This MICCAI paper is the Open Access version, provided by the MICCAI Society. It is identical to the accepted version, except for the format and this watermark; the final published version is available on SpringerLink.

Supplementary Material

1 Dataset and Cross-Validation Statistics

All local validation results use the same prevalence-preserving, patient-disjoint five-fold partition described in the main paper, unless noted otherwise. Table 1 summarizes the per-fold class distribution in terms of patch counts and distinct contributing patients.

Table 1. Per-fold class distribution of the prevalence-preserving split. Each cell reports the number of patches/distinct patients. The split is patient-disjoint across folds.

Fold	CT	DM	IC	LI	MP	NC	PL	PN	WM	NOTA
1	21173/18	1507/2	8532/11	2026/2	4092/8	16485/15	55/2	7060/14	5246/8	4268/1
2	21173/20	1507/2	8532/11	2026/2	4092/9	16485/15	521/1	7060/14	5246/9	3482/1
3	21173/22	1507/3	8532/11	2026/2	4092/8	16485/15	1620/1	7060/15	5246/8	8624/1
4	21173/26	1507/3	8532/9	203/2	4092/8	16485/14	336/1	7060/14	5246/8	12259/1
5	21173/16	1385/2	8532/6	445/2	1364/7	16485/10	61/2	7060/11	5246/6	12259/1

2 WM/IC Router Motivation and Characterization

2.1 WM/IC Router Motivation

Before routing, the dominant residual error in the ensemble is the mutual confusion between WM and IC. On pooled OOF predictions from the calibrated M1+M2 ensemble without routing, 24.5% of ground-truth WM patches are predicted as IC and 14.1% of IC patches as WM. IC accounts for 54% of all WM errors, while WM accounts for 63% of all IC errors. This concentrated, bidirectional confusion motivates targeted routing of the WM/IC pair. It also motivates examining whether the fused embeddings retain WM/IC-discriminative information that is not fully captured by the multiclass prediction head.

To assess this, we compare the WM–IC discrimination AUROC of the multiclass head's within-pair margin, $P(\text{IC})/[P(\text{IC})+P(\text{WM})]$, with embedding-based logistic-regression probes after PCA projection (Table 2). Evaluation uses pooled patient-disjoint OOF predictions over all ground-truth WM and IC patches and over the subset for which WM and IC are the top two predicted classes for at least one ensemble member ("Triggered WM/IC").

The PCA-2 probe achieves AUROC 0.8688 over all ground-truth WM/IC patches and 0.8623 over the triggered subset, exceeding the corresponding multiclass margin values of 0.8456 and 0.8379. This indicates that a simple embedding-based readout provides stronger WM/IC discrimination than the multiclass

within-pair margin. We further evaluate higher PCA dimensionalities: AUROC decreases from 0.8688 (2 components) to 0.8548 (8) and 0.7884 (16), with the same trend on the triggered subset. These results suggest that the WM/IC-discriminative signal is concentrated in a low-dimensional subspace, while additional components may introduce less informative variation, supporting the use of the PCA-2 probe in the routing design.

Table 2. WM–IC discrimination AUROC on pooled patient-disjoint OOF predictions for the multiclass within-pair margin and embedding probes with varying PCA dimensionality.

Method	**All WM/IC**	**Triggered WM/IC**
Multiclass margin (blend)	0.8456	0.8379
Embedding probe, PCA-2	**0.8688**	**0.8623**
Embedding probe, PCA-8	0.8548	0.8479
Embedding probe, PCA-16	0.7884	0.7759

2.2 WM/IC Router Characterization

To characterize the behavior of the WM/IC routing gate, we further analyze its trigger frequency, the composition of triggered patches, and its effect on WM/IC classification. All results are computed from pooled patient-disjoint OOF predictions for M1+M2, comparing predictions before and after routing.

Routing was triggered for 52,184 of 367,836 patches (14.2%), corresponding to cases where WM and IC were the top two predicted classes for at least one member. Of the triggered patches, 49,156 (94.2%) had a ground-truth label of WM or IC, whereas 3,028 (5.8%) belonged to another class.

Table 3 reports the class-specific effect of routing. The router increases WM F1 from 0.5548 to 0.5725 and IC F1 from 0.7855 to 0.8061. Precision and recall also increase for both WM and IC after routing.

Table 3. WM/IC performance before and after routing on pooled five-fold OOF predictions.

Class	**Configuration**	**Precision**	**Recall**	**F1**
WM	Before router	0.5635	0.5464	0.5548
	After router	0.5939	0.5526	0.5725
IC	Before router	0.7948	0.7764	0.7855
	After router	0.8058	0.8065	0.8061

To further examine the WM↔IC errors targeted by the router, Table 4 reports the corresponding confusion counts over ground-truth WM/IC patches predicted as WM or IC.

Table 4. WM/IC confusion counts before and after routing on pooled OOF predictions for ground-truth WM/IC patches classified as WM or IC.

Configuration	WM→WM	WM→IC	IC→WM	IC→IC
Before router	14331	6434	6031	33120
After router	14495	6299	4741	34405

Routing reduces WM→IC errors from 6434 to 6299 (−2.1%) and IC→WM errors from 6031 to 4741 (−21.4%), while increasing correct WM and IC assignments by 164 (+1.1%) and 1285 (+3.9%), respectively. The larger reduction in IC→WM confusion indicates that the router is particularly effective at recovering IC cases that would otherwise be assigned to WM.

Of the 52,184 triggered patches, 3,028 (5.8%) have a ground-truth label outside the WM/IC pair. These off-target triggers are dominated by CT (2,446; 80.8%) and MP (411; 13.6%), with only small contributions from DM, PN, LI, NC, and NOTA. Because routing only reallocates probability within the WM/IC pair, it cannot directly increase the probability assigned to the true non-WM/IC class at the member level. At the M1+M2 ensemble level, routing changes the final prediction for 587 of these patches and yields a net gain of 19 correct classifications. Thus, off-target triggering does not degrade aggregate performance on this subset and is marginally beneficial overall. Predictions whose top two classes do not form the WM/IC pair are not routed, leaving other class-pair decisions unchanged. Empirically, precision, recall, and F1 for LI and PL remain unchanged before and after routing.

3 Fusion Ablation

We further evaluate individual foundation-model encoders, alternative fusion strategies, and successive refinements of the learned-fusion pipeline. All results are computed from pooled OOF predictions under the same patient-disjoint five-fold protocol used for local validation. The AUROC and specificity are macro-averaged.

3.1 Individual encoder baselines

We first evaluate each LoRA-finetuned pathology encoder in the full pipeline independently, without ensembling, probability calibration, or WM/IC routing. Table 5 reports their performance on pooled five-fold OOF predictions. The prevalence-trained UNI2-h achieves the highest accuracy (0.8038), MCC

(0.7597), and specificity (0.9761), whereas the class-balanced UNI2-h yields the highest AUROC (0.9271), macro-F1 (0.6236), and recall (0.6364). Virchow2 provides competitive accuracy and AUROC but lower macro-F1 and recall. These differences indicate complementary performance profiles across encoders and training priors, motivating their subsequent combination.

Table 5. Performance of individual LoRA-finetuned encoders on pooled five-fold OOF predictions. Bold indicates the best local cross-validation result per metric.

Encoder	Acc.	MCC	AUROC	macro-F1	Recall	Spec.
Virchow2 (prev.)	0.7978	0.7523	0.9039	0.5923	0.5803	0.9755
UNI2-h (prev.)	**0.8038**	**0.7597**	0.8718	0.6180	0.5996	**0.9761**
UNI2-h (balanced)	0.7818	0.7389	**0.9271**	**0.6236**	**0.6364**	0.9749

We further evaluate two additional pathology encoders to broaden the single-model comparison in Table 5. First, we evaluate the frozen CONCH representation used by the final fusion model with its logistic-regression readout. As a standalone classifier, CONCH performs below each LoRA-finetuned pathology encoder across the reported metrics (Table 6). Second, we evaluate H-optimus-0, another pathology foundation model, using the prevalence-preserving training configuration and LoRA fine-tuning. For each fold, the checkpoint with the highest held-out macro-F1 is retained.

Table 6. Performance of the frozen CONCH representation with its logistic-regression readout and the LoRA-finetuned H-optimus-0 encoder on pooled five-fold OOF predictions.

Encoder	Acc.	MCC	AUROC	macro-F1	Recall	Spec.
CONCH (frozen)	0.7638	0.7097	0.8933	0.5429	0.5231	0.9709
H-optimus-0 (LoRA)	0.7989	0.7536	0.8825	0.5972	0.5866	0.9754

H-optimus-0 achieves performance comparable to the individual Virchow2 and UNI2-h pathology encoders, but was not retained in the final ensemble. In our development experiments, adding H-optimus-0 to the existing encoder ensemble provided only a marginal improvement in pooled OOF performance. We therefore retained the existing ensemble configuration. This choice also reduced inference cost: official evaluation is performed on a single 48 GB GPU with a 24-hour runtime limit over more than 600k patches, while the submitted five-fold, two-member system was estimated to require approximately 16 hours. Adding another comparatively large backbone would require an additional encoder forward pass for each patch and fold, increasing the risk of exceeding the evaluation-time budget for the small observed OOF gain.

3.2 Fusion baselines

We next evaluate alternative fusion baselines, including optimized probability averaging and logistic-regression (LR) stacking. Table 7 reports three fusion baselines. The three-encoder configuration (3 enc.) uses Virchow2 together with the prevalence-preserving and class-balanced UNI2-h models, while the four-encoder configuration (4 enc.) additionally includes CONCH. The first two rows compare optimized probability averaging without and with CONCH, respectively, and the third replaces probability averaging with LR stacking over the same four encoder predictions. Details of the probability-averaging and LR-stacking baselines are described below.

Probability averaging. For each configuration, member weights are selected by grid search over the set of weights summing to one on the pooled five-fold OOF predictions, maximizing MCC. Because weight selection and evaluation use the same pooled OOF predictions, this constitutes an optimistically tuned averaging baseline.

LR stacking. We evaluate LR stacking as an alternative learned prediction-level fusion baseline for comparison with the proposed fusion head. For each patch, the log class-probabilities from all ensemble members are concatenated, standardized, and used to train an ℓ_2-regularized multinomial logistic regression. We use the same five-fold partition summarized in Table 1, fitting the stacker on predictions from four folds and applying it to the remaining held-out fold. The five held-out folds are then pooled for evaluation.

Table 7. Performance of fusion baselines on pooled five-fold OOF predictions.

Configuration	Acc.	MCC	AUROC	macro-F1	Recall	Spec.
Prob. average (3 enc.)	0.8145	0.7734	0.9203	0.6310	0.6214	0.9777
Prob. average (4 enc.)	0.8160	0.7750	0.9215	0.6313	0.6195	0.9778
LR stack (4 enc.)	0.7659	0.7145	0.8930	0.5479	0.5451	0.9721

Adding CONCH to the optimized three-encoder probability ensemble yields small improvements in accuracy, MCC, AUROC, macro-F1, and specificity, with a slight decrease in recall. This suggests that CONCH provides modest complementary information even under simple prediction-level fusion. In contrast, LR stacking underperforms probability averaging across all metrics, indicating that a linear prediction-level meta-classifier does not provide an advantage over optimized probability averaging in this setting.

3.3 Learned-fusion and incremental component ablations

To evaluate the contribution of each component in the fusion pipeline, we perform a controlled incremental ablation in which components are introduced se-

quentially while the remaining setup is held fixed. Table 8 reports the resulting changes from random initialization through warm-starting with low-rank head adaptation, calibration, and WM/IC routing. Because the low-rank head update is defined as an adaptation around the frozen pretrained classification heads, it is introduced jointly with warm-starting. The random-initialization baseline instead uses a conventionally trainable fusion head. We additionally remove CONCH from the complete system while retaining the remaining training, calibration, routing, and ensemble procedure to assess its end-to-end contribution.

Table 8. Incremental learned-fusion and end-to-end CONCH ablations on pooled five-fold OOF predictions. Parentheses denote changes from the preceding row, except for "- CONCH," which is compared against the M1+M2.

Configuration	**Acc.**	**MCC**	**AUROC**	**macro-F1**
Random init	0.814	0.773	0.925	0.632
+ warm-start LoRA	0.818 (+0.004)	0.778 (+0.005)	0.919 (−0.006)	0.639 (+0.007)
+ calibration	0.818 (< 0.001)	0.777 (< 0.001)	0.932 (+0.013)	0.639 (< 0.001)
+ router (M1+M2)	0.822 (+0.004)	0.782 (+0.005)	0.934 (+0.002)	0.643 (+0.004)
- CONCH	0.818 (−0.004)	0.778 (−0.005)	0.929 (−0.005)	0.641 (−0.002)

"Warm-start LoRA" denotes low-rank adaptation of the frozen pretrained classification heads introduced jointly with warm-start.

Warm-starting LoRA improves accuracy, MCC, and macro-F1 over random initialization, with a modest decrease in AUROC. Calibration primarily improves AUROC, from 0.919 to 0.932, while leaving the other metrics nearly unchanged. Routing subsequently improves all four reported metrics, yielding the strongest overall performance in the incremental sequence. Removing CONCH from the full model reduces accuracy, MCC, AUROC, and macro-F1 by 0.004, 0.005, 0.005, and 0.002, respectively, indicating a consistent end-to-end contribution from the vision–language representation.

4 LI/PL Tail-Recovery Ablation

The final pipeline applies a targeted tail-recovery strategy to the patient-limited LI and PL classes, combining oversampling of LI/PL training patches with stronger augmentation of the oversampled samples. To evaluate its contribution, we remove this strategy while holding the remainder of the training pipeline fixed. We then form the $M_1 + M_2$ ensemble using the same calibration and WM/IC routing procedure and evaluate pooled patient-disjoint OOF predictions. Table 9 reports per-class and macro-F1 with and without LI/PL tail recovery.

The tail-recovery strategy improves macro-F1 by 0.0110, with the largest gains observed for the targeted classes, PL (+0.0403) and LI (+0.0296). Addi-

Table 9. Per-class and macro-F1 with and without LI/PL tail-recovery strategy at the $M_1 + M_2$ ensemble level on pooled five-fold OOF predictions.

Class	**w/o tail recovery**	**w/ tail recovery**	**ΔF1**
CT	0.8448	0.8465	+0.0017
DM	0.6799	0.6973	+0.0174
IC	0.8069	0.8061	−0.0008
LI	0.1094	0.1390	+0.0296
MP	0.5320	0.5191	−0.0129
NC	0.9260	0.9256	−0.0004
PL	0.1026	0.1429	+0.0403
PN	0.7794	0.7894	+0.0100
WM	0.5454	0.5725	+0.0271
NOTA	0.9926	0.9910	−0.0016
Macro-F1	0.6319	0.6429	**+0.0110**

tional improvements are observed for WM (+0.0271), DM (+0.0174), and PN (+0.0100). The largest decrease is observed for MP (−0.0129), while IC, NC, and NOTA change only marginally.

Table 10 reports the corresponding aggregate metrics. Removing the tail-recovery strategy decreases all six metrics, with the largest differences in macro-F1 (−0.0110) and recall (−0.0100), followed by AUROC (−0.0060) and MCC (−0.0031). Together with the class-specific results, this indicates that the strategy primarily benefits the rare LI/PL classes while also improving overall performance.

Table 10. Overall effect of LI/PL tail recovery on pooled five-fold OOF predictions for the $M_1 + M_2$ ensemble.

Configuration	**Acc.**	**MCC**	**AUROC**	**macro-F1**	**Recall**	**Spec.**
w/o tail recovery	0.8194	0.7792	0.9277	0.6319	0.6222	0.9782
Δ vs. w/ tail recovery.	−0.0024	−0.0031	−0.0060	−0.0110	−0.0100	−0.0003

5 Sub-patch Resolution Analysis

To evaluate whether resizing from 512×512 to 224×224 may discard fine-scale cues relevant to morphologically confusable classes, we compare the standard full-patch view with a higher-effective-resolution sub-patch formulation using the same encoder and training setup.

We evaluate UNI2-h on the balanced dataset for this comparison. In the standard configuration, each 512×512 patch is resized directly to 224×224. In

the sub-patch configuration, each patch is divided into a 2×2 grid of 256×256 tiles, each resized to 224×224 and processed independently. The four tile logits are aggregated by log-sum-exp (LSE) multiple-instance pooling to produce a single patch-level prediction. The same aggregation is used during training and inference, with the encoder jointly LoRA-finetuned under the LSE pooling scheme. All other training settings are held fixed across the two configurations. Evaluation uses the same five-fold cross-validation on the balanced dataset. Table 11 reports overall performance, while Table 12 reports per-class F1 for the full-patch and sub-patch configurations.

Table 11. UNI2-h performance for the standard 224px full-patch resize and 2×2 sub-patch formulation under five-fold cross-validation on the balanced dataset. Δ denotes sub-patch − full-patch resize.

Configuration	Acc.	MCC	AUROC	macro-F1	Recall	Spec.
Full patch resize	0.6992	0.6656	0.9365	0.6407	0.6514	0.9663
2×2 sub-patch	0.6837	0.6489	0.9339	0.6118	0.6314	0.9645
Δ	−0.0155	−0.0167	−0.0026	−0.0289	−0.0200	−0.0018

Table 12. Per-class F1 for full-patch resize and 2×2 sub-patch processing under five-fold cross-validation on the balanced dataset. Δ denotes sub-patch minus full-patch resize.

Configuration	CT	DM	IC	LI	MP	NC	PL	PN	WM	NOTA
Full-patch resize	0.617	0.761	0.726	0.269	0.681	0.778	0.227	0.776	0.601	0.972
2×2 sub-patch	0.609	0.725	0.715	0.180	0.643	0.760	0.139	0.753	0.605	0.990
Δ	−0.008	−0.036	−0.011	−0.089	−0.038	−0.018	−0.088	−0.023	+0.004	+0.018

The sub-patch formulation does not improve performance over the standard full-patch view. The largest class-specific F1 decreases are observed for LI (−0.089) and PL (−0.088), followed by MP (−0.038) and DM (−0.036). CT (−0.008) and WM (+0.004) remain largely unchanged, while NOTA shows a notable improvement (+0.018).

Contrary to the intended benefit of the sub-patch formulation, the additional local detail does not translate into improved performance. One possible explanation is that the increased local detail provided by the sub-patches is offset by the loss of whole-patch spatial context and by the weak-label aggregation required across tiles. Overall, these results provide no evidence that recovering finer spatial detail through the tested sub-patch approach improves the rare or morphologically confusable classes. We note, however, that sub-patch processing may benefit from a more carefully designed training and aggregation protocol.

6 CONCH concept bank and prompt templates

This section lists the concept descriptions and prompt templates used to construct the CONCH-derived vision–language features described in the main paper. The concept bank and associated phrasings were defined from the class definitions and pathology domain knowledge. No model evaluation results were used to select or revise the concepts.

6.1 Prompt templates

The four templates are {}, `an H&E image of` {}, `histopathology showing` {}, and `a pathology image of` {}. Each phrasing associated with a concept is instantiated using all four templates, and the resulting text embeddings are averaged to form the concept prototype.

6.2 Disease-related concepts

Each of the nine target classes is represented by one concept, with multiple phrasings capturing characteristic morphology and distinctions between morphologically related classes.

- **CT**: densely cellular glioma; hypercellular neoplastic astrocytes; sheets of atypical tumor cells; high-grade astrocytoma with nuclear pleomorphism; brisk mitotic activity.
- **PN**: pseudopalisading necrosis; tumor cells palisading around necrosis; serpiginous necrosis rimmed by dense tumor nuclei; cells lined up along necrotic edge.
- **NC**: geographic necrosis; large map-like necrotic zone; confluent bland necrosis without palisading; acellular necrotic tissue.
- **MP**: microvascular proliferation; glomeruloid vascular structures; endothelial proliferation forming tufts; multilayered proliferating blood vessels.
- **IC**: tumor infiltrating cerebral cortex; neoplastic cells among cortical neurons; gray matter infiltrated by glioma; neuronal satellitosis by tumor cells.
- **WM**: tumor infiltrating white matter; glioma in white matter tracts; infiltration along myelinated fibers; hypercellular white matter with tumor cells.
- **LI**: leptomeningeal infiltration; tumor cells in the meninges; subarachnoid space infiltrated by glioma; tumor spreading along pial surface.
- **DM**: sheets of macrophages; foamy histiocytes; lipid-laden macrophages; macrophage-rich region; histiocytic infiltrate.
- **PL**: perivascular lymphocytic cuffing; scattered lymphocytes; small round lymphocytes around vessels; chronic inflammatory infiltrate.

6.3 Background and artifact concepts

Nine additional concepts provide background and artifact prototypes for distinguishing NOTA from the disease-related classes: normal brain parenchyma; normal cortex with neurons; normal white matter; blood vessel; hemorrhage ("hemorrhage", "red blood cells"); staining artifact; tissue folding artifact; blank slide background; and surgical cautery artifact.

7 Training and Inference Parameters

All LoRA-finetuned pathology encoders in the final ensemble follow a common training framework, with the training dataset, number of epochs, and fold-specific loss hyperparameters varying by configuration. Table 13 summarizes the shared training settings, while Table 14 reports the fold-specific loss parameters and checkpoint-selection criteria.

Table 13. Shared training configuration for the LoRA-finetuned pathology encoders.

Component	**Setting**
Backbones	Virchow2, UNI2-h
Trainable parameters	LoRA adapters + linear classification head
LoRA rank / α	r=4, α=8
LoRA target modules	`blocks.{i}.attn.qkv` (both backbones)
Token pooling	CLS token + mean-pooled patch tokens
Optimizer	AdamW, weight decay 0.01
Learning rate	1×10^{-5}
LR schedule	cosine annealing to 0 over the training epochs; stepped per epoch, without warmup
Epochs	5 (prev.), 10 (balanced)
Batch size	64
Mixed precision	bfloat16 (AMP)
Loss	one-vs-rest BCE with focal term (γ) and effective-number positive weighting (ρ)
Random seed	42
Normalization / resize	backbone-specific; details below
Base augmentation	light; details below
Tail augmentation	stronger augmentation on oversampled copies; details below

"prev." and "balanced" denote the prevalence-preserving and class-balanced training configurations, respectively.

7.1 Backbone preprocessing

Each encoder uses its native `timm` preprocessing and ImageNet normalization with mean $(0.485, 0.456, 0.406)$ and standard deviation $(0.229, 0.224, 0.225)$. UNI2-

h resizes each patch directly to 224×224 using bilinear interpolation. Virchow2 resizes to 256 using bicubic interpolation and then center-crops to 224×224.

7.2 Augmentation

Every augmented view first receives shared geometric augmentation consisting of a horizontal flip (p=0.5), a vertical flip (p=0.5), and a random rotation selected from $\{0°, 90°, 180°, 270°\}$. The base augmentation, applied to all classes, additionally uses mild color jitter (brightness/contrast/saturation ± 0.05, hue ± 0.02; p=0.5). The stronger tail augmentation is applied to oversampled LI/PL copies and consists of stain-mimicking color jitter (brightness/contrast/saturation ± 0.3, hue ± 0.1; p=0.9), `RandomResizedCrop` to 224×224 (scale $[0.6, 1.0]$, aspect ratio $[0.85, 1.18]$; p=0.8), a small affine transform (translation $\pm 5\%$, shear 8°; p=0.4), and Gaussian blur (kernel 5, $\sigma \in [0.1, 2.0]$; p=0.3).

7.3 Fold-specific loss and checkpoint selection

The positive-weight exponent ρ and focal-loss parameter γ vary by fold as shown in Table 14. These settings were selected from $\rho \in \{0, 0.25, 0.5, 0.75\}$ and $\gamma \in \{0, 1, 2\}$ by a per-fold grid search: for each fold, the (ρ, γ) configuration whose checkpoint maximized the held-out selection metric was retained.

Table 14. Fold-specific loss hyperparameters and checkpoint-selection metric.

Encoder	Sel.	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
UNI2-h (balanced)	F1	ρ=0.25 γ=0	ρ=0.5 γ=0	ρ=0.5 γ=0	ρ=0.5 γ=1	ρ=0.75 γ=0
UNI2-h (prev.)	F1	ρ=0.5 γ=0	ρ=0.75 γ=2.0	ρ=0.5 γ=0	ρ=0.5 γ=0	ρ=0.5 γ=1.0
Virchow2 (prev.)	MCC	ρ=0.0 γ=2.0	ρ=0.5 γ=2.0	ρ=0.5 γ=2.0	ρ=0.75 γ=2.0	ρ=0.25 γ=2.0

"prev." and "balanced" denote the prevalence-preserving and class-balanced training configurations, respectively. "Sel." denotes the checkpoint-selection metric for each encoder configuration.

7.4 Fusion-head specification

The learned fusion heads use the architecture described in the main paper. The head dimensions are: M_1 input dimension 8704, hidden dimension 30, and output dimension 10; M_2 input dimension 9234, hidden dimension 40, and output dimension 10. The low-rank fusion update uses rank 4. The warm-started first layer

is block-diagonal and initialized from the corresponding pretrained per-encoder classification heads, while the output layer is initialized using the optimized ensemble weights described in the main paper.

7.5 Calibration

A scalar temperature is fit separately for each ensemble member and fold by minimizing negative log-likelihood on its OOF predictions and is then frozen for inference. The resulting multiclass calibration temperatures are reported in Table 15. The WM/IC router additionally uses fold-specific binary temperatures, T_{bin}, reported in Table 16.

Table 15. Fold-specific temperature-scaling parameters for the two ensemble members.

Member	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
M_1	1.654	1.120	1.231	1.379	0.710
M_2	1.683	1.234	1.292	1.383	0.852

Table 16. Fold-specific binary WM/IC temperatures (T_{bin}) used by the router to redistribute WM/IC probability mass.

Member	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
M_1	0.935	1.224	1.058	1.424	0.388
M_2	0.838	1.227	0.978	1.774	0.408

7.6 Inference-time composition

At inference, each fold checkpoint produces softmax probabilities that are first temperature-scaled using the corresponding fold-specific temperature (Table 15). The WM/IC router is then applied: when WM and IC are the top two predicted classes, the corresponding WM/IC classifier redistributes their combined probability mass using the fold-specific binary temperature in Table 16. The five calibrated and routed fold outputs are averaged within each member, after which M_1 and M_2 are combined with equal weight. All preprocessing, calibration, and routing parameters are estimated only from the corresponding training/OOF data.

8 Official Validation and Challenge Baseline

During model development, we made 46 submissions to the official BraTS-Path validation server. Patient-disjoint five-fold OOF performance remained the primary local model-development criterion, but official validation feedback was also used as a secondary criterion when deciding whether to retain subsequent modifications. Specifically, a newly introduced component or modification was retained only when it improved both local OOF performance and official validation performance.

Challenge baseline For additional challenge-level context, Table 17 compares the final submission with the organizers' reference baseline on the official validation set. The proposed system improves all six reported metrics, including accuracy from 0.771 to 0.9264, MCC from 0.689 to 0.8976, and F1 from 0.400 to 0.7364.

Table 17. Official-validation performance of the BraTS-Path challenge baseline and the final submitted model.

Model	Acc.	AUROC	F1	MCC	Recall	Spec.
Challenge baseline	0.771	0.871	0.400	0.689	0.458	0.972
Our submission	0.9264	0.9540	0.7364	0.8976	0.7481	0.9899