Sparse Task Vector Mixup with Hypernetworks for Efficient Knowledge Transfer in Whole-Slide Image Prognosis

Appendix

A. Derivations of subspace alignment ratio

Here, we complete the derivation of subspace alignment ratio (SAR), following [18]. Recall the definition of SAR given in the main paper, as follows:

$$\text{SAR}(\tau_t, \tau_{\text{new}}) = \frac{\|\text{Proj}_{\tau_{\text{new}}^\alpha} \tau_t\|_F}{\|\tau_t\|_F}. \tag{1}$$

τ_{new}^α is the subspace spanned by the top α left-singular vectors of τ_{new}. It represents a dominant subspace of τ_{new}, expressed as follows:

$$\tau_{\text{new}}^\alpha = U_{\text{new}}^\alpha (U_{\text{new}}^\alpha)^\top. \tag{2}$$

U_{new}^α corresponds to the first R_α columns of U_{new}. U_{new} is obtained from the SVD (Singular Value Decomposition) decomposition of τ_{new}, *i.e.*,

$$\tau_{\text{new}} = U_{\text{new}} \sum (V_{\text{new}})^\top, \tag{3}$$

where $\sum = \text{diag}(\sigma_1, \cdots, \sigma_R)$ contains the singular values of τ_{new}. R_α is calculated by

$$\begin{aligned} R_\alpha &= \min \{R_\alpha : \|\tau_{\text{new}} - \tau_{\text{new}}^\alpha\|_F \leq (1-\alpha)\|\tau_{\text{new}}^\alpha\|_F\} \\ &= \min \left\{ R_\alpha : \frac{\sum_{r=R_\alpha+1}^{R} \sigma_r^2}{\sum_{r=1}^{R} \sigma_r^2} \leq (1-\alpha)^2 \right\}. \end{aligned} \tag{4}$$

This ensures that the approximation error between τ_{new}^α and τ_{new} is equal to or less than $1-\alpha$. Therefore, τ_{new}^α indicates the dominant subspace of τ_{new} (*e.g.*, $\alpha = 0.95$). $\text{Proj}_{\tau_{\text{new}}^\alpha} \tau_t$ is the projection of τ_t onto τ_{new}^α, calculated by

$$\text{Proj}_{\tau_{\text{new}}^\alpha} \tau_t = U_{\text{new}}^\alpha (U_{\text{new}}^\alpha)^\top \tau_t. \tag{5}$$

B. A formal analysis of task vector mixup

Let $\mathcal{M}_t$ and $\mathcal{M}_s$ denote two independent models, initialized by random model weights $\mathcal{M}_0$ and trained on two disjoint datasets $\mathcal{D}_t = \{(x_i^t, y_i^t)\}_{i=1}^{N_t}$ and $\mathcal{D}_s = \{(x_j^s, y_j^s)\}_{j=1}^{N_s}$ by optimizing a supervised loss function $\mathcal{L}$ with the SGD (Stochastic Gradient Descent) algorithm, respectively.

From an optimization perspective, task vectors are the sum of the cumulative gradient ∇ contributed by each labeled sample in iterative training, *i.e.*,

$$\begin{aligned} \tau_t &= \mathcal{M}_t - \mathcal{M}_0 = \sum_{i=1}^{N_t} \nabla \mathcal{L}(\mathcal{M}(x_i^t), y_i^t), \\ \tau_s &= \mathcal{M}_s - \mathcal{M}_0 = \sum_{j=1}^{N_s} \nabla \mathcal{L}(\mathcal{M}(x_j^s), y_j^s). \end{aligned} \tag{6}$$

We write an interpolation between τ_t and τ_s as follows:

$$\tau_{\text{mix}} = \lambda \tau_t + (1-\lambda)\tau_s, \tag{7}$$

where $\lambda \in [0, 1]$. Thus, we have

$$\begin{aligned} \tau_{\text{mix}} &= \lambda \sum_i \nabla \mathcal{L}(\mathcal{M}(x_i^t), y_i^t) + (1-\lambda) \sum_j \nabla \mathcal{L}(\mathcal{M}(x_j^s), y_j^s), \\ &= \frac{\sum_i \sum_j \left(\lambda \nabla \mathcal{L}(\mathcal{M}(x_i^t), y_i^t) + (1-\lambda) \nabla \mathcal{L}(\mathcal{M}(x_j^s), y_j^s)\right)}{N_t * N_s}. \end{aligned} \tag{8}$$

Let $\hat{y} = \mathcal{M}(x)$. When $\frac{\partial \mathcal{L}}{\partial \hat{y}}$ is a convex function *w.r.t.* $\hat{y}$, there is

$$\begin{aligned} &\lambda \nabla \mathcal{L}(\hat{y}_i^t, y_i^t) + (1-\lambda) \nabla \mathcal{L}(\hat{y}_j^s), y_j^s) \\ &\geq \nabla \mathcal{L}\left(\lambda \hat{y}_i^t + (1-\lambda)\hat{y}_j^s, \lambda y_i^t + (1-\lambda) y_j^s\right). \end{aligned} \tag{9}$$

Let $h = \mathcal{M}_E(x)$ where $\mathcal{M}_E$ is the MIL encoder of $\mathcal{M}$. Since the last layer of $\mathcal{M}$ is usually a fully-connected layer for prediction, we further have

$$\begin{aligned} &\lambda \nabla \mathcal{L}(\hat{y}_i^t, y_i^t) + (1-\lambda) \nabla \mathcal{L}(\hat{y}_j^s), y_j^s) \\ &\geq \nabla \mathcal{L}\left(\lambda h_i^t + (1-\lambda) h_j^s, \lambda y_i^t + (1-\lambda) y_j^s\right) \end{aligned} \tag{10}$$

This means that under certain conditions τ_{mix} can be cast (approximately) as the cumulative gradients obtained by training on the interpolation of any paired *MIL representation* (h_i^t, h_j^s). Although this interpolation does not directly lead to that applied to inputs (x_i^t, x_j^s) due to the nonlinearity of $\mathcal{M}$, existing studies [1, 27] have shown that representation interpolation helps to train a VRM-based model with better generalization ability.

C. More details on experimental settings

C.1. Datasets

There are 13 WSI datasets used in experiments. We show their characteristics in Tab. S1. Each dataset is split into 5 folds for cross-validation in algorithm evaluation, derived from [15]. Histopathology WSIs are processed by UNI2-h [6]. Specifically, for one image, tissue regions are segmented and tiled into image patches with 256 × 256 pixels at 20× magnification, followed by patch feature extraction using UNI2-h. The feature extraction converts each patch into a 1,536-dimensional feature vector.

C.2. Implementation details

C.2.1. Model implementation and training

For any prognostic model obtained by traditional cancer-specific training, *i.e.*, $\mathcal{M}_t$ or any $\mathcal{M}_s$, its implementation

Table S1. **Characteristics of WSI datasets**.

Dataset	# Patients	# WSIs	# Patches (avg.)	# Patches (max)	Survival rate
BRCA	1,035	1,106	10,694	58,176	92.4%
KIPAN	880	910	10,117	72,795	83.6%
LUNG	848	924	13,199	220,294	78.2%
GBMLGG	841	1,623	16,209	216,885	52.4%
COADREAD	568	577	6,458	38,322	86.6%
STES	512	539	8,429	31,458	72.7%
UCEC	504	565	14,203	73,141	89.3%
HNSC	427	449	6,530	56,558	71.7%
SKCM	412	453	9,593	136,934	60.7%
BLCA	372	443	14,465	103,807	68.5%
LIHC	355	362	8,875	35,839	77.5%
CESC	266	276	6,212	54,021	81.6%
SARC	248	591	27,346	243,906	68.5%

and network training follow Song et al. [25].

Specifically, the network is implemented by an ABMIL architecture with an MLP as its instance embedding layer $f_{\text{emb}}(\cdot)$, a gated attention layer $f_{\text{attn}}(\cdot)$ for multi-instance aggregation, and a fully-connected layer as its prediction head at the end. Each model is trained under the following settings: 20 epoch numbers, a learning rate of 0.0001 with a cosine annealing schedule (the epoch number of warming up is set to 1), an optimizer of AdamW with a weight decay of 0.00001, and a batch size of 1 (one bag) with 16 bags for gradient accumulation, and a frequently-adopted NLL loss for supervised learning in survival modeling [43]. For the other models presented in Tab. 1, their training settings are identical to those stated above. All experiments are run on two NVIDIA GeForce RTX 3090 GPUs.

For knowledge transfer methods, *i.e.*, representation-based and model merging-based baselines, the $\mathcal{M}_s$ used in experiments is a prognostic model trained on the first data split in $\mathcal{D}_s$; $\mathcal{M}_t$ is from the data split corresponding to current training fold to prevent data leakage. For our STEPH, the hyper-parameter γ is derived by a grid search over $\{0, 0.0025, 0.005, 0.01, 0.05\}$ through cross-validation on training samples. It is set to 0.005 for the dataset with over 500 patients and 0.05 for the others. The other hyper-parameters, K and β, are set to default values across all datasets without hyper-parameter searching, as stated in the main paper. For more details, please refer to our source code (see Supplementary File).

C.2.2. Efficiency evaluation

In Fig. 2, we evaluate the inference overhead of different models using a bag of 10,000 instances. Note that a bag of 10,000 instances is a general case for gigapixel WSIs, as shown in Tab. S1.

C.2.3. Loss landscape visualization

For τ_s and τ_t, we uniformly sample task vector coefficients (denoted as C_s and C_t) from $[0, 1]$ with a step size of 0.04. Given a group of coefficients (C_s, C_t), we obtain a merged model by $\mathcal{M}_{\text{new}} = \mathcal{M}_0 + C_s\tau_s + C_t\tau_t$ and then measure its $\mathcal{L}_{\text{test}}$ on t. Based on the test loss under different coefficient combinations, we generate a full loss landscape using a Gaussian smooth with $\sigma = 1.0$.

D. Additional experimental results

D.1. Dynamics of λ and w in training

We provide more results on the changes of λ and w in training. As shown in Fig. S1 (t = KIPAN), four models of other cancers (HNSC, STES, CESC, and SARC) are leveraged in task vector mixup ($\lambda_i < 0.8$) and aggregation ($w_i > 0$). When t = LUNG (Fig. S2), there are three models of other cancers (HNSC, STES, and CESC) involved more in $\mathcal{M}_t^*$.

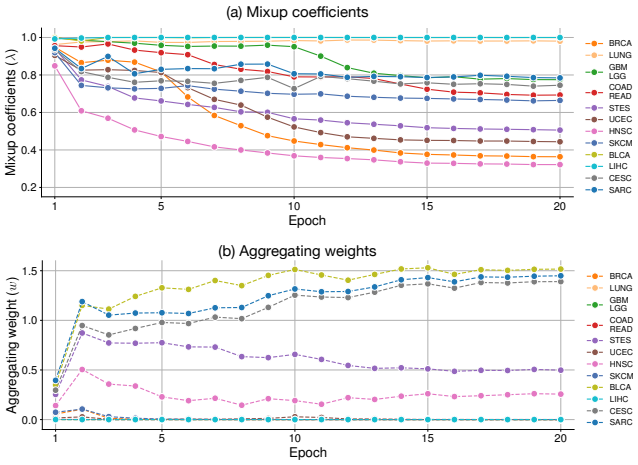

Figure S1. **Dynamics of λ and w in training** (t = KIPAN).

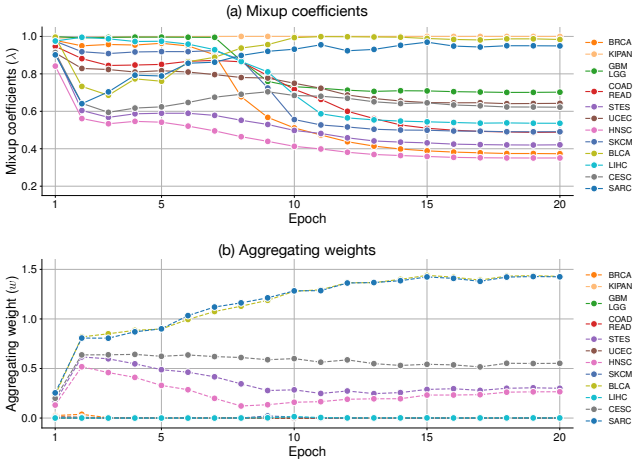

Figure S2. **Dynamics of λ and w in training** (t = LUNG).

Table S2. **Comparison with MIL networks for WSI-based survival prediction** on 13 cancer datasets.

Method	BRCA	KIPAN	LUNG	GBM LGG	COAD READ	STES	UCEC	HNSC	SKCM	BLCA	LIHC	CESC	SARC	Avg.
ABMIL	0.6648 (± 0.032)	0.8094 (± 0.013)	0.5496 (± 0.034)	0.7756 (± 0.020)	0.6725 (± 0.028)	0.6648 (± 0.040)	0.7098 (± 0.043)	0.6201 (± 0.047)	0.5708 (± 0.037)	0.6438 (± 0.015)	0.7265 (± 0.048)	0.6500 (± 0.058)	0.5312 (± 0.056)	0.6609
TransMIL	0.6802 (± 0.026)	0.8121 (± 0.018)	0.5669 (± 0.021)	**0.7873** (± 0.019)	0.6651 (± 0.059)	0.6582 (± 0.056)	0.7471 (± 0.072)	0.6315 (± 0.014)	0.5786 (± 0.039)	0.6173 (± 0.029)	0.7385 (± 0.044)	0.6365 (± 0.068)	0.5152 (± 0.042)	0.6642
ILRA	0.6645 (± 0.026)	0.8017 (± 0.016)	0.5374 (± 0.062)	0.7739 (± 0.016)	0.6147 (± 0.041)	0.6451 (± 0.053)	0.7442 (± 0.057)	0.5262 (± 0.024)	0.5586 (± 0.042)	**0.6608** (± 0.046)	0.7231 (± 0.044)	0.6336 (± 0.016)	0.5351 (± 0.055)	0.6476
R^{2}T-MIL	0.6619 (± 0.040)	0.8071 (± 0.021)	0.5805 (± 0.055)	0.7775 (± 0.019)	0.6843 (± 0.039)	0.6472 (± 0.033)	0.7253 (± 0.064)	0.6169 (± 0.037)	0.5376 (± 0.056)	0.6014 (± 0.024)	0.7163 (± 0.039)	0.6386 (± 0.090)	0.5326 (± 0.032)	0.6559
Patch-GCN	0.7017 (± 0.024)	0.8181 (± 0.009)	**0.5850** (± 0.027)	0.7853 (± 0.019)	0.6675 (± 0.041)	0.6575 (± 0.035)	0.7369 (± 0.058)	0.6019 (± 0.037)	0.5770 (± 0.040)	0.6164 (± 0.025)	0.7487 (± 0.062)	0.6585 (± 0.065)	0.5736 (± 0.034)	0.6714
STEPH	**0.7408** (± 0.060)	**0.8228** (± 0.015)	0.5836 (± 0.042)	0.7763 (± 0.022)	**0.7069** (± 0.029)	**0.6698** (± 0.050)	**0.7830** (± 0.072)	**0.6569** (± 0.042)	**0.5948** (± 0.027)	0.6413 (± 0.038)	**0.7585** (± 0.036)	**0.6965** (± 0.065)	**0.6024** (± 0.070)	**0.6949**

D.2. Task vector mixup

As presented in Fig. S3, similar results are observed on additional task vector pairs. In particular, Fig. S3(c) shows a marginal benefit (*i.e.*, a relatively small improvement in $\mathcal{L}_{\text{test}}$) from task vector mixup; such pairs could be filtered in the later sparse aggregating stage.

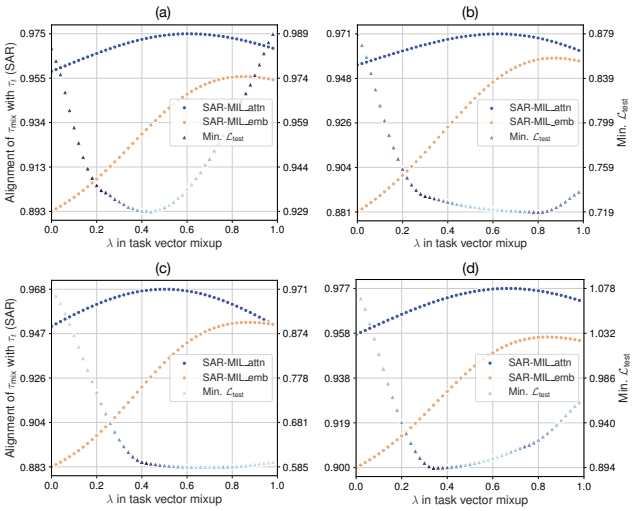

Figure S3. **Additional results of task vector mixup**.

D.3. Hyper-parameter K

As shown in Tab. S3, using more than half of task vector mixtures often leads to a performance drop. This further confirms the effectiveness of imposing sparsity on τ_{mix}.

Table S3. **STEPH with different** K (the number of selected task vector mixtures in sparse aggregation).

K	1	3	5	7	9	11	13
Avg.	0.6882	0.6929	**0.6949**	0.6935	0.6932	0.6926	0.6912

D.4. Comparison with MIL networks

We further compare STEPH with representative MIL networks to see whether it is competitive with popular MIL methods. These networks are based on the Transformer or graph convolution architecture. We observed that their original settings often lead to severe overfitting (perfect training metrics), especially on those datasets with fewer than 400 cases. Moreover, they struggle to process WSIs with over 200,000 instances (see Tab. S1) in mini-batch training. To address these issues, we adapt their settings to this task for stable training and apply patch sampling to ultra-large WSIs. Refer to our source code for more details.

From Tab. S2 and Tab. S4, we observe that STEPH outperforms existing MIL networks, with an overall improvement of 3.5% over the best baseline on 13 datasets. Besides, Patch-GCN [4] is a competitive network in terms of predictive performance and model efficiency. This implies that merging Patch-GCN models may lead to better performance. We leave it as our future work.

Table S4. **Comparison with MIL networks in terms of model efficiency**. A WSI with 10,000 patches is used in model inference.

Method	Learnable param. (M)	Test-time inference (GFLOPs)
ABMIL	**1.20**	23.6
TransMIL	2.95	60.3
ILRA	4.21	55.3
R^{2}T-MIL	2.98	40.7
Patch-GCN	1.69	**21.7**
STEPH	1.34	40.1

D.5. Complete ablation results on all datasets

We present the complete results of key ablation components in Tab. S5. The contribution of key designs can be observed in most cases: hypernetwork-driven λ on 11 out of 13 datasets and w on 12 out of 13 datasets, TVM over $\lambda = 1$ on 8 out of 13 datasets, and sparse aggregation on 11 out of 13 datasets.

Table S5. Complete ablation results on 13 datasets.

Ablation	BRCA	KIPAN	LUNG	GBM LGG	COAD READ	STES	UCEC	HNSC	SKCM	BLCA	LIHC	CESC	SARC	**Avg.**
w/o mixup ($\lambda = 1$)	0.7183 (±0.051)	0.8264 (±0.016)	0.5876 (±0.039)	0.7745 (±0.024)	0.6971 (±0.024)	0.6781 (±0.030)	0.7840 (±0.069)	0.6420 (±0.022)	0.5954 (±0.037)	0.6387 (±0.038)	0.7400 (±0.046)	0.6886 (±0.043)	0.5358 (±0.070)	0.6851
w/o $\mathcal{H}_{\text{mix}}$ (param. λ)	0.7323 (±0.061)	0.8152 (±0.019)	0.5901 (±0.040)	0.7728 (±0.024)	0.7044 (±0.027)	0.6645 (±0.047)	0.7852 (±0.073)	0.6548 (±0.041)	0.5903 (±0.031)	0.6404 (±0.038)	0.7579 (±0.034)	0.6951 (±0.066)	0.5944 (±0.066)	0.6921
w/o sparse	0.7263 (±0.052)	0.8212 (±0.016)	0.5865 (±0.043)	0.7746 (±0.021)	0.7009 (±0.027)	0.6640 (±0.055)	0.7803 (±0.075)	0.6526 (±0.038)	0.5891 (±0.024)	0.6433 (±0.041)	0.7541 (±0.034)	0.6937 (±0.063)	0.5994 (±0.086)	0.6912
w/o $\mathcal{H}_{\text{agg}}$ (param. w)	0.7133 (±0.015)	0.8042 (± 0.014)	0.5417 (±0.033)	0.7800 (±0.017)	0.6564 (±0.033)	0.5984 (±0.036)	0.7359 (±0.049)	0.5266 (±0.037)	0.5753 (±0.032)	0.6289 (±0.049)	0.7035 (±0.034)	0.6424 (±0.025)	0.5299 (±0.030)	0.6490
STEPH	0.7408 (±0.060)	0.8228 (±0.015)	0.5836 (±0.042)	0.7763 (±0.022)	0.7069 (±0.029)	0.6698 (±0.050)	0.7830 (±0.072)	0.6569 (±0.042)	0.5948 (±0.027)	0.6413 (±0.038)	0.7585 (±0.036)	0.6965 (±0.065)	0.6024 (±0.070)	**0.6949**

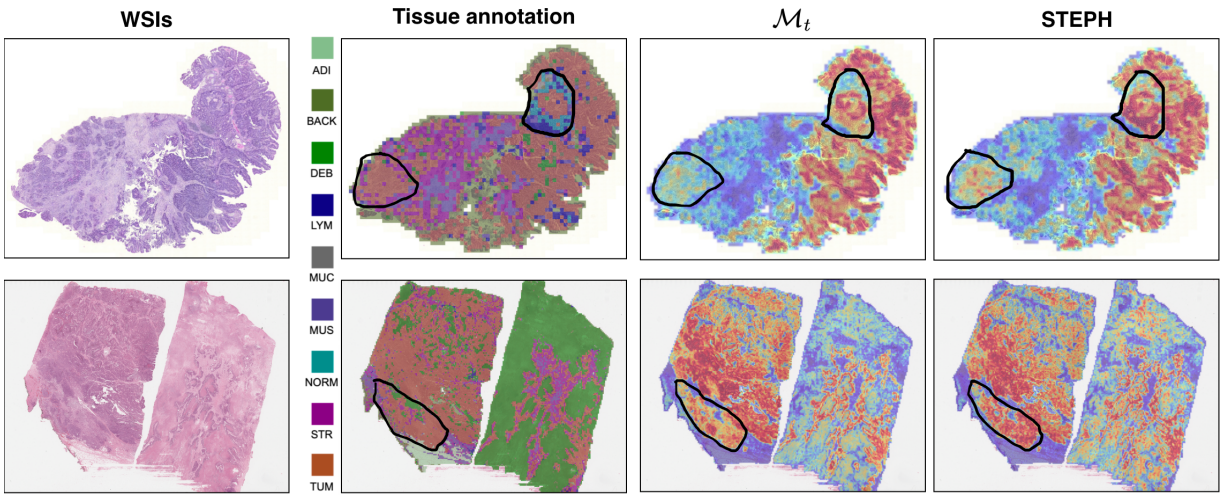

Figure S4. Visualization of attention maps given by models. The two WSIs are from the test set of COADREAD. Tumor (TUM).

D.6. Training efficiency

Comparison with other model merging methods To show the training efficiency of STEPH, we measure the mean training time per epoch on BRCA and report it for all model merging methods. Tab. S6 shows that STEPH only introduces marginal training costs compared to the methods with AdaMerging.

Table S6. Mean training time per epoch on BRCA.

Method	Model Avg.	AdaMerging	TIES AM	Surgery AM	Iso-C AM	STEPH
Time (s)	0.00 (± 0.00)	78.08 (± 0.92)	78.42 (± 2.20)	78.00 (± 1.34)	78.50 (± 1.54)	**80.45** (± 1.63)

Increasing the number of source models We further vary the number of source models (denoted by m) to show the training efficiency of STEPH. The number of chosen task vector mixtures, K, is set to be equal to m, *i.e.*, all source models are utilized. Tab. S7 shows that increasing m would not introduce too much training time.

Table S7. Mean training time per epoch on BRCA with different numbers of source models (m).

m (K)	6	7	8	9	10	11	12
Time (s)	72.91 (± 1.20)	72.98 (± 1.42)	74.51 (± 3.00)	75.62 (± 1.54)	75.60 (± 1.62)	75.74 (± 2.18)	**76.69** (± 1.79)

D.7. Interpretability

We visualize and compare the attention maps given by $\mathcal{M}_t$ and STEPH, following the setting of [15]. From Fig. S4, we observe that STEPH captures more prognostic cues: tumor areas and the boundary of tumor infiltration.

E. More discussions

Comparison with ROUPKT [15] STEPH is based on model merging, whereas ROUPKT is based on representation transfer (as shown in Fig. 1). Two methods are distinct in design philosophy, though both are used for knowledge transfer. In the comparative results given by Tab. 1, ROUPKT performs better than STEPH on 4 datasets. We

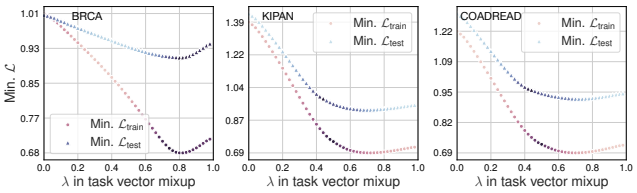

Figure S5. Loss of TVM between BLCA and others (w/ top w).

examine the cause of the largest performance gap, BLCA, by analyzing the loss landscape of TVM. From Fig. S5, we observe that λ with the best $\mathcal{L}_{\text{test}}$ is often larger (0.7$\sim$0.8). It would be helpful to decrease the degree of penalty by setting a smaller β for $\mathcal{L}_{\text{mix}}$ to encourage a larger λ. However, β is simply set to 0.05 for all datasets in this paper without fine-tuning. Fine-tuning β on BLCA may improve the performance of STEPH.

Choice of foundation models UNI2-h [6] is adopted because *i*) it is trained with private data (not TCGA) so data leakage can be avoided and *ii*) it has demonstrated excellent generalizability. Other models, like CONCH [17], are also competitive. Yet, one more foundation model leads to substantial computational costs since there are 8,818 WSIs (>100 million patches) and 5-fold cross-validation experiments on 13 datasets. Given these, we choose one of the representatives.

F. Reproducibility statement

To ensure reproducibility, we provide detailed descriptions of the datasets in Sec. C.1 and comprehensive implementation specifics in Sec. C.2. Our source code is publicly available at `https://github.com/liupei101/STEPH`. It includes the checkpoints and training logs of our model.