairwise logit matrix $\mathbf{S}_i \in \mathbb{R}^{B\times B}$ is defined as

$$[\mathbf{S}_i]_{ab} = \frac{\tilde{\mathbf{z}}^{(a)\top}_i \tilde{\mathbf{y}}^{(b)}_i}{\tau}, \tag{9}$$

with temperature τ=0.1. The InfoNCE loss is then the average of two cross-entropy terms that treat the diagonal entries as the positives:

$$\mathcal{L}^{\text{ctr}}_i = \frac{1}{2}\left[-\frac{1}{B}\sum_{b=1}^{B}\log\frac{e^{[\mathbf{S}_i]_{bb}}}{\sum_{b'=1}^{B} e^{[\mathbf{S}_i]_{bb'}}} \;-\; \frac{1}{B}\sum_{b=1}^{B}\log\frac{e^{[\mathbf{S}_i]_{bb}}}{\sum_{a=1}^{B} e^{[\mathbf{S}_i]_{ab}}}\right]. \tag{10}$$

This batch-level discriminative signal complements the pointwise cosine objective by penalizing embeddings that collapse to similar representations across different samples.

Final objective — The full training loss sums over all M teachers and incorporates the MoE loss:

$$\mathcal{L} = \frac{1}{M}\sum_{i=1}^{M}\mathcal{L}_i + \alpha_{\text{MoE}}\,\mathcal{L}_{\text{MoE}}. \tag{11}$$

3.4 Downstream aggregation

At downstream time, the student produces M teacher-specific tokens which must be aggregated into a single representation. We use a simple mean-pooling operator which, as supported by our experimental results (Section 4), leads to high performance. This might be explained by the disentanglement of teacher representations as studied in Figure 3b. We also briefly experiment with learning a set of teacher-specific weights along with a MIL aggregator (Table 2) to perform a weighted mean-pooling.

4 Experimental results

4.1 Experimental Setup

Pre-training dataset — All distillation experiments use a pre-training dataset of 10M tiles extracted from 10K whole-slide images sourced from The Cancer Genome Atlas (TCGA), spanning a wide range of tissue types and organ systems, including lung, breast, colon, kidney, and brain.

Teacher models — We consider a heterogeneous set of M=10 frozen teacher encoders covering both histopathology-specific and general-purpose vision backbones, and both vision-only and vision-language pretraining objectives. The pool consists of: H-optimus-1 [56], Virchow2 [80], UNI2-h [12], Prov-GigaPath [74], Kaiko-ViT-B/8 [1], CONCH 1.5 [41, 13], Hibou-L [46], H0-mini [20], KEEP [78], and DINOv3-ViT-L/16 [62]. These teachers differ in model capacity (from 86M to 1.1B parameters), embedding dimensionality (from 384 to 1,536), and patch grid resolution (7×7 to 28×28). All teachers are kept frozen throughout student training.

Concurrent distillation methods — We consider two close concurrent methods: H0-mini [20], which is part of our pool of teachers and is trained with single-teacher distillation, and GPFM [42] trained with multi-teacher distillation. Checkpoints for both models are publicly released allowing us to compare them against OMNI. We do not compare with TICON [6] which is a slide-level encoder

and is thus not directly comparable, or with Pathryoshka [22] as no public code or pre-trained weights are currently publicly available. H0-mini is trained with a supervision coming from H-Optimus-0 [55], and GPFM from UNI [12], Prov-GigaPath [74] and Phikon [18]. Both of them are trained on a set of publicly available slides (H0-mini: $6k$ WSIs, $43M$ tiles; GPFM: $72k$ WSIs, $190M$ tiles). We thus train our model on a more diverse set of teachers but from less data ($10k$ WSIs, $10M$ tiles). Despite these discrepancies, we believe that comparing OMNI to both methods is relevant as they are the current state-of-the-art open-source distilled models that can be used in clinical applications.

Student model and MoE configuration — We train a family of four student models of increasing capacity: OMNI-T (7.3M), OMNI-S (28.7M), OMNI-B (114M), and OMNI-L (330M). For each, the last K transformer blocks use MoE layers with $E{=}5$ experts and top-$k{=}2$ routing ($K{=}3$ for T/S/B, $K{=}5$ for L), and the model outputs $M{=}10$ independent per-teacher CLS tokens. Full architecture details are provided in Appendix F.

Tile-level evaluation — Tile-level representations are evaluated using the **THUNDER** benchmark [45], which covers 13 datasets spanning diverse tissue types, cancer subtypes, and staining protocols: BACH [5] (breast tissue subtypes), BRACS [7] (breast carcinoma subtypes), BreakHis [63] (breast tumor magnification), CCRCC [9] (clear-cell renal carcinoma), CRC [33] (colorectal cancer tissue types), ESCA [65] (esophageal carcinoma), MHIST [71] (colorectal polyp grading), PCam [68] (lymph node metastasis detection), WILDS [35] (multi-site breast cancer), and four SPIDER subtasks [47] covering bone marrow cytology (SPIDER-B, SPIDER-C, SPIDER-S, SPIDER-T). Importantly, as we train our model on a subset of TCGA, we have removed the three TCGA-derived datasets from THUNDER to avoid data leakage that could advantage our method in empirical comparisons. We evaluate linear probing (LP), reporting macro-averaged F1 score (Table 1). We additionally evaluate adversarial robustness under a PGD attack ($\varepsilon{=}1.5{\times}10^{-3}$), reporting the absolute F1 drop relative to clean F1-score, as well as calibration via Expected Calibration Error (ECE) (Table 3). KNN and few-shot classification results are provided in the supplementary material (Table 8).

Slide-level evaluation — For slide-level evaluation we use the **PathoBench** benchmark [77] on CPTAC cohorts [15], covering 7 cancer types and 25 tasks in total: BRCA (3 tasks), CCRCC (2), COAD (7), GBM (3), LSCC (4), LUAD (4), and UCEC (2). Tasks include subtype classification and molecular marker prediction. Following the PathoBench protocol, we report balanced accuracy (BACC), and we evaluate the CPTAC tasks using a 50-fold train/test split, while employing 5-fold cross-validation for all other WSI datasets. For all splits, we ensure there is no data leakage between the training and test sets. Slide-level tile features are extracted using the TRIDENT pipeline [77] and aggregated via mean pooling, followed by a linear classifier (Table 1).

Further hyperparameters and implementation details are provided in Appendix (Sections E, F, G).

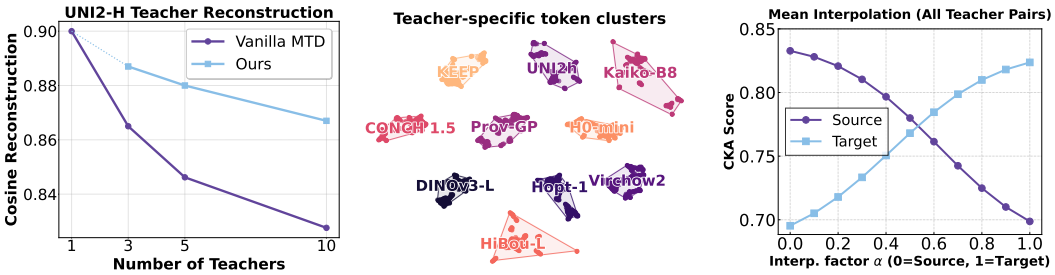
No data leakage caused by an increased number of teachers — According to the publicly-available information, no test dataset from our downstream evaluations is included in the training set of the teacher foundation models. We base our study on the reported training datasets in Table S1 (appendix) from [45], which is partially derived from Table 1 in [38]. As can be seen in these tables, our considered teachers were trained from private sources, *TCGA*, *GTEx*, *Quilt1M*, *OpenPath* and natural-domain datasets, which, to the best of our knowledge, do not overlap with our test datasets.

4.2 Downstream performance

Linear probing benchmarking — Table 1 reports linear probing performance across tile- and slide-level benchmarks. The full OMNI family demonstrates a consistent scaling trend: even OMNI-T, with only 7.3M parameters and 1.39 GFLOPs, achieves competitive slide-level performance (63.2 Bacc), while OMNI-S already surpasses most distilled baselines at a fraction of their computational cost. OMNI-B and OMNI-L push further, by being ranked second and first respectively, while being up to $10\times$ more efficient in GFLOPs than the best teacher models. Notably, OMNI-L achieves the best tile-level F1 (87.8%) and slide-level Bacc (63.5%), outperforming all teachers including UNI2-h (681M) and H-optimus-1 (1.1B) despite being significantly smaller. Both OMNI-B and OMNI-L also substantially outperform the concurrent multi-teacher distillation baseline GPFM across all metrics, with GPFM ranking below all OMNI variants despite using 303M parameters. To assess statistical significance, we performed paired Wilcoxon signed-rank tests between OMNI-L and all foundation models: OMNI-L outperforms significantly the large majority of teachers at both tile and slide level

Table 1: Linear probing performance at tile level (F1 %, 13 datasets), slide level (Bacc %, 7 CPTAC datasets / 25 tasks), and their mean. Mean rank is computed per dataset then averaged. * indicates significance of paired Wilcoxon signed-rank test vs. OMNI-L: $^{*}p < 0.05$; $^{**}p < 0.01$. **Bold**: best; underline: second best per column.

Model	Params.	Dim	GFLOPs	Tile-level F1 (%) 13 ds.	Tile-level Rank (↓) 13 ds.	Slide-level Bacc (%) 25 tasks	Slide-level Rank (↓) 7 datasets	Mean Score	Mean Rank (↓)
Foundation models									
UNI2-h [12]	681M	1536	180.4	87.4	4.6	61.5**	9.0	74.5	6.8
Virchow2 [80]	631M	1280	164.6	85.9	6.6	61.5*	8.3	73.7	7.4
H-optimus-1 [56]	1.1B	1536	296.0	86.3*	6.2	62.5	4.7	74.4	5.4
KEEP [78]	414M	1024	59.7	84.9*	9.5	60.7**	11.0	72.8	10.2
Prov-GigaPath [74]	1.1B	1536	223.4	84.2**	9.3	61.0**	10.7	72.6	10.0
CONCH 1.5 [41, 13]	307M	1024	79.54	84.3**	11.0	61.5*	9.4	72.9	10.2
Hibou-B [46]	86M	768	21.97	82.5**	10.7	60.3**	11.0	71.4	10.8
Kaiko-ViT-B/8 [1]	86M	768	66.86	82.0**	13.2	58.9**	13.9	70.5	13.6
Distilled models									
GPFM [42]	303M	1024	81.08	83.2**	9.5	60.9**	9.1	72.0	9.3
H0-mini [20]	86M	768	22.3	84.9**	8.3	61.5**	9.4	73.2	8.9
Vanilla MTD									
ViT-B single CLS	86M	768	21.97	85.4*	5.9	61.8**	8.7	73.6	7.3
Ours									
OMNI-T	7.3M	192	**1.39**	83.9	11.0	63.2	4.0	73.6	7.5
OMNI-S	28.7M	384	5.48	85.3	7.6	62.9	4.1	74.1	5.8
OMNI-B	114M	768	21.8	87.1	4.0	63.2	4.4	75.2	4.2
OMNI-L	330M	1024	67.45	**87.8**	**2.5**	**63.5**	**2.1**	**75.7**	**2.3**



(a) (b) (c)

Figure 3: **Analysis of teacher representations**. **(a)** Cosine reconstruction quality of UNI2-H as the number of teachers increases. As teachers compete for a single shared token, vanilla MTD reconstruction quality degrades significantly, whereas our approach is substantially more robust to the growing number of teachers. **(b)** UMAP of the $M{=}10$ teacher-specific tokens from OMNI-B on $3k$ images. Well-separated clusters confirm each token specializes toward its assigned teacher. **(c)** Smooth latent interpolation between two teacher-specific tokens (averaged over all teacher pairs), showing a consistent X-shaped crossover pattern as the conditioning vector is linearly interpolated.

($^{*}p < 0.05$; $^{**}p < 0.01$). Additional results on the THUNDER benchmark for the knn and few-shot classification tasks are presented in the appendix (Section A, Table 8).

Gains when combined with a stronger aggregator — In addition to lightweight mean-pooling, we also evaluate the performance of our method with ABMIL [27] aggregation on the BRACS dataset. Table 2 shows that OMNI also performs well when coupled with ABMIL. We also experimented with optimizing the weights of all teacher-specific tokens. Instead of uniformly averaging teacher tokens, learning teacher-specific weights increases performance, leading to our method ranking first.

Comparison with GPFM — We have compared our model to the publicly-released GPFM model in Table 1. Additionally, we pre-train OMNI-L (to match the number of parameters of GPFM ViT-L) on the same set of 3 teachers used by GPFM (UNI, Prov-GigaPath, PHIKON). The only difference is thus about the used training dataset, but one should note that our dataset is a small subset of the one used to pre-train GPFM. As seen in Table 4, when trained on the same teachers, a smaller dataset and with the same parameter size, we outperform the pre-trained GPFM model.

Table 2: ABMIL slide-level results on BRACS (7-class subtyping). Balanced accuracy (%), mean ± std across 5 folds. †Weighted mean pooling, where a scalar weight per teacher token is learned jointly with the ABMIL aggregator on the downstream task.

Model	BACC (%) (↑)
UNI2-h [12]	47.5 ± 1.7
Virchow2 [80]	44.1 ± 0.8
H-optimus-1 [56]	41.6 ± 2.3
KEEP [78]	46.9 ± 1.7
H0-mini [20]	44.5 ± 2.1
Prov-GigaPath [74]	41.5 ± 1.6
CONCH 1.5 [41, 13]	46.2 ± 1.0
GPFM [42]	43.5 ± 2.2
Vanilla MTD	45.7 ± 1.1
OMNI-T (ours)	$\underline{49.2 \pm 1.6}$
OMNI-T† (ours)	$\mathbf{49.9 \pm 2.0}$
OMNI-S (ours)	47.2 ± 0.9
OMNI-S† (ours)	46.0 ± 1.1
OMNI-B (ours)	47.2 ± 2.7
OMNI-B† (ours)	$\mathbf{49.9 \pm 3.0}$
OMNI-L (ours)	45.2 ± 1.2
OMNI-L† (ours)	46.3 ± 1.2

Table 3: Adversarial robustness (mean F1 drop, %) under PGD (ε=1.5×10^{-3}) and calibration (mean ECE, %), averaged across 13 datasets.

Model	F1 drop (↓)	ECE (↓)
UNI2-h [12]	30.4	4.0
Virchow2 [80]	29.4	3.8
H-optimus-1 [56]	58.1	4.0
KEEP [78]	43.9	4.4
H0-mini [20]	33.9	4.3
Kaiko-ViT-B/8 [1]	39.4	3.5
Prov-GigaPath [74]	40.8	3.7
CONCH 1.5 [41, 13]	77.1	4.7
OMNI-T (ours)	31.6	<u>3.4</u>
OMNI-S (ours)	22.3	3.9
OMNI-B (ours)	<u>20.1</u>	3.7
OMNI-L (ours)	**16.1**	**2.8**

Table 4: Comparison to GPFM [42], with both models trained on the same 3 teachers. F1 (%) averaged across 13 THUNDER tasks.

Model	F1 (%) (↑)
GPFM [42]	83.2
OMNI-L (ours)	**85.5**

Comparison with natural-domain MTD methods — We consider two recent state-of-the-art multi-teacher distillation approaches, namely UNIC [57] and DUNE [58], and apply their main contributions that could be integrated into our framework and have the potential to also improve the scaling abilities of the vanilla baseline. In particular, we evaluate the gains coming from the teacher dropping strategy in UNIC and the Transformer-based decoder architecture in DUNE. Table 5 shows that while they improve the downstream performance of the vanilla baseline, the gains are notably lower than the ones brought by our method. Additionally, adding them on top of OMNI does not bring further improvement, showing that such design choices are not needed along with our method.

Table 5: Main contributions of natural-domain MTD methods (1) UNIC and (2) DUNE, applied on top of vanilla MTD or OMNI-B. F1 averaged over 13 THUNDER tasks.

Model	F1 (%) (↑)
Vanilla MTD	85.4
+ Teacher drop. [57] (1)	85.6
+ Transf. decoder [58] (2)	85.9
OMNI-B (ours)	**87.1**
+ (1) + (2)	<u>87.0</u>

Uncertainty and robustness — Table 3 reports the robustness to adversarial attacks and the calibration of linear probes trained on top of embeddings from various foundation models, including ours, following the THUNDER benchmark [45]. OMNI-L is both the most robust to adversarial attacks, with an F1 drop of 16.1% compared with 29.4% for the most robust teacher (Virchow2), and the best calibrated model, with an ECE of 2.8%. OMNI-B and OMNI-S are also more robust than all teacher models. This could be explained by our diverse teacher-specific representations which, when fused, lead to a more robust representation.

Ablation studies — Table 6 ablates the importance of our proposed attention masking (*OMNI-B* vs *No attn. mask*), MoE blocks (*OMNI-B* vs *No MoE tail*), and infoNCE loss (*OMNI-B* vs *No InfoNCE loss*). When removing each contribution, performance drops compared with *OMNI-B*, showing that all of them are necessary. All variants match or outperform on average *Vanilla MTD* and the combination of all contributions reaches the highest performance.

Structured attention masking — Our current design does not fully isolate teacher tokens from each other as information can be passed implicitly through multiple layers. However, we believe that such a formulation already serves the purpose of specializing different teacher tokens. Figure 3b indeed shows that despite a lack of an explicit full isolation, it leads to disentangled teacher-specific representations. The usefulness of our attention mask formulation is empirically showcased in Table 6

Table 6: Ablation study on the contribution of the structured attention, InfoNCE loss, and MoE blocks. A ✗ indicates that the corresponding component is *removed* from the full OMNI-B model. Experiments are conducted on ViT-Base.

Model	InfoNCE	MoE	Attn. mask	Tile (F1)	Slide (Bacc)	Average
OMNI-B (ours)				**87.1**	**63.2**	**75.2**
No InfoNCE loss	✗			86.6 (-0.5)	61.1 (-2.1)	73.9 (-1.3)
No MoE tail		✗		86.6 (-0.5)	61.2 (-2.0)	73.9 (-1.3)
No attn. mask			✗	86.2 (-0.9)	61.6 (-1.6)	73.9 (-1.3)
Vanilla MTD	✗	✗	✗	85.4 (-1.7)	61.8 (-1.4)	73.6 (-1.6)

Table 7: Effect of full isolation of teacher tokens. F1 (%) averaged across 13 THUNDER tasks.

Model	F1
OMNI-B	**87.1**
+ full iso.	86.7 (-0.4)

highlighting the drop in downstream performance if replacing it with vanilla self-attention between all tokens. We conduct an experiment where patch tokens can only attend to each other, this time having a full isolation. Table 7 reports the performance of such a baseline and, as can be seen, a stricter full isolation does not seem to bring further improvement and might not be necessary.

4.3 Analysis of per-teacher representations

Teacher token specialization and scaling — Figure 3a shows that vanilla MTD suffers a steep degradation in reconstruction quality as the number of teachers grows, dropping from 0.90 cosine similarity at 1 teacher to 0.83 at 10. Our approach significantly mitigates this collapse, retaining 0.87 at 10 teachers. This demonstrates that the teacher-specific tokens with structured attention masking preserve individual reconstruction quality even under heavy multi-teacher pressure. The UMAP visualization in Figure 3b further confirms that each teacher token specializes, with well-separated clusters across all 10 teacher slots, rather than collapsing into a common average embedding.

Latent space geometry — Figure 3c reveals a smooth and monotonic crossover when linearly interpolating two teacher tokens. It indeed shows the evolution of the average alignment (CKA similarity) between ground-truth teacher embeddings and the embedding resulting from the linear interpolation of two teacher-specific tokens as a function of their respective weights. As can be seen, the average CKA similarity to the source decreases steadily while similarity to the target increases, with a crossover near $\alpha \approx 0.55$. This indicates that the student model has learned a geometrically well-structured latent space in which teacher representations are continuously interpolable.

5 Conclusion

Multi-teacher distillation is a powerful framework to train efficient encoders from the supervision of diverse foundation models. In this paper, we show that vanilla MTD is not scaling well to an increasing number of teachers. We thus proposed a new per-teacher representation based on key design choices, namely learnable teacher tokens, a teacher-specific attention scheme, MoE layers and a per-teacher infoNCE contrastive loss. These contributions allowed us to reach state-of-the-art downstream performance and outperform a large set of foundation models, while being up to $10\times$ more efficient and with many fewer parameters. This led us to release a collection of efficient models called **OMNI** with different sizes, i.e., OMNI-T, OMNI-S, OMNI-B, OMNI-L. We believe that such lightweight and powerful vision encoders are well suited for clinical applications requiring high-quality feature extraction at a low compute cost.

Limitations — Thanks to the complementarity of our teacher-specific representations, we rely on a simple mean-pooling aggregation mechanism. While we have explored optimizing a task-specific weighting of teachers on the BRACS dataset, this direction has yet to be studied more. Additionally, despite using the standard MTD patch loss, we did not explore the learning of teacher-specific dense representations. Both research directions are left for future work. Despite our empirical validation being composed of 39 tasks spanning diverse public datasets, organs, magnifications and institutions, the clinical use of our model would require further validation.

Broader impact — New foundation models have the potential to benefit positively clinical decisions and ultimately patient outcomes. However, such a clinical deployment should follow regulatory guidelines that come with careful validations and important considerations of ethical challenges.

Acknowledgments — This work has been supported by the *Agence Nationale de la Recherche* through *ANR-23-IAHU-0002* (*IHU PRISM*), *ANR-21-CE45-0007*, *ANR-23-CE45-0029*, *ANR-23-IACL-0003* (*DATAIA CLUSTER*) and the *Health Data Hub* as part of the second edition of the *France-Québec* call for projects *Intelligence Artificielle en santé*. This work was performed using HPC resources from *GENCI-IDRIS* (Grant *2025-AD011015593R1*). This research was also partially supported by NSF grants IIS-2123920, IIS-2212046.

References

[1] Nanne Aben, Edwin D de Jong, Ioannis Gatopoulos, Nicolas Känzig, Mikhail Karasikov, Axel Lagré, Roman Moser, Joost van Doorn, Fei Tang, et al. Towards large-scale training of pathology foundation models. *arXiv*, 2024.

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35: 23716–23736, 2022.

[3] Saghir Alfasly, Peyman Nejat, Sobhan Hemati, Jibran Khan, Isaiah Lahr, Areej Alsaafin, Abubakr Shafique, Nneka Comfere, Dennis Murphree, Chady Meroueh, et al. Foundation models for histopathology—fanfare or flair. *Mayo Clinic Proceedings: Digital Health*, 2024.

[4] Saghir Alfasly, Ghazal Alabtah, Sobhan Hemati, Krishna Rani Kalari, Joaquin J Garcia, and HR Tizhoosh. Validation of histopathology foundation models through whole slide image retrieval. *Scientific Reports*, 2025.

[5] Guilherme Aresta, Teresa Araújo, Scotty Kwok, Sai Saketh Chennamsetty, Mohammed Safwan, Varghese Alex, Bahram Marami, Marcel Prastawa, Monica Chan, Michael Donovan, et al. Bach: Grand challenge on breast cancer histology images. *Medical image analysis*, 2019.

[6] Varun Belagali, Saarthak Kapse, Pierre Marza, Srijan Das, Zilinghan Li, Sofiène Boutaj, Pushpak Pati, Srikar Yellapragada, Tarak Nath Nandi, Ravi K Madduri, et al. Ticon: A slide-level tile contextualizer for histopathology representation learning. *arXiv*, 2025.

[7] Nadia Brancati, Anna Maria Anniciello, Pushpak Pati, Daniel Riccio, Giosuè Scognamiglio, Guillaume Jaume, Giuseppe De Pietro, Maurizio Di Bonito, Antonio Foncubierta, Gerardo Botti, et al. Bracs: A dataset for breast carcinoma subtyping in h&e histology images. *Database*, 2022.

[8] Jack Breen, Katie Allen, Kieran Zucker, Lucy Godson, Nicolas M Orsi, and Nishant Ravikumar. A comprehensive evaluation of histopathology foundation models for ovarian cancer subtype classification. *NPJ Precision Oncology*, 2025.

[9] Otso Brummer, Petri Pölönen, Satu Mustjoki, and Oscar Brück. Computational textural mapping harmonises sampling variation and reveals multidimensional histopathological fingerprints. *British Journal of Cancer*, 2023.

[10] Gabriele Campanella, Shengjia Chen, Manbir Singh, Ruchika Verma, Silke Muehlstedt, Jennifer Zeng, Aryeh Stock, Matt Croken, Brandon Veremis, Abdulkadir Elmas, et al. A clinical benchmark of public self-supervised pathology foundation models. *Nature Communications*, 2025.

[11] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.

[12] Richard J Chen, Tong Ding, Ming Y Lu, Drew FK Williamson, Guillaume Jaume, Andrew H Song, Bowen Chen, Andrew Zhang, Daniel Shao, Muhammad Shaban, et al. Towards a general-purpose foundation model for computational pathology. *Nature Medicine*, 2024.

[13] Tong Ding, Sophia J Wagner, Andrew H Song, Richard J Chen, Ming Y Lu, Andrew Zhang, Anurag J Vaidya, Guillaume Jaume, Muhammad Shaban, Ahrong Kim, et al. Multimodal whole slide foundation model for pathology. *arXiv*, 2024.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv*, 2020.

[15] Nathan J Edwards, Mauricio Oberti, Ratna R Thangudu, Shuang Cai, Peter B McGarvey, Shine Jacob, Subha Madhavan, and Karen A Ketchum. The cptac data portal: a resource for cancer proteomics research. *Journal of proteome research*, 2015.

[16] Zhiyuan Fang, Jianfeng Wang, Lijuan Wang, Lei Zhang, Yezhou Yang, and Zicheng Liu. Seed: Self-supervised distillation for visual representation. In *ICLR*, 2021.

[17] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *JMLR*, 2022.

[18] Alexandre Filiot, Ridouane Ghermi, Antoine Olivier, Paul Jacob, Lucas Fidon, Axel Camara, Alice Mac Kain, Charlie Saillard, and Jean-Baptiste Schiratti. Scaling self-supervised learning for histopathology with masked image modeling. *medRxiv*, 2023.

[19] Alexandre Filiot, Paul Jacob, Alice Mac Kain, and Charlie Saillard. Phikon-v2, a large and public feature extractor for biomarker prediction. *arXiv*, 2024.

[20] Alexandre Filiot, Nicolas Dop, Oussama Tchita, Auriane Riou, Rémy Dubois, Thomas Peeters, Daria Valter, Marin Scalbert, Charlie Saillard, Geneviève Robin, et al. Distilling foundation models for robust and efficient models in digital pathology. In *MICCAI*, 2025.

[21] Ioannis Gatopoulos, Nicolas Känzig, Roman Moser, Sebastian Otálora, et al. eva: Evaluation framework for pathology foundation models. In *MIDL*, 2024.

[22] Christian Grashei, Christian Brechenmacher, Rao Muhammad Umer, Jingsong Liu, Carsten Marr, Ewa Szczurek, and Peter J Schüffler. Pathryoshka: Compressing pathology foundation models via multi-teacher knowledge distillation with nested embeddings. *arXiv preprint arXiv:2511.23204*, 2025.

[23] Fredrik K Gustafsson and Mattias Rantalainen. Evaluating computational pathology foundation models for prostate cancer grading under distribution shifts. *arXiv*, 2024.

[24] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[25] Zhi Huang, Federico Bianchi, Mert Yuksekgonul, Thomas J Montine, and James Zou. A visual–language foundation model for pathology image analysis using medical twitter. *Nature medicine*, 2023.

[26] Wisdom Ikezogwo, Saygin Seyfioglu, Fatemeh Ghezloo, Dylan Geva, Fatwir Sheikh Mohammed, Pavan Kumar Anand, Ranjay Krishna, and Linda Shapiro. Quilt-1m: One million image-text pairs for histopathology. *NeurIPS*, 2023.

[27] Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *ICML*, 2018.

[28] Guillaume Jaume, Paul Doucet, Andrew Song, Ming Yang Lu, Cristina Almagro Pérez, Sophia Wagner, Anurag Vaidya, Richard Chen, Drew Williamson, Ahrong Kim, et al. Hest-1k: A dataset for spatial transcriptomics and histology image analysis. *NeurIPS*, 2024.

[29] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *ECCV*, 2022.

[30] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

[31] Mingu Kang, Heon Song, Seonwook Park, Donggeun Yoo, and Sérgio Pereira. Benchmarking self-supervised learning on diverse pathology datasets. In *CVPR*, 2023.

[32] Mikhail Karasikov, Joost van Doorn, Nicolas Känzig, Melis Erdal Cesur, Hugo Mark Horlings, Robert Berke, Fei Tang, and Sebastian Otálora. Training state-of-the-art pathology foundation models with orders of magnitude less data. *arXiv*, 2025.

[33] Jakob Nikolas Kather, Niels Halama, and Alexander Marx. 100,000 histological images of human colorectal cancer and healthy tissue. *Zenodo*, 2018.

[34] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv*, 2024.

[35] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *ICML*, 2021.

[36] Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, et al. Matryoshka representation learning. *Advances in Neural Information Processing Systems*, 35: 30233–30249, 2022.

[37] Jaeung Lee, Jeewoo Lim, Keunho Byeon, and Jin Tae Kwak. Benchmarking pathology foundation models: Adaptation strategies and scenarios. *Computers in Biology and Medicine*, 2025.

[38] Dong Li, Guihong Wan, Xintao Wu, Xinyu Wu, Ajit J Nirmal, Christine G Lian, Peter K Sorger, Yevgeniy R Semenov, and Chen Zhao. A survey on computational pathology foundation models: Datasets, adaptation strategies, and evaluation tasks. *arXiv*, 2025.

[39] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[40] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *ACL*, 2022.

[41] Ming Y Lu, Bowen Chen, Drew FK Williamson, Richard J Chen, Ivy Liang, Tong Ding, Guillaume Jaume, Igor Odintsov, Long Phi Le, Georg Gerber, et al. A visual-language foundation model for computational pathology. *Nature Medicine*, 2024.

[42] Jiabo Ma, Zhengrui Guo, Fengtao Zhou, Yihui Wang, Yingxue Xu, Jinbang Li, Fang Yan, Yu Cai, Zhengjie Zhu, Cheng Jin, et al. Towards a generalizable pathology foundation model via unified knowledge distillation. *arXiv*, 2024.

[43] Jiabo Ma, Yingxue Xu, Fengtao Zhou, Yihui Wang, Cheng Jin, Zhengrui Guo, Jianfeng Wu, On Ki Tang, Huajun Zhou, Xi Wang, et al. Pathbench: A comprehensive comparison benchmark for pathology foundation models towards precision oncology. *arXiv*, 2025.

[44] Roba Al Majzoub, Hashmat Malik, Muzammal Naseer, Zaigham Zaheer, Tariq Mahmood, Salman Khan, and Fahad Khan. How good is my histopathology vision-language foundation model? a holistic benchmark. *arXiv*, 2025.

[45] Pierre Marza, Leo Fillioux, Sofiène Boutaj, et al. THUNDER: Tile-level histopathology image understanding benchmark. *Neural Information Processing Systems (NeurIPS) D&B Track*, 2025.

[46] Dmitry Nechaev, Alexey Pchelnikov, and Ekaterina Ivanova. Hibou: A family of foundational vision transformers for pathology. *arXiv*, 2024.

[47] Dmitry Nechaev, Alexey Pchelnikov, and Ekaterina Ivanova. Spider: a comprehensive multi-organ supervised pathology dataset and baseline models. *arXiv*, 2025.

[48] Peter Neidlinger, Omar SM El Nahhas, Hannah Sophie Muti, Tim Lenz, Michael Hoffmeister, Hermann Brenner, Marko van Treeck, Rupert Langer, Bastian Dislich, Hans Michael Behrens, et al. Benchmarking foundation models as feature extractors for weakly-supervised computational pathology. *arXiv*, 2024.

[49] Pedro C Neto, Diana Montezuma, Sara P Oliveira, Domingos Oliveira, João Fraga, Ana Monteiro, João Monteiro, Liliana Ribeiro, Sofia Gonçalves, Stefan Reinhard, et al. An interpretable machine learning system for colorectal cancer diagnosis from pathology slides. *NPJ precision oncology*, 2024.

[50] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv*, 2018.

[51] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv*, 2023.

[52] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[53] Mike Ranzinger, Jon Barker, Greg Heinrich, Pavlo Molchanov, Bryan Catanzaro, and Andrew Tao. Phi-s: Distribution balancing for label-free multi-teacher distillation, 2024. URL `https://arxiv.org/abs/2410.01680`.

[54] Mike Ranzinger, Greg Heinrich, Jan Kautz, and Pavlo Molchanov. Am-radio: Agglomerative vision foundation model reduce all domains into one. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12490–12500, June 2024.

[55] Charlie Saillard, Rodolphe Jenatton, Felipe Llinares-López, Zelda Mariet, David Cahané, Eric Durand, and Jean-Philippe Vert. H-optimus-0, 2024. URL `https://github.com/bioptimus/releases/tree/main/models/h-optimus/v0`.

[56] Charlie Saillard, Rodolphe Jenatton, Felipe Llinares-López, Zelda Mariet, David Cahané, Eric Durand, and Jean-Philippe Vert. H-optimus-1, 2025. URL `https://huggingface.co/bioptimus/H-optimus-1`.

[57] Mert Bülent Sarıyıldız, Philippe Weinzaepfel, Thomas Lucas, Diane Larlus, and Yannis Kalantidis. Unic: Universal classification models via multi-teacher distillation. In *ECCV*, 2024.

[58] Mert Bülent Sarıyıldız, Philippe Weinzaepfel, Thomas Lucas, Pau De Jorge, Diane Larlus, and Yannis Kalantidis. Dune: Distilling a universal encoder from heterogeneous 2d and 3d teachers. In *CVPR*, 2025.

[59] George Shaikovski, Adam Casson, Kristen Severson, Eric Zimmermann, Yi Kan Wang, Jeremy D Kunz, Juan A Retamero, Gerard Oakley, David Klimstra, Christopher Kanan, et al. Prism: A multi-modal generative foundation model for slide-level histopathology. *arXiv*, 2024.

[60] Jinghuan Shang, Karl Schmeckpeper, Brandon B May, Maria Vittoria Minniti, Tarik Kelestemur, David Watkins, and Laura Herlant. Theia: Distilling diverse vision foundation models for robot learning. *arXiv preprint arXiv:2407.20179*, 2024.

[61] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv*, 2017.

[62] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv*, 2025.

[63] Fabio A Spanhol, Luiz S Oliveira, Caroline Petitjean, and Laurent Heutte. A dataset for breast cancer histopathological image classification. *IEEE transactions on biomedical engineering*, 2015.

[64] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP*, 2019.

[65] Yuri Tolkach, Lisa Marie Wolgast, Alexander Damanakis, Alexey Pryalukhin, Simon Schallenberg, Wolfgang Hulla, Marie-Lisa Eich, Wolfgang Schroeder, Anirban Mukhopadhyay, Moritz Fuchs, et al. Artificial intelligence for tumour tissue detection and histological regression grading in oesophageal adenocarcinomas: a retrospective algorithm development and validation study. *The Lancet Digital Health*, 2023.

[66] Anurag Vaidya, Andrew Zhang, Guillaume Jaume, Andrew H Song, Tong Ding, Sophia J Wagner, Ming Y Lu, Paul Doucet, Harry Robertson, Cristina Almagro-Perez, et al. Molecular-driven foundation model for oncologic pathology. *arXiv*, 2025.

[67] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017.

[68] Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant cnns for digital pathology. In *MICCAI*, 2018.

[69] Eugene Vorontsov, Alican Bozkurt, Adam Casson, George Shaikovski, Michal Zelechowski, Kristen Severson, Eric Zimmermann, James Hall, Neil Tenenholtz, Nicolo Fusi, et al. A foundation model for clinical-grade computational pathology and rare cancers detection. *Nature medicine*, 2024.

[70] Xiyue Wang, Junhan Zhao, Eliana Marostica, Wei Yuan, Jietian Jin, Jiayu Zhang, Ruijiang Li, Hongping Tang, Kanran Wang, Yu Li, et al. A pathology foundation model for cancer diagnosis and prognosis prediction. *Nature*, 2024.

[71] Jerry Wei, Arief Suriawinata, Bing Ren, Xiaoying Liu, Mikhail Lisovsky, Louis Vaickus, Charles Brown, Michael Baker, Naofumi Tomita, Lorenzo Torresani, Jason Wei, and Saeed Hassanpour. A petri dish for histopathology image analysis. In *Artificial Intelligence in Medicine*, 2021.

[72] Georg Wölflein, Dyke Ferber, Asier R Meneghetti, Omar SM El Nahhas, Daniel Truhn, Zunamys I Carrero, David J Harrison, Ognjen Arandjelović, and Jakob Nikolas Kather. Benchmarking pathology feature extractors for whole slide image classification. *arXiv*, 2023.

[73] Jinxi Xiang, Xiyue Wang, Xiaoming Zhang, Yinghua Xi, Feyisope Eweje, Yijiang Chen, Yuchen Li, Colin Bergstrom, Matthew Gopaulchan, Ted Kim, et al. A vision–language foundation model for precision oncology. *Nature*, 2025.

[74] Hanwen Xu, Naoto Usuyama, Jaspreet Bagga, Sheng Zhang, Rajesh Rao, Tristan Naumann, Cliff Wong, Zelalem Gero, Javier González, Yu Gu, et al. A whole-slide foundation model for digital pathology from real-world data. *Nature*, 2024.

[75] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv*, 2022.

[76] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33:5824–5836, 2020.

[77] Andrew Zhang, Guillaume Jaume, Anurag Vaidya, Tong Ding, and Faisal Mahmood. Accelerating data processing and benchmarking of ai models for pathology. *arXiv*, 2025.

[78] Xiao Zhou, Luoyi Sun, Dexuan He, Wenbin Guan, Ruifen Wang, Lifeng Wang, Xin Sun, Kun Sun, Ya Zhang, Yanfeng Wang, et al. A knowledge-enhanced pathology vision-language foundation model for cancer diagnosis. *arXiv*, 2024.

[79] Beier Zhu, Yulei Niu, Yucheng Han, Yue Wu, and Hanwang Zhang. Prompt-aligned gradient for prompt tuning. In *ICCV*, 2023.

[80] Eric Zimmermann, Eugene Vorontsov, Julian Viret, Adam Casson, Michal Zelechowski, George Shaikovski, Neil Tenenholtz, James Hall, David Klimstra, Razik Yousfi, et al. Virchow2: Scaling self-supervised mixed magnification models in pathology. *arXiv*, 2024.

Appendix

A Additional results

knn and few-shot classification — Table 8 reports knn and 16-shot few-shot classification results on the THUNDER benchmark [45] across 13 datasets. OMNI-B and OMNI-L achieve the best mean rank on knn, and OMNI-L ranks second on 16-shot few-shot classification, behind UNI2-h. Both models remain significantly more efficient than the top teacher models, with OMNI-B requiring about $8\times$ fewer GFLOPs than UNI2-h.

Robustness to the teacher ordering — In Figure 1b, teachers are ranked by their individual performance on the THUNDER benchmark and added from strongest to weakest. To verify that the observed scaling trend does not depend on this ordering, we repeat the experiment with the reverse order, adding teachers from weakest to strongest. Results are reported in Figure 4. In this setting, both methods benefit from the stronger teachers added at each step. However, vanilla MTD quickly saturates, gaining only 0.5 F1 points when going from 3 to 10 teachers. In contrast, our method keeps improving as more teachers are added, and the gap between the two methods widens with the number of teachers, from 0.5 points with 3 teachers to 1.7 points with 10 teachers. This confirms that the better scaling of our method holds regardless of the order in which teachers are integrated.

Table 8: KNN and 16-shot few-shot classification results (F1 %, 13 datasets). Mean rank is computed per dataset then averaged. **Bold**: best; underline: second best per column.

				KNN		16-shot	
Model	**Params.**	**Dim**	**GFLOPs**	**F1 (%)** 13 ds.	**Rank** (↓) 13 ds.	**F1 (%)** 13 ds.	**Rank** (↓) 13 ds.
UNI2-h	681M	1536	180.4	**85.0**	4.7	**81.9**	**3.4**
Virchow2	631M	1280	164.6	84.7	5.0	75.9	9.6
H-optimus-1	1.1B	1536	296.0	83.9	5.3	78.3	7.1
KEEP	414M	1024	59.7	83.4	7.7	79.3	7.4
Prov-GigaPath	1.1B	1536	223.4	81.3	9.1	77.2	8.3
CONCH 1.5	307M	1024	79.54	82.0	9.4	77.2	9.1
Kaiko-ViT-B/8	86M	768	66.86	80.0	10.5	78.6	7.5
Distilled models							
GPFM	303M	1024	81.08	82.2	7.9	77.8	7.2
H0-mini	86M	768	22.3	81.4	8.4	76.6	8.8
OMNI-T (ours)	7.3M	192	1.39	81.5	8.8	77.5	8.1
OMNI-S (ours)	28.7M	384	5.48	82.3	6.9	78.7	6.0
OMNI-B (ours)	114M	768	21.8	84.3	**3.6**	79.9	4.8
OMNI-L (ours)	330M	1024	67.45	<u>84.8</u>	**3.6**	<u>80.7</u>	<u>3.7</u>

Additional slide-level evaluation on IMP — To further increase the diversity of our evaluation, we also benchmark our method on the IMP colorectal cancer dataset [49] from PathoBench, using an ABMIL aggregator [27]. As the IMP dataset comprises around $5{,}000$ WSIs totaling more than 5TB, embedding it with every model requires significant compute. We thus restrict this evaluation to a subset of representative and competitive baselines. As shown in Table 9, OMNI-B outperforms the vanilla MTD baseline and H0-mini, and matches the state-of-the-art teacher foundation model UNI2-h while requiring about $8\times$ fewer GFLOPs.

B MoE routing statistics

MoE layers are introduced to give the student more flexibility in how it processes the different token types, in particular the $M{=}10$ teacher-specific tokens. This flexibility is only useful if the router actually assigns different experts to different tokens, rather than collapsing onto a single expert pair. We therefore analyze the routing decisions of our model on 5K randomly sampled training tiles.

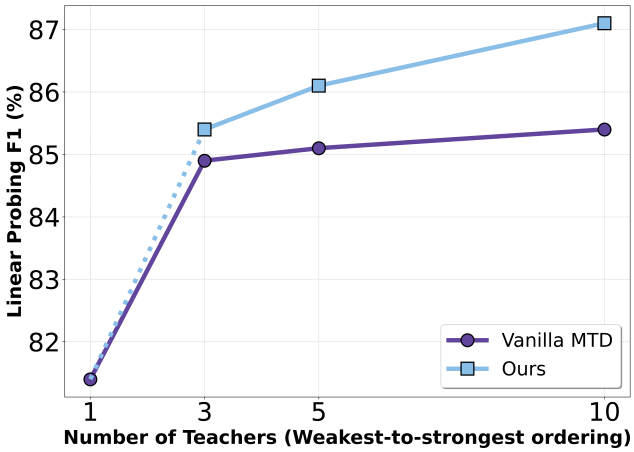


Figure 4: **Scaling with a reverse teacher ordering.** Tile-level linear probing F1 (%) averaged across 13 THUNDER tasks as a function of the number of teachers, when adding teachers from weakest to strongest. Our method keeps improving as teachers are added, while vanilla MTD saturates.

Table 9: ABMIL slide-level classification on the IMP colorectal dataset (PathoBench). Balanced accuracy (%) with 95% CI and efficiency (GFLOPs).

Model	GFLOPs	Bacc (CI)
H0-mini [20]	22.3	94.0 (92.3–95.4)
UNI2-h [12]	180.4	**95.0** (94.0–96.5)
Vanilla MTD	21.97	93.7 (92.3–95.5)
OMNI-B (ours)	21.8	<u>94.9</u> (93.3–96.1)

Protocol — For each image and each of the three MoE layers, we record the ordered top-2 expert set selected for every token. With $E{=}5$ experts, there are 20 possible ordered sets. We then compute the largest number of tokens that share the same set, separately for teacher tokens and patch tokens. A fully collapsed router would assign the same set to all tokens. For reference, if the 20 sets were drawn uniformly at random, the largest group would typically contain 2–3 of the 10 teacher tokens, and about 8% of the 256 patch tokens.

Teacher tokens — Figure 5a reports the distribution of this maximum over images. In the first two MoE layers, at most 2 or 3 teacher tokens share the same expert set in more than 96% of images. This is close to the random-assignment reference, so routing is highly diverse across teachers. The last MoE layer shows more sharing, with a mode at 6 teacher tokens (78.7% of images). Routing nonetheless never collapses, as no image has all 10 teacher tokens on the same expert set. This increased sharing in the deepest layer suggests that groups of related teachers are handled by common experts close to the output.

Patch tokens — Figure 5b reports the same statistic for patch tokens, binned as a percentage of the 256 patches per image. In all layers, the most shared expert set covers only 10–30% of patches, far from the collapse regime (90–100%). In the first layer, 97.2% of images fall in the 10–20% bin. Sharing increases slightly with depth, and in the last layer the 20–30% bin holds 53.8% of images.

Overall, both teacher and patch tokens are routed to diverse sets of experts, and the load-balancing loss $\mathcal{L}_{\text{MoE}}$ prevents the router from collapsing.

C Causes of the scaling failure

Besides the single-CLS bottleneck, optimization difficulty and conflicts between teachers could also partially explain the scaling failure of vanilla MTD. We conducted experiments to probe the impact of these two causes, and did not detect any conflict between the supervision coming from different

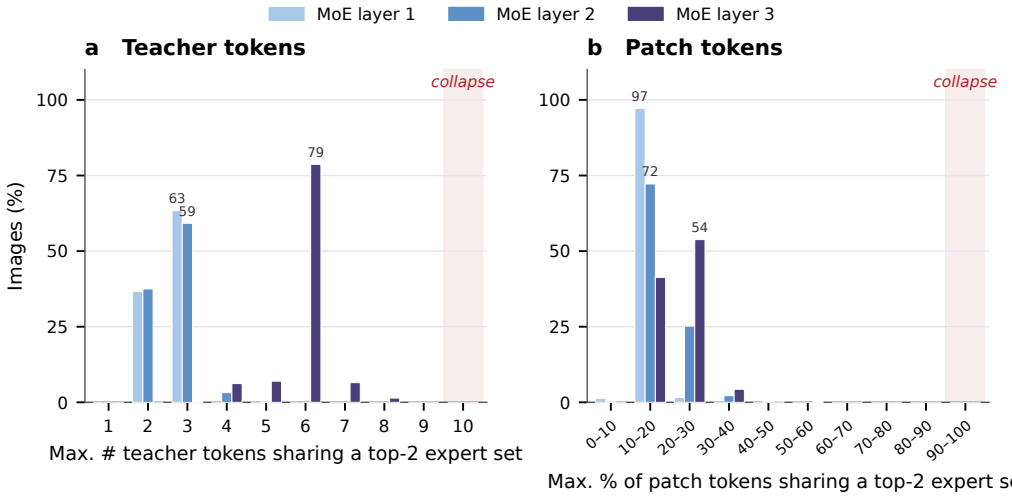


Figure 5: **MoE routing diversity** of OMNI-B over 5K random training tiles (top-2 routing, E=5). **(a)** Fraction of images by the maximum number of teacher tokens sharing the same ordered top-2 expert set. **(b)** Fraction of images by the maximum percentage of patch tokens sharing the same ordered top-2 expert set. Shaded bins mark full routing collapse.

teachers or any optimization instability in our context of digital pathology foundation model teachers. We detail these experiments below.

Teacher conflicts — To investigate potential teacher conflicts, we adapt the gradient conflict analysis commonly used in multi-task learning [76] to the MTD setting. We quantify conflict by computing the gradient of each teacher's individual loss with respect to the student parameters. We then calculate the average pairwise cosine similarity between each teacher's gradient vector and those of the rest of the pool, where negative values would indicate actively opposing update directions. As shown in Table 10, training the vanilla MTD baseline on our pathology teachers does not produce major gradient conflicts. Instead, the update directions remain positively aligned across the ten teachers, with a mean cosine similarity of 0.162.

Table 10: Gradient conflict analysis for vanilla MTD (across 500k iterations).

Teacher	Mean grad. cosine w/ others	Min	Max
UNI2-h	0.229	0.049	0.418
Virchow2	0.137	0.035	0.206
H-optimus-1	0.225	0.046	0.423
Prov-GigaPath	0.201	0.050	0.344
Kaiko-ViT-B/8	0.135	0.039	0.209
Hibou-L	0.168	0.062	0.250
H0-mini	0.233	0.090	0.423
KEEP	0.149	0.043	0.216
DINOv3-ViT-L/16	0.050	0.035	0.090
CONCH1.5	0.094	0.039	0.155
Overall	**0.162**	**0.035**	**0.423**

Optimization difficulty — To probe for optimization difficulty or training instability, we tracked the reconstruction loss throughout training. If optimization difficulties were the primary cause of the scaling failure, we would expect highly volatile loss curves, spikes, or an inability to converge. Instead, as shown in Figure 6a (sampled across 700k training steps for the N=10 case), the loss curves of both vanilla MTD and our method are smooth and stable. The loss of vanilla MTD converges faster, but to a higher value.

Single-CLS bottleneck — Finally, to link the performance degradation to the CLS token bottleneck, we analyze the converged loss as a function of the number of teachers N. As shown in Figure 6b, when scaling from 1 to 10 teachers, the converged loss of vanilla MTD rapidly degrades, increasing from 0.104 to 0.175. A single CLS token lacks the representational capacity to simultaneously compress the diverse feature distributions of multiple teachers. By introducing teacher-specific tokens, OMNI reaches a lower converged loss for every number of teachers considered.

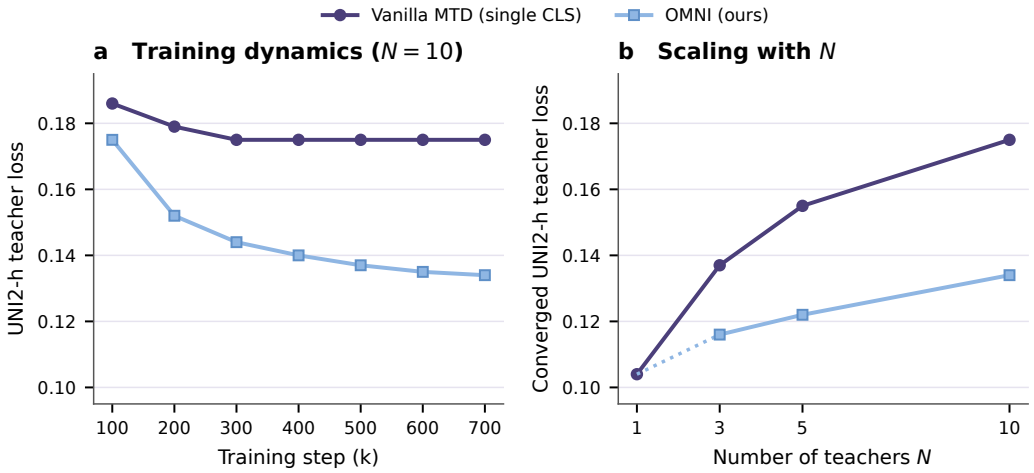


Figure 6: **UNI2-h teacher reconstruction loss. (a)** Loss during training with N=10 teachers. Both methods converge smoothly, with OMNI reaching a lower value. **(b)** Converged loss as a function of the number of teachers N. The single-CLS bottleneck of vanilla MTD degrades sharply as N grows. With a single teacher, both methods reduce to the same model.

D Hyperparameter sensitivity analysis

Table 11 presents a sensitivity analysis on the hyperparameter λ_{ctr}, where values 0, 0.1, 0.25, 0.5 are compared. As can be seen, removing the loss ($\lambda_{ctr} = 0$) leads to a decrease in performance (as already shown in our ablation study in Table 6), similarly to fixing a too high weight ($\lambda_{ctr} = 0.5$). Our method is however robust to moderate weight variations with the same performance for $\lambda_{ctr} = 0.1$ and $\lambda_{ctr} = 0.25$.

Table 11: Sensitivity to the contrastive loss weight λ_{ctr}. F1 (%) averaged across 13 THUNDER tasks.

$\lambda_{ctr} = 0$	$\lambda_{ctr} = 0.1$	$\lambda_{ctr} = 0.25$	$\lambda_{ctr} = 0.5$
86.6	**87.1**	**87.1**	86.8

E Patch loss details

Patch-grid interpolation for heterogeneous teacher tokenizations — In our multi-teacher setting (M=10), the number of patch tokens produced by teacher backbones can vary substantially (e.g., 7×7, 14×14, 16×16, 28×28), which prevents a direct one-to-one comparison between student and teacher patch embeddings. To handle this mismatch, we treat patch tokens as samples on a regular 2D grid and *resize* the student patch-token grid to the teacher grid using differentiable bilinear interpolation before applying the teacher-specific patch projector.

Let the student produce patch tokens $\mathbf{Z}_\theta^{\text{patch}}(d_j) \in \mathbb{R}^{N_s \times D_s}$ for sample d_j, and teacher i produce patch embeddings $\mathbf{Y}_i^{\text{patch}}(d_j) \in \mathbb{R}^{N_i \times D_i}$. We assume both N_s and N_i correspond to square grids, i.e., $N_s = H_s W_s$ and $N_i = H_i W_i$ with $H_s = W_s = \sqrt{N_s}$ and $H_i = W_i = \sqrt{N_i}$. We first reshape student tokens into a feature map $\mathbf{F}_\theta^s(d_j) \in \mathbb{R}^{D_s \times H_s \times W_s}$

Then, for each teacher i, we resize this feature map to the teacher grid using bilinear interpolation:

$$\mathbf{F}_\theta^{s\to i}(d_j) = \mathrm{Interp}_{H_i\times W_i}(\mathbf{F}_\theta^s(d_j)) \in \mathbb{R}^{D_s\times H_i\times W_i}. \tag{12}$$

Finally, we flatten back to obtain *teacher-aligned* student tokens $\bar{\mathbf{Z}}_\theta^{\mathrm{patch},i}(d_j) \in \mathbb{R}^{N_i\times D_s}$. Dense supervision is provided only on a subset of teacher positions $\mathcal{I}_i$ (precomputed sampled indices), so the patch-level distillation term becomes

$$\mathcal{L}_{\mathrm{MTD},i}^{\mathrm{patch}}(\theta,\psi_i,\bar{w}_i,d_j) = \frac{1}{|\mathcal{I}_i|}\sum_{u\in\mathcal{I}_i}\Phi\Big[p_{\bar{w}_i}^{\mathrm{patch}}\Big(\bar{\mathbf{z}}_{\theta,u}^{\mathrm{patch},i}(d_j)\Big),\ \mathbf{y}_{i,u}^{\mathrm{patch}}(d_j)\Big], \tag{13}$$

where $\bar{\mathbf{z}}_{\theta,u}^{\mathrm{patch},i}(d_j) \in \mathbb{R}^{D_s}$ denotes the u-th aligned student token on the grid of teacher i, $\mathbf{y}_{i,u}^{\mathrm{patch}}(d_j) \in \mathbb{R}^{D_i}$ the corresponding teacher patch embedding, and $p_{\bar{w}_i}^{\mathrm{patch}} : \mathbb{R}^{D_s} \to \mathbb{R}^{D_i}$ the linear projection head of teacher i, applied independently to each token. Φ is the cosine and smooth L1 loss defined in Section 3. In practice, several teachers share the same grid size N_i. We therefore compute $\bar{\mathbf{Z}}_\theta^{\mathrm{patch},i}(d_j)$ once per distinct grid size and reuse it across teachers for efficiency. The final patch loss of teacher i is obtained by averaging this term over the samples of the batch.

F Implementation details

We provide here additional implementation details for reproducibility. All models are trained from random initialization on 10M tiles extracted from 10K whole-slide images sourced from TCGA, spanning diverse tissue types and organ systems. Teacher embeddings are pre-computed and stored offline; teachers are kept fully frozen throughout student training. Table 12 summarizes the main hyperparameters for all OMNI variants.

G Downstream evaluation details

We provide here additional details on the downstream evaluation benchmarks used in this work.

Slide-level evaluation protocol — For slide-level evaluation, we follow the PathoBench protocol [77] on CPTAC cohorts, and report balanced accuracy (bacc) over a 50-fold train/test split. The full list of 25 tasks across 7 cancer types is provided in Table 13.

Mean pooling as the default aggregation choice — We use mean-pooling over teacher-specific tokens as our default aggregation method for two main reasons. First, it isolates the gains brought by our pre-training contributions from those of any additional downstream adaptation. This allows us to clearly measure the benefits of our teacher-specific formulation and its accompanying architectural choices, and to show that no extra downstream computation is needed to improve upon vanilla MTD. Second, mean-pooled embeddings can be computed once, offline, before training a task-specific probe. This is particularly interesting for MIL models: learning per-teacher weights would instead require storing all teacher-specific embeddings for every patch, multiplying storage requirements by $M{=}10$. It would also require loading these embeddings into GPU memory during training, which slows down training and increases memory constraints. Mean-pooling avoids both issues, which is why we adopt it as our default setting.

Our method combined with simple and efficient mean-pooling is thus sufficient to better scale to more teachers. Performance can nonetheless be further improved with a learned aggregation: as shown in Table 2, a weighted pooling of teacher-specific tokens brings additional gains on the BRACS dataset.

H Compute budget summary

Pre-training consists of two stages. First, we pre-computed and stored offline the embeddings of all 10 teacher models over the 10M-tile TCGA dataset, requiring approximately 400 A100 GPU hours in total. Each subsequent student pre-training run then took approximately 40 H100 GPU hours. For downstream evaluation, benchmarking on THUNDER and PathoBench required approximately 10,000 V100 GPU hours in total, the majority coming from tile feature extraction across the large number of whole-slide images for all foundation models.

Table 12: Training hyperparameters for OMNI models.

Hyperparameter	**OMNI-T**	**OMNI-S**	**OMNI-B**	**OMNI-L**
Optimization				
Optimizer	Adam			
Learning rate	1×10^{-4}			
Training iterations	700k			
Batch size	128			
Architecture				
Embedding dimension	192	384	768	1024
Depth	12	12	12	24
Attention heads	3	6	12	16
Parameters	7.3M	28.7M	114M	330M
MoE tail blocks	3	3	3	5
Number of experts E	5			
Top-k routing	$k = 2$			
MoE auxiliary loss coefficient α_{MoE}	0.05			
Image size	224×224			
Patch size	14×14			
Loss				
CLS loss	Cosine + SmoothL1 (β=1.0)			
SmoothL1 weight λ_{reg}	1.0			
Contrastive loss	InfoNCE			
Contrastive weight λ_{ctr}	0.1			
Contrastive temperature τ	0.1			
Patch loss	✓			
Data				
Pre-training dataset	TCGA			
Number of tiles	10M			
Number of WSIs	10k			
Number of teachers M	10			

Table 13: Slide-level tasks from the PathoBench benchmark [77] on CPTAC cohorts. All tasks are evaluated using balanced accuracy (BACC) with a 50-fold train/test split.

Cohort	**Cancer Type**	**Task**	**Task Type**
CPTAC-BRCA	Breast carcinoma	Immune class	Subtyping
		PIK3CA mutation	Mutation prediction
		TP53 mutation	Mutation prediction
CPTAC-CCRCC	Clear cell renal cell carcinoma	BAP1 mutation	Mutation prediction
		VHL mutation	Mutation prediction
CPTAC-COAD	Colon adenocarcinoma	ACVR2A mutation	Mutation prediction
		ARID1A mutation	Mutation prediction
		Immune class	Subtyping
		MSI-H	Biomarker prediction
		PIK3CA mutation	Mutation prediction
		SETD1B mutation	Mutation prediction
		TP53 mutation	Mutation prediction
CPTAC-GBM	Glioblastoma multiforme	EGFR mutation	Mutation prediction
		Immune class	Subtyping
		TP53 mutation	Mutation prediction
CPTAC-LSCC	Lung squamous cell carcinoma	ARID1A mutation	Mutation prediction
		Histologic grade	Grading
		Immune class	Subtyping
		KEAP1 mutation	Mutation prediction
CPTAC-LUAD	Lung adenocarcinoma	EGFR mutation	Mutation prediction
		KRAS mutation	Mutation prediction
		STK11 mutation	Mutation prediction
		TP53 mutation	Mutation prediction
CPTAC-UCEC	Uterine corpus endometrial carcinoma	CTNNB1 mutation	Mutation prediction
		PTEN mutation	Mutation prediction
Total		**25 tasks**	**7 cohorts**

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The methodological contributions and results presented in the abstract and introduction are respectively supported by Sections 3 and 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: Our conclusion (Section 5) ends with a paragraph discussing limitations.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A].

   Justification: We do not provide any theoretical results.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The considered datasets and tasks are presented in Section 4. All experiment details can be found in Section 4 and the appendix. We also used fully reproducible benchmarks in our experiments.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The code to reproduce our results along with pre-trained models will be publicly released upon acceptance.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Our experimental setup is detailed in Section 4 and in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We perform a paired Wilcoxon signed-rank test to evaluate the statistical significance of the performance differences between our method and concurrent baselines (Table 1). We also report standard deviations when reporting dataset-specific performance among different folds (Table 2). We also follow the guidelines of the benchmarks we report results from, i.e. THUNDER [45] and PathoBench [77] as we believe they employ standard evaluation protocols in the community when it comes to evaluating foundation models.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Refer to Appendix H.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We follow NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Our conclusion (Section 5) ends with a paragraph discussing the broader impact with our work (after limitations).

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: We expect all users to follow the guidelines accompanying our released models.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite the original papers and of all assets (code, data, models) used in the paper.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [No]

Justification: The code to reproduce our results along with pre-trained models will be publicly released upon acceptance. These will be accompanied with proper documentation.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.