TICON: A Slide-Level Tile Contextualizer for Histopathology Representation Learning

Supplementary Material

This technical appendix presents the following materials:

- Additional experiments and results (Sec. A)
  - Effect of no multi-target pretraining
  - Benchmarking against existing Slide Encoders
  - Ablation on TICON's Architecture
  - Extended Analysis on HEST-Bench
  - TransMIL as Aggregator in TANGLE Pretraining
  - Tangle vs Meanpool
  - CKA analysis for shared semantic space
- Additional implementation details (Sec. B)
  - TANGLE Setup
  - Harmonizing the Field of View for Tile Encoders
  - CATCH Data Curation
  - Evaluation setting
  - TICON's architecture and pretraining setting
- Additional visualizations -
  - Overview of TICON's multi-target versus single-target pretraining paradigms in Fig. 5.
  - Overview of TICON's inference modes in Fig. 6.

A. Additional experiments and results

1. **Effect of no multi-target pretraining:** We validate the effectiveness of our proposed Omni-Feature Masked Modeling (OFMM) objective in Table 6. Specifically, we compare our default TICON, trained to reconstruct embeddings from multiple diverse tile encoders (Multi-target), against a variant trained solely to reconstruct the input encoder's own features (Single-target). Fig. 5 describes the differences in multi-target and single-target pretraining. Both models are trained within the same unified pretraining framework. We observe that multi-target training generally outperforms the single-target baseline, surpassing it in 7 out of 12 comparisons across HEST-Bench and CATCH tasks. More importantly, TICON with multi-target prediction consistently improves upon the non-contextual (tile-encoder only) baseline in all 12 instances (12/12), whereas the single-target variant fails to do so in 2 cases (10/12). Furthermore, the overall state-of-the-art performance for each task is consistently achieved by TICON with the default multi-target objective. This confirms that compelling the model to predict varied semantic "views" of the same tile encourages the learning of a more robust and generalized contextual representation.
   Takeaway: *Our Omni-Feature Masked Modeling (OFMM) benefits from the Multi-target prediction objective, yielding more robust representations than Single-target prediction.*

Table 6. Comparison with no multi-target pretraining in TICON evaluated on tile-level tasks with slide context. PCC reported for HEST and F1-score for CATCH. Individual aggregator is pretrained with TANGLE prior to linear probing for slide-level tasks. $\boxed{\cdot}$ denotes the state-of-the-art on the respective tasks.

		Multi-target	H-optimus-1	UNI2-h	CONCHv1.5
Tile-level tasks	HEST	Tile-encoder only	0.422	0.414	0.379
		✗	0.422	0.415	**0.387**
		✓	$\boxed{\textbf{0.427}}$	**0.419**	0.381
	CATCH	Tile-encoder only	86.2	85.5	81.6
		✗	$\boxed{\textbf{87.6}}$	**87.1**	84.0
		✓	$\boxed{\textbf{87.6}}$	86.9	**84.9**
Slide-level tasks	BRACS	Tile-encoder only	60.4	53.8	60.6
		✗	58.4	**60.0**	62.4
		✓	**62.6**	58.9	$\boxed{\textbf{63.8}}$
	CPTAC	Tile-encoder only	68.7	69.5	71.6
		✗	72.1	**70.7**	72.4
		✓	**72.5**	70.3	$\boxed{\textbf{72.7}}$

2. **Benchmarking against existing Slide Encoders:** In Table 7, we evaluate the contextualization capabilities of prior slide-level foundation models, specifically TITAN [9] and Gigapath-SE [66], benchmarking them against TICON. For tile-level evaluation, we pass the WSI's tile embeddings (using CONCHv1.5 for TITAN and Gigapath for Gigapath-SE) through their respective slide encoders.
   We discard the global '[CLS]' token and utilize the output tile embeddings for the CATCH dataset. For slide-level tasks, we follow our established protocol by training a TANGLE aggregator on the contextualized tile embeddings produced by each slide encoder. We make two key observations: (1) Both TITAN and Gigapath-SE, when acting as contextualizers, yield improvements on 2 out of 3 tasks compared to their non-contextual baselines. (2) TICON consistently outperforms both slide encoders, even when restricted to the same input tile encoder. Furthermore, TICON achieves superior overall performance on tile-level tasks by leveraging its flexibility to process state-of-the-art tile encoders (e.g., H-optimus-1 on CATCH), a capability lacking in the baseline slide encoders, which are tied to a specific input tile encoder.
   We further highlight the efficacy of our omni-feature

multi-target masked pretraining by comparing the results in Table 6 and Table 7. We observe that TICON outperforms TITAN across all 3 tasks (using the same CONCHv1.5 input) *only* when the multi-target objective is employed in pretraining. In contrast, without multi-target pretraining, TICON falls behind TITAN on 2 of the 3 tasks, underscoring the critical importance of cross-encoder alignment. Consequently, our multi-target approach enables highly data-efficient pretraining: despite relying on only 11K WSIs from the open-source TCGA dataset, TICON surpasses baselines like TITAN, which benefit from pretraining on over 350K WSIs.

Takeaway: *While current slide encoders offer gains when used as contextualizers, TICON's multi-target pretraining delivers superior performance and data efficiency, consistently outperforming baselines even with magnitudes less pretraining data.*

3. **Ablation on TICON's Architecture:** We conduct a comprehensive ablation study on TICON's architectural design choices in Table 8:
   - **Masking ratio (m_r):** We compare masking ratios of 90% and 75%. While the model remains robust even at a high masking ratio of 90%, we observe that $m_r = 75\%$ generally yields the most stable and optimal performance across tasks; thus, we adopt it as our default.
   - **Embedding size (D):** We evaluate TICON's shared embedding space with dimensions of $D \in \{384, 768, 1536\}$. Increasing the capacity to $D = 1536$ results in consistent performance gains across all encoders on HEST-Bench. On CATCH, while smaller dimensions lead to inconsistent rankings when inferred with different tile encoders, $D = 1536$ provides the most robust and stable performance, justifying its selection for our final model.
   - **Depth (l):** We vary the depth of the shared encoder ($l \in \{4, 6, 12\}$). We find that a depth of 6 layers ($l = 6$) offers the optimal trade-off between performance and computational efficiency. Similar to our observations regarding embedding size, depths of 4 or 12 result in fluctuating performance rankings on HEST-Bench and CATCH, whereas $l = 6$ consistently provides stable and high performance across both tasks and diverse tile encoders.

4. **Extended Analysis on HEST-Bench:** Table 9 presents an extended analysis of the gene expression prediction task on HEST-Bench. In the main paper, our baselines relied solely on the '[CLS]' token from the tile encoder and so do TICON's input. Here, we adopt the more rigorous protocol common in recent literature, where the '[CLS]' token is concatenated with the mean-pooled patch tokens from the tile encoder. We then concatenate TICON's contextualized output (which corresponds to the '[CLS]' tokens input) with the non-contextual '[CLS]' and mean-pooled patch tokens.
   We observe that while including mean-pooled features significantly boosts the baseline performance, particularly for UNI2-h, TICON successfully maintains its lead, pushing the state-of-the-art frontier across all tile encoders. It is worth noting that our current TICON model is pretrained using only the '[CLS]' token; explicitly incorporating mean-pooled embeddings during TICON's pretraining could potentially yield further performance improvements on benchmarks like HEST and other tasks.
   Takeaway: *TICON's slide-level contextualization provides complementary information to local feature aggregation, consistently boosting performance even against strengthened baselines that incorporate mean-pooled representations.*

5. **TransMIL as Aggregator in TANGLE Pretraining:** We investigate the choice of the underlying aggregation mechanism for TANGLE slide-level contrastive pretraining in Table 10. We compare our default Attention-Based MIL (ABMIL) aggregator against the Transformer-based TransMIL [55]. Evaluating on Patho-Bench (BRACS and CPTAC), we observe that the simpler ABMIL aggregator consistently outperforms TransMIL. This indicates that learning contextualization via a Transformer within a multimodal contrastive setting may be suboptimal. Crucially, this highlights the distinct advantage of TICON as a standalone contextualizer: it demonstrates that learning context through a dedicated masked modeling objective (reconstructing visual embeddings) is more effective than attempting to

Table 7. Comparison of our TICON with TITAN and Gigapath-SE (slide encoder) on tile-level and slide-level tasks. † For Gigapath tile encoder, we report our results with TICON$_{tune}$. Individual aggregator is pretrained with TANGLE prior to linear probing for slide-level tasks. $\boxed{\cdot}$ denotes the state-of-the-art on the respective tasks.

		METHOD	H-optimus-1	UNI2-h	CONCHv1.5	Gigapath
Tile tasks	CATCH	Tile-encoder only	86.2	85.5	81.6	83.0
		w/ TITAN	NA	NA	84.4	NA
		w/ Gigapath-SE	NA	NA	NA	82.9
		w/ TICON	**87.6**	**86.9**	**84.9**	**83.2**†
Slide-level tasks	BRACS	Tile-encoder only	60.4	53.8	60.6	58.4
		w/ TITAN$_{tangle}$	NA	NA	63.7	NA
		w/ Gigapath-SE$_{tangle}$	NA	NA	NA	59.1
		w/ TICON$_{tangle}$	**62.6**	**58.9**	**63.8**	**59.9**†
	CPTAC	Tile-encoder only	68.7	69.5	71.6	68.9
		w/ TITAN$_{tangle}$	NA	NA	71.4	NA
		w/ Gigapath-SE$_{tangle}$	NA	NA	NA	69.6
		w/ TICON$_{tangle}$	**72.5**	**70.3**	**72.7**	**70.3**†

Table 8. **Ablation Studies.** We evaluate tile-level tasks with slide context, reporting the Pearson correlation coefficient (PCC) for HEST-Bench and the F1-score for CATCH. We compare the baseline tile encoders against our TICON framework across three variations: (a) Masking ratio (m_r), (b) Slide Encoder's embedding size (D), and (c) Slide Encoder's depth (l). † represents the default parameter.

(a) Masking ratio (m_r)

Method	m_r	HEST	CATCH
H-optimus-1	NA	0.422	86.2
w/ TICON	90%	**0.428**	87.5
	75%†	0.427	**87.6**
UNI2-h	NA	0.414	85.5
w/ TICON	90%	**0.420**	86.3
	75%†	0.419	**86.9**
CONCHv1.5	NA	0.379	81.6
w/ TICON	90%	0.371	84.8
	75%†	**0.381**	**84.9**

(b) Embedding size (D)

Method	D	HEST	CATCH
H-optimus-1	NA	0.422	86.2
w/ TICON	384	0.425	87.1
	768	0.423	**87.9**
	1536†	**0.427**	87.6
UNI2-h	NA	0.414	85.5
w/ TICON	384	0.417	**87.1**
	768	0.417	86.7
	1536†	**0.419**	86.9
CONCHv1.5	NA	0.379	81.6
w/ TICON	384	0.362	84.6
	768	0.378	**85.1**
	1536†	**0.381**	84.9

(c) Depth (l)

Method	l	HEST	CATCH
H-optimus-1	NA	0.422	86.2
w/ TICON	4	0.425	**87.8**
	6†	**0.427**	87.6
	12	**0.427**	86.4
UNI2-h	NA	0.414	85.5
w/ TICON	4	0.418	86.4
	6†	**0.419**	86.9
	12	0.418	**87.3**
CONCHv1.5	NA	0.379	81.6
w/ TICON	4	**0.383**	**84.9**
	6†	0.381	**84.9**
	12	0.376	84.4

Table 9. HEST performance with CLS only and with CLS+meanpool concatenated.

Method	**HEST-Bench** (9 tasks)
H-optimus-1 (CLS)	0.422
w/ TICON	0.427
H-optimus-1 (CLS+meanpool)	0.431
w/ TICON	**0.437**
UNI2-h (CLS)	0.414
w/ TICON	0.419
UNI2-h (CLS+meanpool)	0.431
w/ TICON	**0.437**
CONCHv1.5	0.379
w/ TICON	**0.381**

Table 10. Comparison with TransMIL as aggregator for non-contextual tile embeddings with Patho-bench [68]. Following Pathobench, we report balanced accuracy for BRACS and AUC for CPTAC. TICON$_{tangle}$ improves over Tangle across both aggregators. Note that a multi-head version of ABMIL is used as aggregator in this study for baselines and ours.

Model	Aggregator	Tile Encoder	BRACS 2 tasks	CPTAC 25 tasks
Tangle [24]	ABMIL	CONCHv1.5	60.6	71.6
	TransMIL	CONCHv1.5	60.3	69.0
TICON$_{tangle}$	ABMIL	CONCHv1.5	**63.8**	**72.7**

derive context implicitly through alignment with auxiliary modalities like gene expression. Consequently, pretraining an aggregator on these TICON-contextualized embeddings leads to a state-of-the-art slide-level foundation model, despite using only 11K WSIs. This finding suggests that large-scale whole slide-encoder pretraining methods like TITAN and PRISM, which access hundreds of thousands of WSI-report pairs, could potentially achieve even greater performance by integrating TICON's contextualized embeddings into their vision-language frameworks.

Takeaway: *Our Contextualizer as a standalone stage benefits Slide-level pretraining with TANGLE over trying to train the contextualizer (transformer based MIL) too with the multimodal contrastive objective.*

6. **Unified vs. Individual TANGLE Pretraining:** In our primary evaluation (main paper Table 3), we pretrained a separate MIL aggregator using TANGLE objective for each tile encoder for baseline and for it's TICON's contextualized embeddings. In Table 11, we explore a *unified* approach: training a single TANGLE model on the contextualized embeddings from all three pretraining tile encoders (UNI2-h, CONCHv1.5, and H-optimus-1). Since TICON maps all inputs to a shared embedding dimension D, we train this unified MIL aggregator by randomly sampling the source tile encoder for each WSI within a batch.

We observe that while the individually pretrained aggregators generally outperform the unified model, the unified TANGLE still surpasses the non-contextual baselines in 5 out of 6 cases and matches performance in the remaining one. Future work could further optimize this unified MIL aggregator pretraining, for instance, through homogeneous batch sampling (fixing the tile encoder per batch) or by incorporating tile-encoder-specific projection MLP heads similar to TICON's design.

Takeaway: *While unified training proved beneficial for TICON's masked modeling objective, we found that for contrastive objectives like TANGLE (WSI-gene align-*

Table 11. Comparison of unified vs. individual tile encoder (default) TANGLE pretraining. Evaluation on slide-Level tasks with Patho-Bench [68]. Following Patho-Bench, we report balanced accuracy for BRACS and AUC for CPTAC.

Model	Tile Encoder	BRACS 2 tasks	CPTAC 25 tasks
Tangle [24]	H-optimus-1	60.4	68.7
	UNI2-h	53.8	69.5
	CONCHv1.5	60.6	71.6
$\text{TICON}_{tangle\_unified}$	H-optimus-1	61.3	72.5
	UNI2-h	58.8	70.0
	CONCHv1.5	60.6	71.9
TICON_{tangle}	H-optimus-1	62.6	72.5
	UNI2-h	58.9	70.3
	CONCHv1.5	**63.8**	**72.7**

ment), individual aggregator training remains superior. The unified TANGLE training requires further exploration.

7. **Tangle vs Meanpool:** We evaluated two different aggregators for slide-level tasks with TICON: Meanpool and Tangle [24], an attention-based aggregator. Both aggregators are used to extract a single global representation embedding for the whole WSI from contextualized tile embeddings. As observed in Table 12, TICON_{tangle} substantially outperforms $\text{TICON}_{meanpool}$ across all tile encoders. This highlights the effectiveness of using a Tangle-based aggregator to derive maximum benefit from the contextualized embeddings. This finding is consistent with recent evaluations of MAE pretrained encoders [45] in natural imaging, which confirm that attention-based aggregators more effectively represent an image at the global level compared to mean-pooling of patch tokens.

Takeaway: *Attention pooling is the superior method to mean-pooling for extracting a single global representation from contextualized embeddings.*

Table 12. Comparison of using Meanpool and Tangle aggregators to build a slide-level foundation model using TICON. We chose the Tangle aggregator because it clearly outperforms Meanpool in both datasets.

Model	Tile Encoder	BRACS 2 tasks	CPTAC 25 tasks
$\text{TICON}_{meanpool}$	Hoptimus-1	50.5	69.8
	UNI-2	50.4	68.3
	Conch-v15	51.8	69.1
TICON_{tangle}	H-optimus-1	62.6	72.5
	UNI2-h	58.9	70.3
	CONCHv1.5	**63.8**	**72.7**

8. **CKA analysis for shared semantic space:** We perform Centered Kernel Alignment (CKA) analysis using 10 tiles randomly located on each of 1000 randomly sampled TCGA WSIs (n = 10, 000). We use the CKA implementation from [20]. As shown in the Table 13, TICON consistently improves feature alignment across all pairs of tile encoders. This observation provides empirical evidence for unified representation learning.

Table 13. Centered Kernel Alignment (CKA) analysis.

CKA	(H-optimus-1,UNI2-h)	(H-optimus-1,CONCHv1.5)	(UNI2-h, CONCHv1.5)
Tile-encoders	0.70	0.50	0.45
w/ TICON	0.85 (+15%)	0.74 (+24%)	0.74 (+29%)

B. Additional implementation details

1. **TANGLE Setup:** For slide-level tasks, we pretrain a multi-head (2 heads) ABMIL aggregator to pool the tile embeddings. We fix the hidden dimension at 512 and incorporate a 3-layer pre-attention MLP, following the architecture specified in the TANGLE pan-cancer pretraining codebase. For the gene modality, we employ a 3-layer MLP with a hidden dimension of 512. For all TANGLE pretraining runs, we use a batch size of 512 and fix the number of randomly sampled tokens per WSI to 4096 to enable batching [24]. Apart from these changes, we adopt all default hyperparameters from the original TANGLE implementation.

2. **Harmonizing the Field of View for Tile Encoders:** We observe that input tile resolutions vary significantly across encoders (e.g., 512×512 px for CONCHv1.5, 224×224 px for H-optimus-1 and Virchow2, and 256×256 px for UNI2-h and Prov-GigaPath). To harmonize these for our omni-pretraining, we standardize the base extraction area to 512×512 px. For native 512×512 encoders like CONCHv1.5, we extract embeddings directly. For encoders requiring smaller inputs, we subdivide the 512×512 area into four 256×256 quadrants, extract embeddings for each (resizing to 224×224 if necessary), and compute their mean. This ensures a one-to-one spatial correspondence across all tile encoders. Furthermore, this pooling strategy significantly enhances computational efficiency during inference for both TICON and the subsequent TANGLE aggregation by reducing the effective sequence length by a factor of four for all encoders except CONCHv1.5 (which natively operates at 512×512).
Crucially, since this mean-pooling strategy preserves the semantic distribution of the embedding space, TICON retains the flexibility at inference time to process either the aggregated representations or the original fine-grained embeddings directly. This capability is essential for benchmarks like HEST, which require gene expres-

sion predictions at the native resolution (224×224 px at $20\times$). Similarly, for THUNDER, we process each tile at the native resolution required by different tile encoders and pass the resulting single embedding through TICON (where the sequence length is 1, effectively TICON acting as a deep MLP).

Comparison with baseline Tile-encoder. To ensure a fair comparison for the "Tile Encoder Only" baselines (excluding HEST and THUNDER), we apply the same aggregation methodology: non-512×512 tile encoder outputs are mean-pooled prior to downstream usage (e.g., k-NN classification for CATCH or TANGLE pretraining). This ensures a strictly one-to-one comparison between the non-contextual tile embeddings and our TICON-contextualized counterparts.

Comparison with baseline Slide-encoders. In contrast, for baseline slide encoders such as Gigapath-SE [66], which expect fine-grained input embeddings (e.g., 256×256 px), we bypass TICON's pooling operation. As reported in Table 7, we train the TANGLE aggregator for Gigapath-SE by performing contextualization directly on these native resolution tiles. Conversely, since TITAN [9] is designed to process CONCHv1.5 embeddings (512×512 px), we apply it directly without modification. For the CATCH tile classification task, where ground-truth labels are defined at the 512×512 px resolution, we adapt the Gigapath-SE output by mean-pooling the contextualized embeddings of the corresponding 2×2 quadrant to yield a single representative embedding for each labeled 512×512 region.

3. **CATCH Data Curation:** We observe that despite the recent proliferation of histopathology benchmarks (e.g., HEST-Bench, THUNDER, and Patho-Bench), there remains a scarcity of datasets that enable tile-level evaluation within a full slide-level context. HEST-Bench serves as a notable exception: while originally proposed as a tile-level task, it retains the spatial coordinates for all tiles, allowing us to reformulate it as a tile-level task with full slide context available. This broader gap in the field primarily stems from the traditional "bag-of-words" paradigm, which typically treats tile-level feature extraction and slide-level aggregation as distinct, decoupled tasks. To bridge this gap and evaluate the impact of context on local predictions, we curate the CATCH dataset to support tile-level classification while retaining the spatial integrity of the Whole Slide Image.
   For this curation, we utilize the segmentation contours available in the CATCH [63] dataset. From the original 13 classes, we exclude "cartilage" due to its low prevalence, retaining the remaining 12 classes. We compute the overlap of 512×512 tiles (at $20\times$ magnification) with the original contours. A tile is assigned a label only if it is fully contained (100% overlap) within a contour of one of the 12 classes. To ensure an unambiguous multi-class classification task, we discard any tiles that intersect with contours of multiple different classes. This process results in a dataset of 916,967 patches derived from 350 WSIs, split into 210 for training, 49 for validation, and 91 for testing. We evaluate performance using k-NN probing, selecting the optimal k on the validation set and reporting the final performance on the test set.
   We plan to release this benchmark along with the whole curation pipeline. We anticipate that our research, by bridging the traditionally decoupled stages of tile embedding extraction and aggregation with a slide-encoder as contextualizer, will catalyze the creation of further benchmarks designed to evaluate the dense prediction capabilities of slide encoders, moving beyond solely global slide-level tasks.

4. **Evaluation Setting:** In this study, we strictly adhere to the default metrics and hyperparameters established by the respective benchmarks.
   For slide-level tasks, we utilize Patho-Bench, adopting the two subtyping tasks (coarse-grained and fine-grained) from BRACS and the 25 mutation prediction tasks from CPTAC. For both datasets, we employ linear probing with a balanced loss and a cost parameter of 0.5, while keeping all other parameters at their defaults. We report performance using the specific metrics provided by the benchmark for each task.
   For tile-level tasks, we follow specific established protocols:
   - **HEST-Bench:** We adopt the default setup of applying PCA to reduce dimensions to 256, followed by ridge regression.
   - **THUNDER:** We utilize their default pipeline to report average k-NN results for the 12 original tasks and the 4 newly added SPIDER [40] tasks.
   - **CATCH:** Aligning with the THUNDER protocol, we employ the same k-NN based evaluation strategy.

   Importantly, for these tile-level tasks, which do not require additional learnable parameters (due to the use of PCA or non-parametric k-NN) with increase in feature size, we enhance the representation by concatenating the original non-contextual tile embeddings with their corresponding TICON-contextualized embeddings (or the output of TICON_{iso} for THUNDER). Conversely, for slide-level tasks, to maintain consistency in TANGLE's input dimension, we utilize only the contextualized tile embeddings, discarding the original non-contextual inputs.
5. **TICON's architecture and pretraining setting:**
   TICON was pretrained using the default hyperparameters listed in Table 14 on a setup consisting of 8 $\times$

NVIDIA A100 40GB GPUs. Memory profiling during the training indicated a consumption of approximately 5GB per GPU. The entire pretraining was completed in 10 hours.

Table 14. Pretraining hyperparameters of TICON

Hyperparameter	**Value**
Batch size	1024
Optimizer	AdamW(0.9, 0.95)
Learning rate	2e-4
Weight decay	0.05
Warmup iterations	10K
Total iterations	100K
Training dtype	bf16
Parallelism	FSDP
Masking type	random
Masking ratio (m_r)	75%
Prediction ratio (p_r)	25%

TICON pretraining consists of an Encoder (ViT 6 layers, 1536 embed dim) with 170 million (M) parameters and a Cross-Decoder (ViT 1 layer, 1536 embed dim) with 28M parameters. Additionally, the input and output projectors (2 layer MLP) each contribute up to 5M parameters. When adapting to unseen tile encoders, we only train the parameters of new projectors for 20K iterations.

Partial prediction. We only reconstruct a partial amount of the target embeddings during pretraining, based on a prediction ratio (p_r), rather than all of the masked embeddings. We choose this partial prediction strategy because our pretraining candidates contain a minimum of 55% tiles with tissue, while the remaining tiles are invalid regions. We exclude these invalid regions from both the visible embeddings (input to the encoder) and the prediction targets (output of the decoder). Since we use a masking ratio (m_r) of 75%, we opt for a prediction ratio (p_r) of 25% of the total embeddings. This restriction ensures that the total operated embeddings (visible 25% in the encoder and target 25% in the decoder) remains below the minimum tissue occupancy of 55% in the pretraining candidates. A recent study [13] demonstrated that a cross-attention-based decoder (Cross-Decoder) is better suited for partial prediction than its self-attention counterpart. Consequently, we adopted a Cross-Decoder for our pretraining architecture.

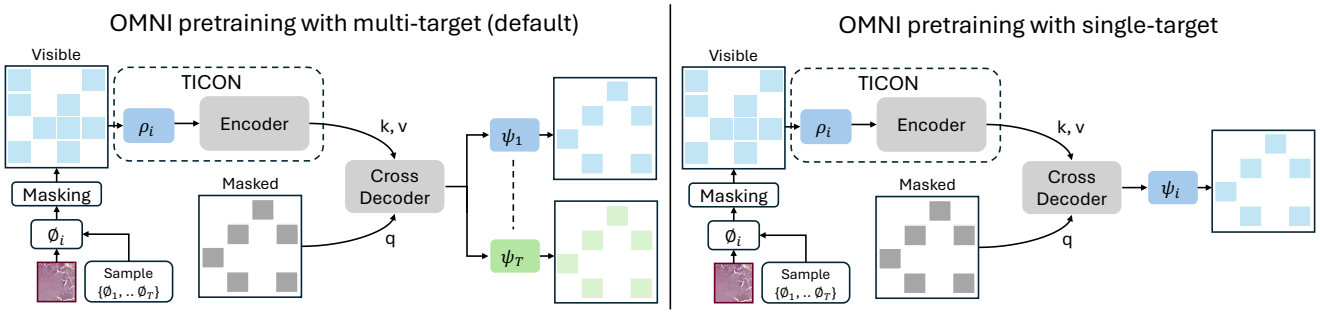

Figure 5. Overview of TICON's multi-target versus single-target pretraining paradigms. **(Left) Default Multi-Target Pretraining:** At each iteration, we randomly sample a single input tile encoder. Its embeddings are projected and encoded, after which the decoder is tasked with reconstructing the masked embeddings for *all* tile encoders (used in omni-pretraining) simultaneously. This mechanism enforces cross-encoder semantic alignment. **(Right) Single-Target Pretraining:** In this ablation setting, the model retains the capacity to process any encoder (omni-compatible) but lacks cross-target supervision. Specifically, the decoder is restricted to reconstructing only the masked embeddings of the input encoder itself. Thus, while the input encoder varies randomly across iterations, the target is always identical to the input, removing explicit cross-encoder prediction.

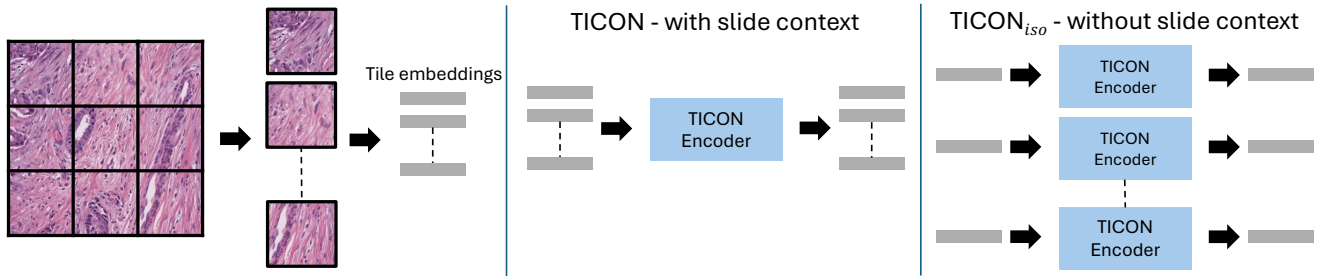

Figure 6. Overview of TICON's inference modes. **(Left)** Standard preprocessing pipeline: tiling the WSI followed by embedding extraction. **(Middle) Contextualized Inference:** The default mode where the entire sequence of WSI tile embeddings is passed through the TICON Encoder. This allows the model to contextualize each tile with information from the full slide-level neighborhood. **(Right) Isolated Inference:** An alternative inference mode where a single tile embedding is passed through TICON independently. In this setting, the Transformer effectively functions as a deep MLP (sequence length of 1). Although not the primary design intent, we empirically discovered that TICON exhibits an emergent property in this mode, enhancing individual tile representations even when slide-level context is unavailable (e.g., in the THUNDER benchmark).