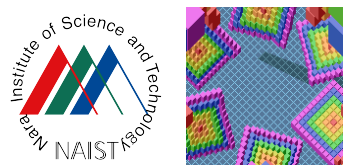
MergeSlide: Continual Model Merging and Task-to-Class Prompt-Aligned Inference for Lifelong Learning on Whole Slide Images

WACV'2026, Tucson, Arizona

Doanh C. Bui[1,*], Ba Hung Ngo[2], Hoai Luan Pham[1], Khang Nguyen[3], Maï K. Nguyen[4], Yasuhiko Nakashima[1]

[1]Nara Institute of Science and Technology, Japan
[2]Graduate School of Data Science, Chonnam National University, Republic of Korea
[3]University of Information Technology, VNU-HCM, Viet Nam
[4]ETIS (UMR 8051), CY Cergy Paris University, ENSEA, CNRS, France



Outline

Outline

Motivation

- Rehearsal-based methods have often been observed to outperform others, yet they are accompanied by risks of data leakage and additional memory usage due to buffer storage.
- The implementation of such models is typically evaluated under the assumption that all tasks contain the same number of classes for whole slide image analysis, which is impractical.
- Performance consistency cannot be guaranteed when the task order is altered, as their training schemes are highly dependent on interactions with previous parameters or WSIs stored in buffers.

Outline

MergeSlide (Continual Model Merging)

Continual Model Merging

Model merging aims to combine the weights/parameters from task-specific models into a unified model while preserving performance. Specifically, given a predictive model $f(\cdot, \boldsymbol{\theta}^*) : x \rightarrow \hat{y}$ and a set of parameters $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_t\}_{t=1}^{T}$, where $\boldsymbol{\theta}_t$ denotes the parameters for the t-th task, the model-merging function $\mathcal{M}^{\mathcal{C}}(\cdot)$ is designed to merge all task-specific parameters **sequentially** into a unified set based on prior knowledge represented by $\boldsymbol{\theta}_{\text{prior}}$:

$$\boldsymbol{\theta}_t^* = \mathcal{M}^{\mathcal{C}}(\boldsymbol{\theta}_t, \boldsymbol{\theta}_{t-1}^*, \boldsymbol{\theta}_{\text{prior}}) \quad \forall \boldsymbol{\theta}_t \in \boldsymbol{\Theta}.$$

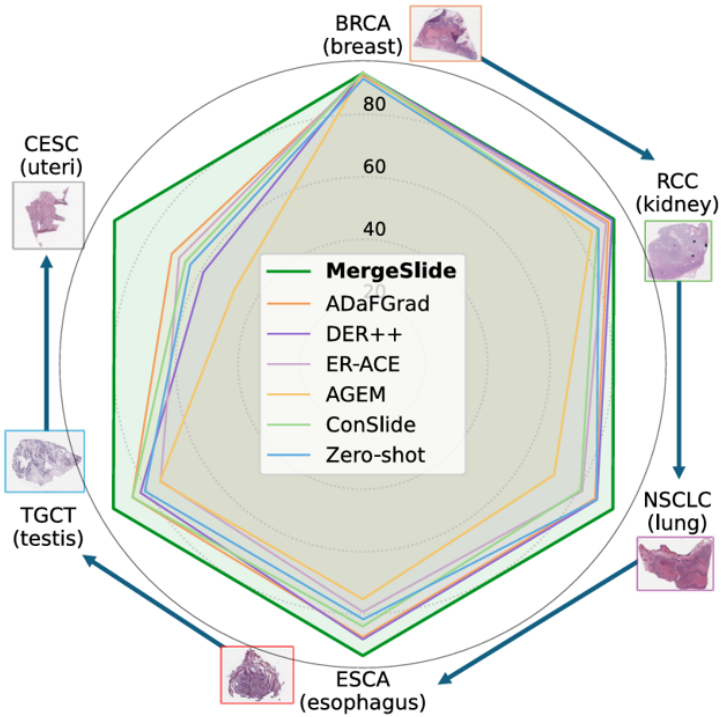
Finally, $\boldsymbol{\theta}_t^*$ should preserve the knowledge from all tasks, and the model $f(\cdot, \boldsymbol{\theta}_t^*) : x \rightarrow \hat{y}$ should be able to perform predictions on all tasks encountered so far.

Herein, $\boldsymbol{\theta}_{\text{prior}}$ could be initialized from pre-trained weights of pathology VLMs.

Outline

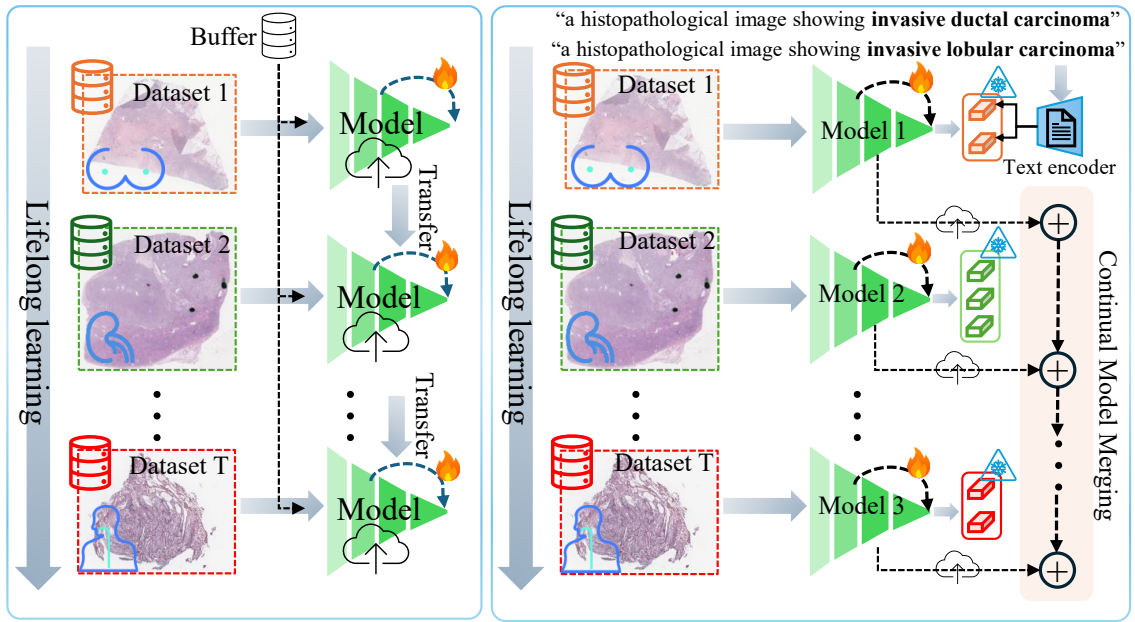
MergeSlide (Overview)

- Solution: MergeSlide first trains an classifer-free model offline using frozen class-aware prompt embeddings. Then, the weights are merged task-by-task.



a) Accuracy over increasing number of tasks

Figure: Performances when tasks increase.



a) Rehearsal-based Methods b) MergeSlide: Finetuning and merging

Figure: MergeSlide vs. Others.

Mergeslide (Method #1)

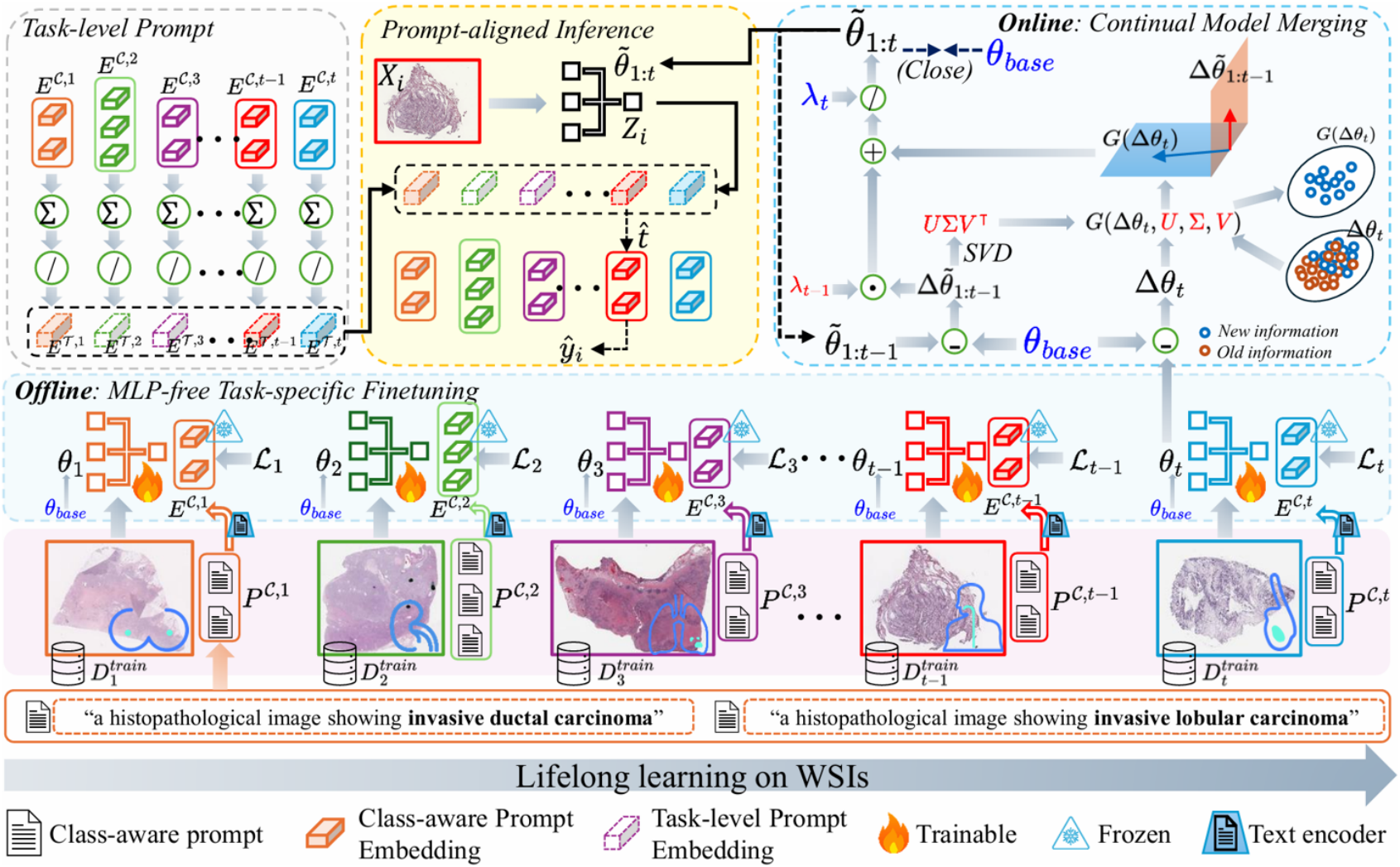


Figure: Overview of MergeSlide.

Mergeslide (Method #2)

- **Step 1**: For each task, effectively describe target cancer subtypes and obtain their embeddings using a pathology VLM. Then, train for a few epochs with an classifer-free backbone.

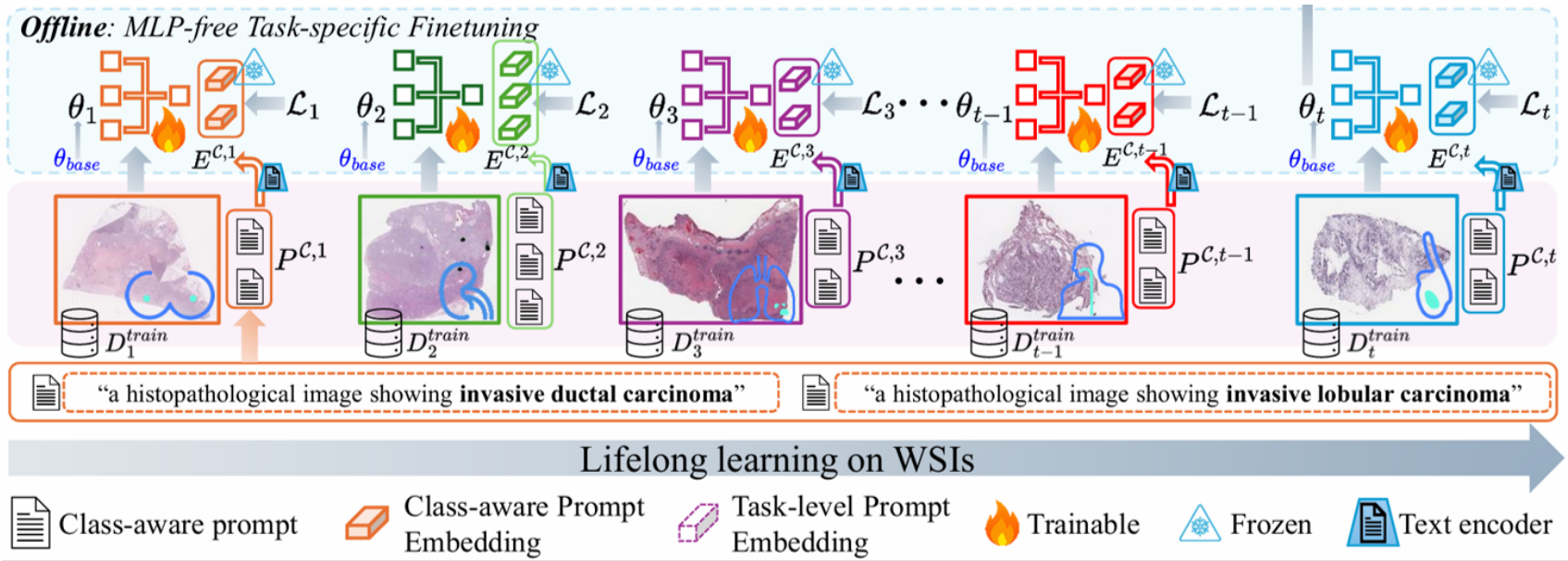


Figure: Step 1. Describing cancer subtypes and finetuning. θ_{base} is initialized from TITAN [5]

Mergeslide (Method #3)

- **Step 2**: Combine task-specific weights into a unified model capable of handling all previously seen cancer tasks without requiring a rehearsal buffer.

Task vector : $\Delta\theta_t = \theta_t - \theta_{base}$

(1) : $$\tilde{\theta}_{1:t} = \theta_{base} + \frac{\lambda_{t-1}\Delta\tilde{\theta}_{1:t-1} + G(\Delta\theta_t)}{\lambda_t},$$

(2) : $$G(\Delta\theta_t) = U\left(\left(U^\top \Delta\theta_t V\right) \odot M\right) V^\top,$$

where $U, \Sigma, V = SVD(\Delta\tilde{\theta}_{1:t-1})$

(3) : $$\lambda_t = t \cdot \frac{\left\|\lambda_{t-1}\Delta\tilde{\theta}_{1:t-1} + G(\Delta\theta_t)\right\|_2}{\sum_{i=1}^{t}\|\theta_i\|_2},$$

where $\lambda_1 = 1$,

(4) : to ensure $$\frac{1}{t}\sum_{i=1}^{t}\|\theta_i\|_2 \leq \max_{i\in\{1,\ldots,t\}}\|\theta_i\|_2$$

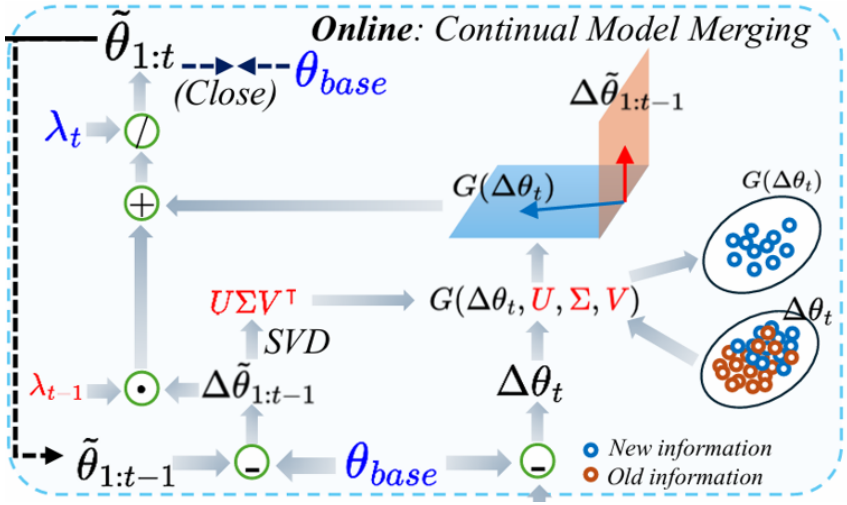


Figure: Step 2. Model Merging

Mergeslide (Method #4)

- In this manner, $G(\Delta\theta_t)$ extracts the novel information of the t-th task relative to previous tasks.

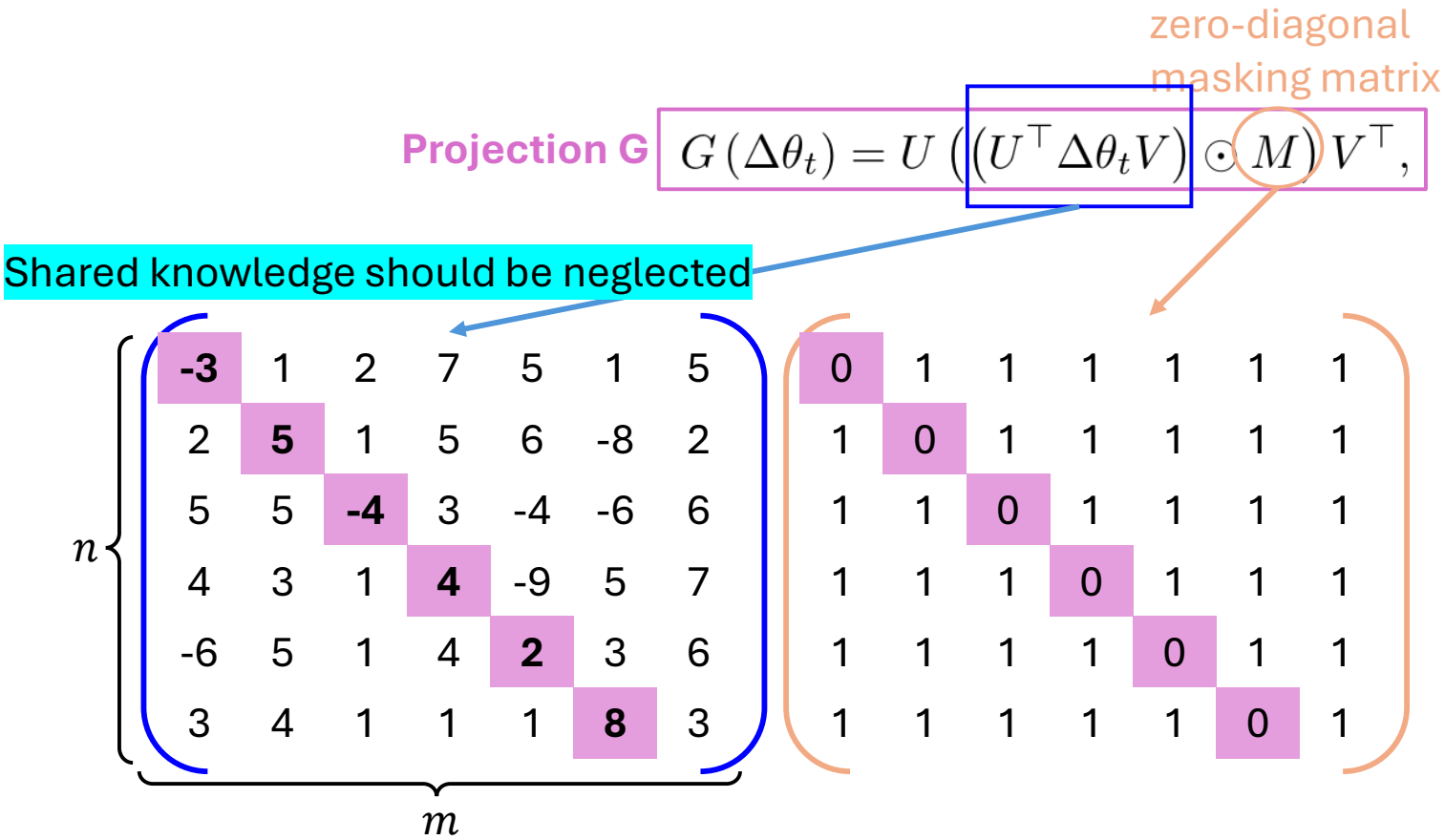


Figure: Explanation of projection $G(\Delta\theta_t)$.

Mergeslide (Method #5)

- **Step 3**: During inference, to enhance the performance on CLASS-IL scenario, the backbone generates slide embeddings, and predictions are made via dot-product similarity with task- and class-level prompt embeddings.

Task prompt for task t: $E^{\mathcal{T},t} = \frac{1}{c_t} \sum E^{\mathcal{C},t}$

$$\hat{t} = \arg\max_t \left\{ Z_i \cdot (E^{\mathcal{T},t})^\top | \forall E^{\mathcal{T},t} \in \mathcal{E}^{\mathcal{T}} \right\}.$$

Prediction: $\hat{p}_i = \{ Z_i \cdot (e_j^{\mathcal{C},\hat{t}})^\top | \forall e_j^{\mathcal{C},\hat{t}} \in E^{\mathcal{C},\hat{t}} \}$,

where $\hat{y}_i = \arg\max \hat{p}_i$.

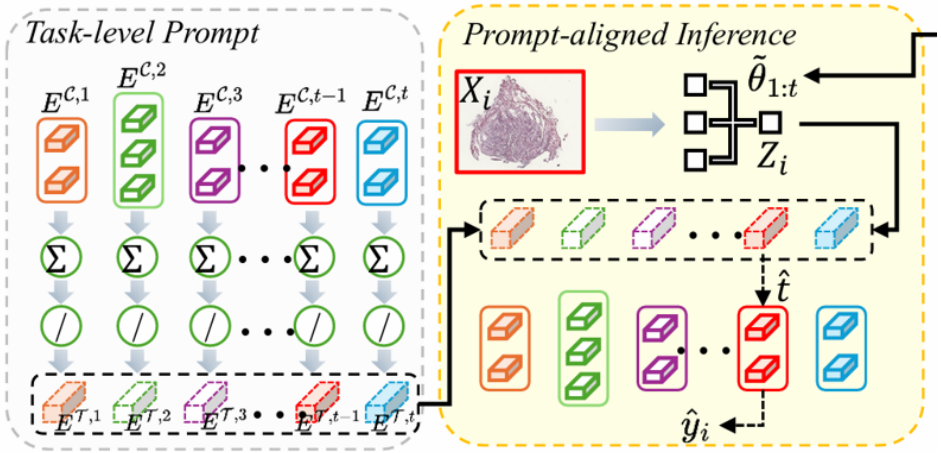


Figure: Step 3. Prompt-aligned Inference

Outline

Main results

- MergeSlide significantly outperforms others!

Method	Buffer size	bACC ↑ (CLASS-IL)	Masked bACC ↑ (TASK-IL)	Mean ACC↑ (CLASS-IL)	FGT ↓	#BWT ↑
Fully Supervised	0 WSIs	81.309 (±3.064)	89.688 (±1.687)	–	–	–
Naive Finetuning		25.673 (±2.571)	86.597 (±1.933)	69.834 (±3.566)	4.649 (±2.558)	-8.035 (±3.795)
Zero-shot		68.243 (±2.558)***	86.814 (±2.105)***	82.593 (±0.017)***	**0.909 (±0.406)**	-0.909 (±0.406)
ER-ACE [3]	10 WSIs	44.354 (±4.280) ***	67.407 (±5.282) ***	71.663 (±3.265)***	5.095 (±2.429)	-0.320 (±1.211)
AGEM [4]		42.076 (±4.287)***	80.364 (±2.206)***	74.514 (±4.016)***	8.167 (±3.836)	-7.714 (±3.978)
DER++ [2]		52.033 (±4.272) ***	84.735 (±2.407)***	81.786 (±1.575)***	4.266 (±1.089)	-3.627 (±1.182)
ConSlide [6]		62.731 (±4.488)***	87.027 (±2.553)***	81.992 (±1.055)***	4.849 (±2.527)	**0.196 (±1.745)**
ADaFGrad [1]		65.267 (±2.235)***	89.458 (±2.299)***	83.651 (±1.755)***	2.772 (±1.633)	-1.589 (±1.693)
ER-ACE [3]	30 WSIs	58.556 (±5.654)***	75.255 (±5.161)***	81.138 (±2.464)***	4.251 (±2.606)	0.088 (±3.317)
AGEM [4]		42.076 (±4.287)***	80.364 (±2.206)***	74.514 (±4.016)***	8.167 (±3.836)	-7.714 (±3.978)
DER++ [2]		55.560 (±2.746)***	86.863 (±2.810)***	83.580 (±1.532)***	3.199 (±1.640)	-2.368 (±1.889)
ConSlide [6]		64.622 (±1.755)***	88.346 (±1.977)***	82.602 (±1.201)***	4.032 (±1.248)	-2.930 (±1.506)
ADaFGrad [1]		69.034 (±3.861)***	89.704 (±2.082)***	85.469 (±1.430)***	2.517 (±1.125)	-1.217 (±1.805)
MergeSlide (ours) (naive)	0 WSIs	80.668 (±1.860)	**92.087 (±1.740)**	90.640 (±1.247)	4.941 (±1.518)	5.384 (±1.317)
MergeSlide (ours) `w/ TCP`		**87.929 (±2.110)**	**92.087 (±1.740)**	**92.686 (±0.899)**	1.848 (±0.858)	-1.604 (±1.024)

Table: Comparison between MergeSlide and other continual learning methods on a benchmark of six TCGA datasets in the forward sequence of B→R→N→E→T→C.

More results #1

Method	Buffer size	bACC ↑ (CLASS-IL)	Masked bACC ↑ (TASK-IL)	Mean ACC↑ (CLASS-IL)	FGT ↓	#BWT ↑
Fully Supervised	0 WSIs	81.151 (±2.283)	90.138 (±1.854)	84.431 (±2.304)	–	–
Naive Finetuning		24.976 (±3.330)	86.614 (±3.543)	42.565 (±1.031)	7.561 (±4.773)	-6.982 (±4.557)
DER++ [2]	30 WSIs	61.037 (±3.319)***	84.758 (±1.929)***	78.983 (±1.955)***	6.644 (±4.062)	-5.676 (±4.490)
ConSlide [6]		70.821 (±2.180)***	88.896 (±2.442)**	77.146 (±1.407)***	3.117 (±1.784)	-2.521 (±1.940)
ADaFGrad [1]		77.096 (±4.474)***	90.216 (±1.677)**	81.775 (±1.227)***	2.304 (±1.072)	-0.821 (±0.918)
MergeSlide (ours) (naive)	0 WSIs	80.636 (±1.865)	**92.109 (±1.700)**	88.268 (±1.459)	4.246 (±1.585)	-1.808 (±1.987)
MergeSlide (ours) `w/ TCP`		**87.930 (±2.112)**	**92.109 (±1.700)**	**91.009 (±1.278)**	**1.807 (±0.604)**	**0.958 (±0.868)**

Table: Comparison between MergeSlide and other continual learning methods on a benchmark of six TCGA datasets in the reversed sequence of C→T→E→N→R→B.

Sequence	ACC	Masked ACC	Mean ACC
B→C→R→T→N→E	91.955 (±0.902)	93.637 (±1.084)	91.861 (±0.914)
E→N→T→R→C→B	91.975 (±0.915)	92.433 (±1.084)	91.381 (±1.126)
C→B→T→R→E→N	91.933 (±1.061)	93.637 (±1.084)	91.336 (±1.154)
N→E→R→T→B→C	91.955 (±0.926)	93.616 (±1.098)	91.021 (±1.041)
σ	0.0149	0.5184	0.3003

Table: Experiments on alternative task orders.

Method	bACC ↑ (CLASS-IL)	Masked bACC ↑ (TASK-IL)	FGT↓	$\Delta^{\text{bACC}}_{out-in}$↓
ER-ACE [3]	60.808 (±4.769)***	77.125 (±4.800)***	4.447 (±2.431)	+2.252
AGEM [4]	41.864 (±4.186)***	76.207 (±3.793)***	12.957 (±4.964)	-0.212
DER++ [2]	57.951 (±3.115)***	83.469 (±2.902)***	3.994 (±1.584)	+2.391
ConSlide [6]	61.336 (±5.791)***	85.945 (±4.393)***	3.214 (±1.967)	-3.286
ADaFGrad [1]	61.836 (±5.562)***	86.110 (±3.217)***	4.108 (±2.603)	-7.198
MergeSlide (ours)	**85.112 (±1.570)**	**89.575 (±2.440)**	**1.910 (±0.089)**	**-2.817**

Table: Comparison between MergeSlide and other continual learning methods on OOD setting.

More results #2

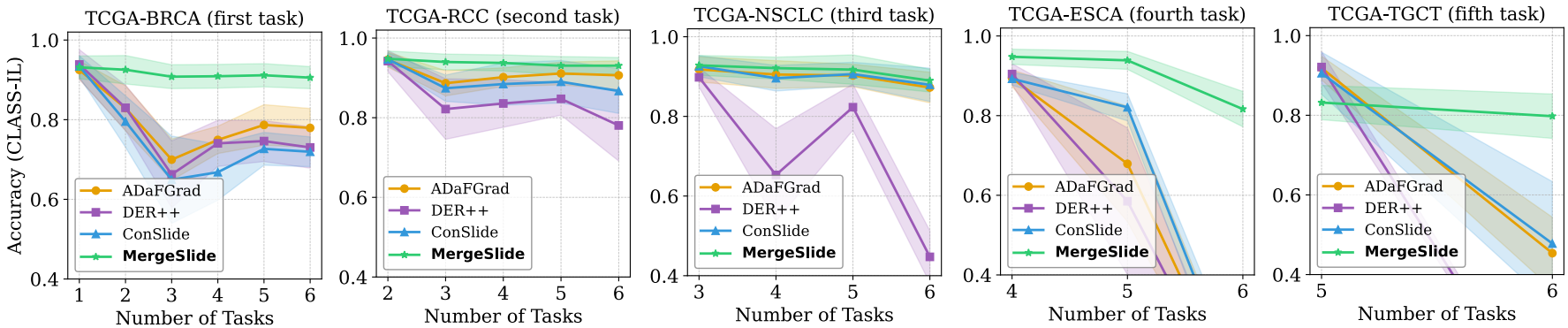


Figure: Performance drops of old tasks.

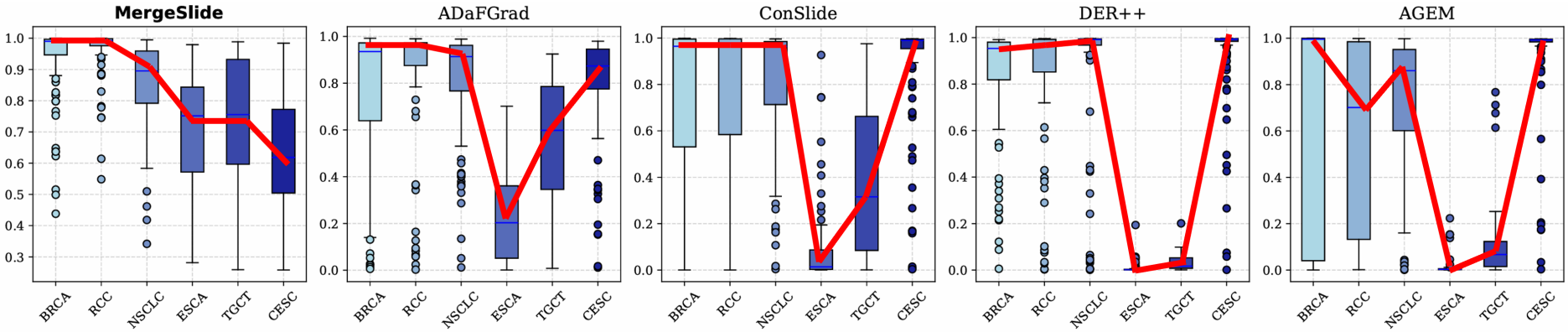


Figure: Confidence score study when new tasks are added.

More results #3

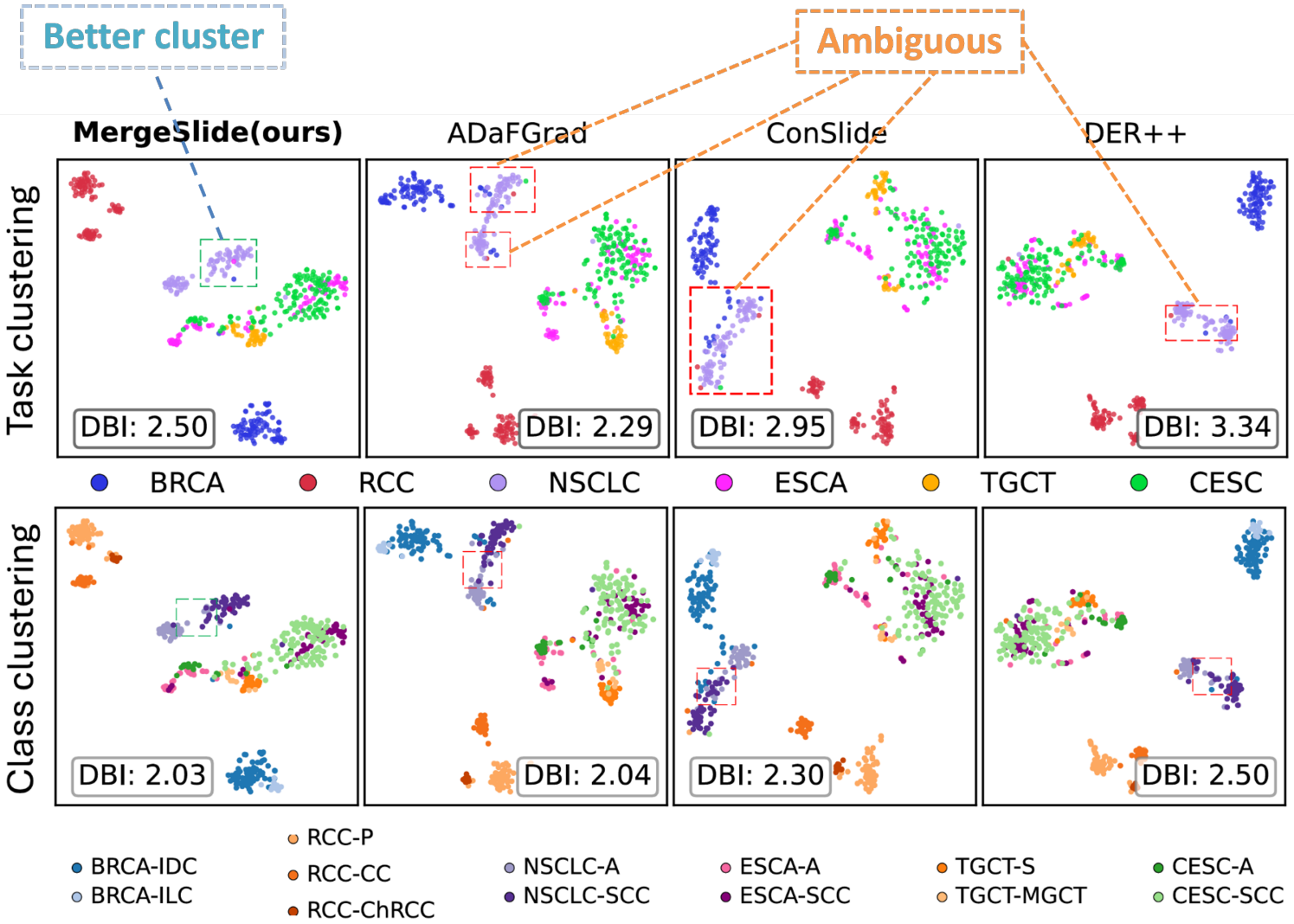


Figure: t-SNE visualization on task and class spaces.

More results #4

Stage	Method		Avg. Time (s)	VRAM (GB)	Slides/s
Training	MergeSlide	Per-task FT	74.26	3.38	–
		Merging	4.50	0.71	–
		Total	78.76	4.09	–
	ConSlide [6]		78.15	2.90	–
	ADaFGrad [1]		104.17	2.70	–
Inference	MergeSlide (naive)		1.52	3.80	55.81
	MergeSlide `w/ TCP`		1.52	3.89	55.82
	ConSlide [6]/ADaFGrad [1]		1.24	0.87	67.89

Table: Computational comparison of average epoch time (s), GPU VRAM (GB), and throughput (slides/s).

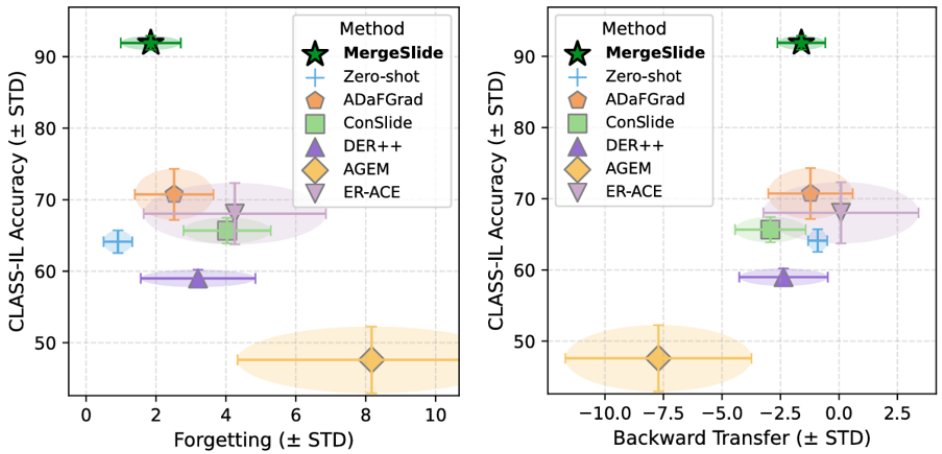


Figure: Accuracy-Forgetting trade-off

Outline

Conclusions (Limitations & Future work)

- **Extending to Broader Tasks:** Future work will evaluate the proposed approaches on more diverse and complex tasks beyond cancer subtyping, such as cancer grade classification, MSI status prediction, and survival analysis, to verify their generalizability and effectiveness in broader clinical contexts.
- **Enhancing Merge Efficiency:** Future research will explore dense simultaneous multi-task merging, where many model merging requests are handled concurrently on the global server, to achieve more efficient aggregation. Also, low-rank SVD is investigated to reduce computational overhead as the number of tasks increases.
- **Multimodal Learning:** Since WSIs capture micro-level tissue details, combining them with other modalities (e.g., CT, MRI, and textual reports) can provide complementary macro and semantic information. Future directions include developing multi-modal learning frameworks to more accurately model diagnostic and prognostic relationships in cancer analysis.

Thank you for your listening!