# SUPPLEMENTARY MATERIALS

## S1. OPTIMIZING MODELS IN PATHOLOGICAL IMAGE CLASSIFICATION

Our pipeline follows the standard process used in most pathological image analysis papers. Below, we briefly outline our model optimization process in the following steps:

1) **Prepare the dataset:** All 7 datasets are randomly separated into training, validation, and test sets following a ratio of 7:1:2. The samples are separated by patient ID, ensuring no information leak among patient similarities. Certain datasets (e.g., pRCC) present official separations; in such cases, we follow the official separation.
2) **Training, validation, and reporting rules:** On these datasets, experiments show the models converge within 50 epochs. Accordingly, we train each experiment for 50 epochs and take the model at the specific epoch with the best validation performance as the trained model (standard pipeline). Finally, the results are reported based on the model's performance on the independent test set. The Adam optimizer and cosine weight decay strategy are used in all training. In each training process, a batch size of 8 is used.
3) **On-line data pre-processing:** Additionally, for all experiments, we implement traditional pre-processing data augmentation strategies on different datasets before exploring. Two sets of pre-processing are implemented. The ROSE, MARS, and pRCC datasets are preprocessed with the first set of pre-processing-based data augmentation strategies. In the training process, random rotation and center-cut operations with a size of $700 \times 700$ pixels are applied to the input images. The input data is then randomly horizontally flipped, vertically flipped, resized, and color jittered (with settings of brightness=0.15, contrast=0.3, saturation=0.3, and hue=0.06). In the validating and testing processes, the $700 \times 700$ pixel center area of each image is cropped and resized without additional operations. Due to the size of the images, we introduce another set of pre-processing data augmentation strategies for the Warwick, GS, WBC, and NCT datasets. Different from the first set, this strategy replaces the random rotation and center-cut operation with a resizing operation in the training process. In the validation and test processes, the second strategy only includes the resize operation.
4) **Data augmentation settings:** Following most other data augmentation papers, a random data augmentation triggering chance of 50% is applied. If there are hyperparameters in the comparison methods, we apply their official default settings. Additionally, we highlight that CellMix is a plug-and-play module performing image permutation. There are no model parameters required, and the calculation is performed after data fetching in the PyTorch data-loader. It can be seamlessly integrated into any classification training pipeline without requiring additional GPU time.

Then, we reveal the hyperparameters that need to be optimized in this work. As a special notice, pathological image datasets are generally small, and the most pivotal hyperparameter is the learning rate, according to our knowledge. Accordingly, we applied an empirical set of learning rates (from 0.0001 to 0.0005; 0.00001 to 0.00005; 0.000001 to 0.000005; 0.0000001 to 0.0000005) and cosine weight decay factors (0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5 times). Specifically, all combinations of these were applied to each baseline and augmented benchmark. The best-performing parameters for the backbone model were applied in all comparisons for the same model on the same dataset. Specifically, the final chosen learning rates and the times of learning rate decay settings for each dataset are listed in Table **??**. We release the detailed training recordings in the list of experiments and results online with the code base at https://github.com/sagizty/CellMix.

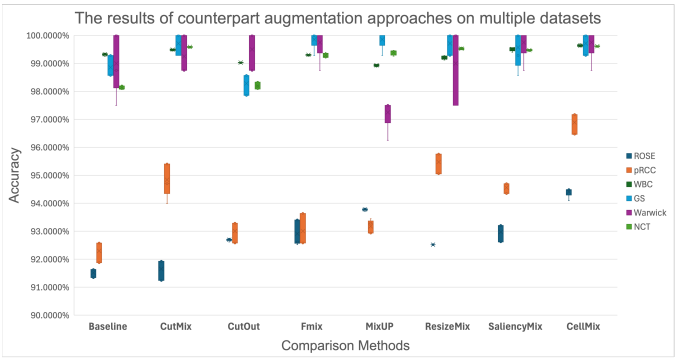


Fig. S1. Accuracy (%) comparison with SOTA mixing-based data augmentation methods on datasets including ROSE, pRCC, WBC, GS, Warwick, MARS, and NCT. All counterpart methods are evaluated with ViT-base.

## S2. ONLINE DETAILS OF EXPERIMENTS

As we aim to explore the robustness of CellMix, additional experiments contributed by the research community are welcomed. We have opened the official GitHub repository for updating results on both internal and public datasets. The results are consistently updated at https://github.com/sagizty/CellMix.

Moreover, we have released all training records and detailed information about the experiments presented in this paper on the same GitHub page. Due to limited writing space, we update detailed result tables online, including error bars for various methods and datasets such as Fig.S1.

## S3. ADDITIONAL EXPERIMENTS

### A. Random Labels

To explore the robustness of CellMix, we additionaly explore the function of labels on augmented images. The soft labels are generated based on the proportion of the mixed patches. One might be wondering if the data augmentation method still works with the wrong labels. To tackle the aforementioned issue, we select different proportions of images and generate random labels for them. Then the dataset is augmented with these random labels and ground truth labels from unselected images.

Table SI illustrates the drop in performance when labels of augmented images are wrong. It verifies that soft labels, as guidance during model training, are indeed learned by the

TABLE SI
ACCURACY (%) FOR THE DATASET WITH DIFFERENT PROPORTIONS OF RANDOM LABELS. FROM 0 TO 0.8, THEY DENOTE THE RATIO OF WRONG LABELS. FOR EXAMPLE, 0.2 DENOTES 20% WRONG LABELS WITH 80% GROUND TRUTH.

| Random Ratio | ROSE | pRCC | WBC |
|---|---|---|---|
| 0 | **94.49** | **97.17** | **99.26** |
| 0.2 | 54.93 | 66.90 | 39.16 |
| 0.4 | 72.68 | 52.67 | 78.84 |
| 0.6 | 72.39 | 70.46 | 58.93 |
| 0.8 | 53.45 | 50.53 | 20.60 |

model. In addition, it shows that CellMix has no over-fitting issue, otherwise, the results of different random ratios would be similar to each other.

### *B. Application on Weakly Supervised Semantic Segmentation*

Though the CellMix is proposed for pathological image classification backbones, we further explore CellMix in the segmentation scenario to highlight and segment regions of diagnostic relevance [**?**]. While most segmentation models rely on pixel-level annotations in training, CellMix encompasses the Weakly Supervised Semantic Segmentation (WSSS) to achieve training with only sample-level categorical labels [**?**].

Specifically, we attach CellMix before the SEAM [**?**] training to enhance the pseudo segmentation mask generator. The original and CellMix permuted samples will be used in this alternative training paradigm. We denote this application version as SegMix and we use the ROSE, WBC, and MARS datasets for preliminary evaluation. Specifically, we implement and compare other WSSS framework baselines (CAM [**?**], SEAM [**?**], and CPN [**?**]) for a fair comparison in WSSS mask generation. To further validate the effectiveness of the loss-driven modules, we fix the patch size to 32 and the shuffle ratio to 0.3, which is denoted as SegMix w/o LD.

In the WSSS training, multi-label soft margin loss is adopted with an Adam optimizer for 8 epochs, with a batch size of 4, an initial learning rate of 0.0001 for the ROSE and WBC datasets, and 0.00009 for the MARS dataset. Following the classification task, all datasets are randomly separated into training, validation, and test sets following a ratio of 7:1:2. For the evaluation metrics, we apply the widely used Dice score (DSC) and IoU to measure the mask generation.

The result is shown in Table SII. Specifically, Subtype1 includes the negative samples in ROSE, eosinophils in WBC, and tubular adenocarcinoma in MARS, while Subtype2 denotes positive samples in ROSE, monocytes in WBC, and mucinous adenocarcinoma in MARS. The selected subtypes are annotated by senior pathologists as they are the most related to clinical diagnosis. The Average denotes the mean value of the two classes. Table SII shows that SegMix achieves the best performance, with DSC 60.7% and IoU 50.2% on ROSE, DSC 41.5% and IoU 29.4% on WBC, and DSC 40.6% and IoU 31.3% on MARS.

In high magnification tasks like WBC, it can be observed that SegMix performs well on the majority of subtypes, particularly on eosinophils (Subtype1), achieving a remarkable

TABLE SII
DICE SCORE (%) AND IOU (%) OF METHODS FOR WEAKLY SUPERVISED SEMANTIC SEGMENTATION. SEGMIX W/O LD DENOTES THE PROPOSED METHOD WITHOUT THE LOSS-DRIVEN SCHEDULERS.

| Dataset | Method | Dice Score (DSC) [%] | | | IoU [%] | | |
|---|---|---|---|---|---|---|---|
| | | Subtype1 | Subtype2 | Average | Subtype1 | Subtype2 | Average |
| ROSE | CAM | 39.1 | 52.5 | 45.8 | 32.5 | 38.5 | 35.6 |
| | SEAM | 35.4 | 69.6 | 52.5 | 28.9 | **58.3** | 43.7 |
| | CPN | 42.7 | 62.9 | 50.1 | 33.2 | 51.4 | 42.3 |
| | SegMix w/o LD | 47.6 | 63.3 | 55.4 | 39.9 | 50.8 | 45.4 |
| | SegMix | **51.4** | **70.0** | **60.7** | **42.4** | 58.1 | **50.2** |
| WBC | CAM | 12.1 | 30.7 | 28.3 | 7.0 | 21.3 | 21.0 |
| | SEAM | 39.5 | 39.0 | 29.8 | 28.9 | 27.2 | 20.4 |
| | CPN | 45.7 | 36.3 | 31.1 | 33.2 | 24.1 | 22.1 |
| | SegMix w/o LD | 41.0 | 35.2 | 34.6 | 31.6 | 23.7 | 26.6 |
| | SegMix | **60.9** | **49.2** | **41.5** | **41.4** | **32.3** | **29.4** |
| MARS | CAM | 56.9 | 57.7 | 38.2 | 44.0 | 42.9 | 29.0 |
| | SEAM | 64.4 | 56.8 | 40.4 | 51.0 | 42.1 | 31.0 |
| | CPN | 60.2 | 54.1 | 39.8 | 47.5 | 41.8 | 30.8 |
| | SegMix w/o LD | 62.8 | **58.1** | 40.3 | 49.6 | **43.7** | 31.2 |
| | SegMix | **64.5** | 57.2 | **40.6** | **51.1** | 42.6 | **31.3** |

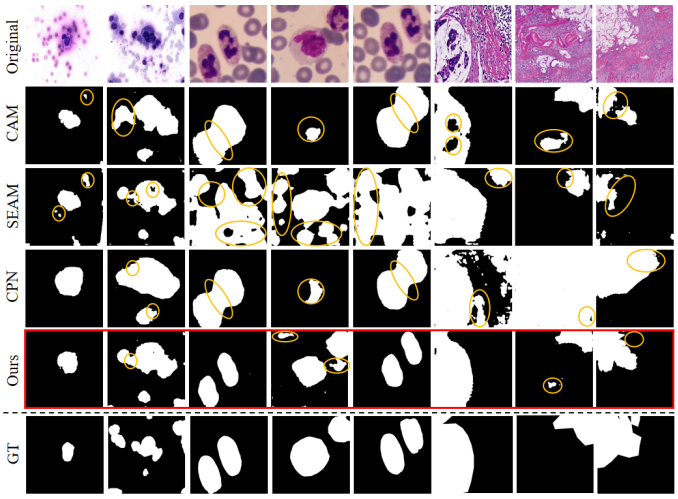


Fig. S2. Visualization of crafted examples from different methods. The first row is the original images from ROSE, WBC and MARS. GT represents the ground truth segmentation masks. Other rows are results of weakly supervised methods. The areas with inaccurate predictions are marked with yellow circles.

improvement of 34.4% for IoU and 48.8% for DSC compared to the baseline. In low magnification tasks, such as MARS, we observe the performance enhancements are less significant. This may be related to intra-tumor heterogeneity, where at the lower magnification, the features are distributed differently [**?**]. In these scenarios, the global distribution towards the sample-level label is more obstructive, and the enhancement of pattern modeling may be less sensitive on shuffling.

In addition to numerical comparison, segmentation visualizations are explored in Fig. S2. The general performance of WSSS is limited as the training is based on sample-level label instead of pixel-level. Still, SegMix outperforms the other WSSS methods generally, where baselines tend to highlight and segment regions that contain some irrelevant pixels. The findings are consistent with Grad-CAM in Fig. **??** and Fig. **??**, which suggest that they may be affected by the perturbations in pathology images, as mentioned in previous sections. In the segmentation scenario, shuffling may change the semantics to the pixel-level. Therefore, we explored the segmentation based on sample-level label weak supervision. Here, the shuffling design enhances the model to work on the regrouped instances relationships (distribution of the features) towards sample-

level semantic. By shuffling and regrouping instances, SegMix learns instance relationships with augmented distributions, therefore enhancing the segmentation performance.

Furthermore, we compare SegMix with a simple ablation (SegMix w/o LD), where the loss-driven modules (patch size and fix-position schedulers) are removed. Even without the loss-driven modules, the method outperforms the baselines, demonstrating that the shuffle module contributes to segmentation by better modeling local instances and global information through patch regrouping. However, with the loss-driven modules, the performance significantly increases, with up to 6.9% for DSC and 4.8% for IoU. This demonstrates the effectiveness of loss-driven modules, which allows the model to adaptively adjust the shuffle strategy based on the previous steps. Meanwhile, by continuously adjusting the patch size and shuffle ratio during the shuffle process, the model can learn multi-scale pathological features from coarse to fine granularity, which improves model performance.

## S4. Pseudo code for the CellMix Algorithm

We illustrate a simplified pseudo code for fix-position ratio scheduler as Algorithm 1.

**Algorithm 1:** Simplified Fix-Position Ratio Scheduler (Loss-driven)

**Input:** Baseline ratio plan: $[f_1, f_2, \ldots, f_n]$
Loss threshold $T$, threshold controlling ratio $T_{\text{control}}$
Strategy $\in$ {*loss-hold*, *loss-back*}
Total epochs $N$
**Data:** Current epoch $e$, current loss $l$, index $i \in \{1, \ldots, n\}$
**Result:** Updated fix-position ratio $f_i$

**Initialize:** $i \leftarrow 1$, $f_i \leftarrow f_1$ // Start at easiest ratio
**for** $e \leftarrow 1$ ***to*** $N$ **do**
  // 1. Train and compute the loss at current ratio $f_i$
  $l \leftarrow \text{ComputeLoss}(\text{model}, f_i)$
  // 2. Compare loss $l$ with threshold $T$
  **if** $l \leq T$ **then**
    // Loss is small enough; move to a more difficult ratio
    $i \leftarrow i + 1$
    **if** $i \leq n$ **then**
      // Reduce ratio by 10% from the baseline $f_i$
      $f_i \leftarrow f_i \times 0.9$
      // Update threshold to expect a further 5% decrease
      $T \leftarrow T \times T_{\text{control}}$
    **end**
  **end**
  **else**
    // Loss is high; apply chosen strategy
    **if** *strategy* == *loss-hold* **then**
      // Do not change the ratio
      $f_i \leftarrow f_i$
    **end**
    **else if** *strategy* == *loss-back* **then**
      // Reduce complexity (increase ratio) by 10%
      $f_i \leftarrow f_i \times 1.1$
    **end**
  **end**
  // 3. Clip ratio to stay in $(0, 1)$
  $f_i \leftarrow \max\big(\min(f_i, 1), 0\big)$
  // (Repeat training next epoch with updated ratio)
**end**
**return** $f_i$