Loss Function Exploration in Medical Image Segmentation

1st Haruto Suzuki
Computer Science Department
Rensselaer Polytechnic Institute
Troy, United States
suzukh@rpi.edu

2nd Ronin Silvestre
Computer Science Department
Rensselaer Polytechnic Institute
Troy, United States
silver3@rpi.edu

***Abstract*—We evaluate the effects of loss function selection on a U-Net architecture, in the context of polyp detection for medical image segmentation tasks. We explore several combinations of common loss functions: Binary Cross Entropy, Dice, Adaptive t-vMF Dice, Focal, Tversky, and Boundary. Experiments are conducted on Kvasir-SEG, ClinicDB, and ColonDB, which are publicly available gastrointestinal polyp segmentation datasets. With each combination of loss functions, we train the model under identical conditions to ensure fair comparison, allowing us to extrapolate the differences between each loss function combination. We assess the performance of our model most notably with metrics the Dice metric to evaluate the similarity between our model predicted segmentation masks and the ground truth, as well as Recall to analyze the percentage of true positive pixels that are correctly identified by our model. Statistical testing is then applied to assess whether there are significant performance differences observed across each loss function combination. Our results demonstrate how the loss function choice can produce noticeable variation in medical image segmentation performance. The findings provide practical guidance for selecting loss functions in medical image segmentation tasks on a U-net architecture.**

***Index Terms*—U-Net, Attention U-Net, Medical Image Segmentation, Polyp Segmentation, Loss Functions, Hybrid Loss Functions, Convolutional Neural Networks (CNNs), Deep Learning (DL), Gastrointestinal Imaging.**

I. Introduction

Segmenting medical images is an important step in healthcare because it helps doctors clearly outline organs and detect problematic areas in patients. Medical Image Segmentation has found its way into areas of research such as lesion, cell, and brain tumor segmentation. Our project specifically focuses on gastrointestinal segmentation, which is the task of finding and segmenting colorectal polyps. If gone unnoticed for too long, these polyps may potentially develop into cancer. Detecting these polyps by hand is a tedious task, often demanding a significant amount of time and effort. Additionally, results may be inconsistent depending on the specialist, due to human error. Thus, the automation of a faster, more consistent process motivates the use of Artificial Intelligence, particularly Deep Learning, in this medical space.

The U-Net architecture, which is a Convolution Neural Network (CNN) that can be characterized by its "U" like structure due to downsampling and upsampling of the input, is one of the most common models for medical image segmentation.

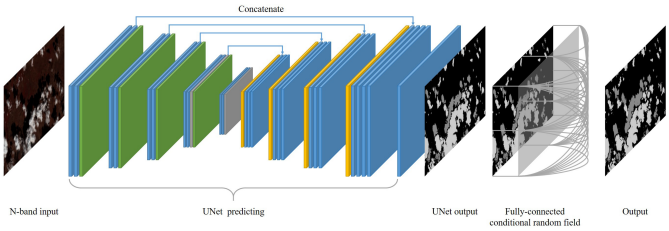


Fig. 1: U-Net Architecture

It uses an encoder-decoder structure featuring skip connections to preserve spatial details even while downsampling, thus contributing to its ability to capture both the overall picture and small details in the image. A diagram of the U-Net Architecture can be found in 1. Much of the existing research in Medical Image Segmentation focuses on improving performance by modifying the model's design. Thus, there is not as much attention on the loss functions used in the training phase of these models. The purpose of the loss function is to control how the model learns from mistakes, and thus, choice of loss function directly affects how quickly the model can learn the patterns of the data, and in our case, how well the model balances between catching all polyps and avoiding false alarms.

In this project, we explore how different combinations of loss functions affect the performance of a U-Net trained for polyp segmentation. In particular, we test combinations of Binary Cross Entropy (BCE), Dice, Adaptive t-vMF Dice, Focal, Tversky, and Boundary losses. We train and test these setups on the public gastrointestinal polyp segmentation datasets Kvasir-SEG, ClinicDB, and ColonDB, and we compare their performance with the Accuracy, Dice, Precision, Recall, F1, Boundary F1, Boundary IOU metrics. We also run thorough statistical tests to see if the differences we find are significant. Our goal is to figure out which loss function combinations work best for image segmentation with U-Net, with the hope that future research can make better choices without solely relying on the choice of model architecture.

II. Related Work

Given U-Net's ability to perform effectively even with limited training data and in situations with significant class imbalance, it has become one of the most popular choices for medical image segmentation tasks. Its encoder-decoder structure with skip connections preserves spatial information while still learning high-level features, making it well-suited for identifying small or irregularly shaped regions in medical images. In segmentation tasks, the choice of loss function can have a major impact on how well the model performs, especially when the target objects occupy only a small part of the image. The loss function controls how the model updates its parameters during training, which directly affects convergence speed, error weighting, and the balance between precision and recall.

Previous studies have proposed the use of focal-type loss functions, such as Focal Tversky Loss [1, 2] and Unified Focal Loss [3], as ways to improve model performance when classes are highly imbalanced. These losses put more focus on difficult-to-classify regions and help address the common issue of models ignoring smaller structures in favor of larger, easier-to-segment areas.

Other work has explored combining multiple loss functions to take advantage of their different strengths [4, 5, 6]. For example, hybrid approaches like BCE with Dice Loss can benefit from the pixel-level precision of BCE while also optimizing for region overlap through Dice Loss [2]. Studies comparing such combinations have reported improved Dice scores, IoU values, and better handling of false positives or false negatives.

Furthermore, some research has focused on more general analyses of loss function behavior to determine when one loss might be better than another. These works aim to provide guidelines for selecting a loss function in medical image segmentation based on dataset characteristics, class distribution, and the specific goals of the segmentation task [7, 8, 9, 10]. This type of analysis is directly relevant to our project, which aims to test and compare several loss functions in a controlled U-Net-based polyp segmentation setup.

III. Methodology

In recent years, many loss functions have been proposed for segmentation. This project investigates the effect of several mainstream loss functions on segmentation performance when using a fixed U-Net with Attention architecture for polyp detection. The loss functions evaluated include Binary Cross-Entropy (BCE), Dice Loss, Adaptive t-vMF Dice Loss, Focal Loss, and Tversky Loss, along with selected hybrid combinations of these functions. In our implementation, some loss functions were also combined with weighting schemes to create hybrid loss functions, aiming to leverage both pixel-level accuracy and region-overlap optimization.

We first implemented a standard U-Net architecture with an encoder–decoder design and skip connections, as it is a proven baseline for medical image segmentation tasks and can be trained efficiently within our compute budget. The encoder progressively downsamples the input to extract high-level features, while the decoder upsamples and combines these with the encoded features via skip connections to reconstruct the segmentation map. After successfully iterating on our foundational U-Net model, we implemented the modifications described in the U-Net with Attention paper [11] as visually described in 2. All results presented in this study were obtained using this attention mechanism to enhance feature selection and improve overall performance. Attention Gate helps the mainly the decoder side to receive the filtered out the key features from the encoder. As shown in 3, more filters are applied such that unlike raw encoded images with noise and unnecessary stuff, it lets decoder focus on important pieces.

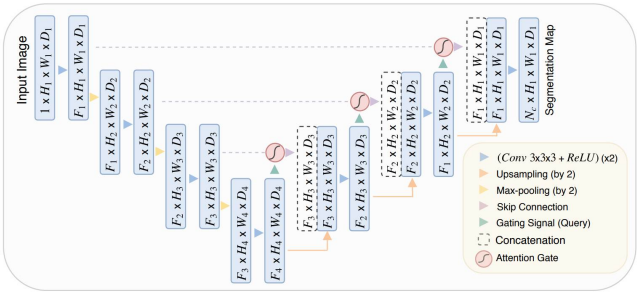


Fig. 2: U-Net with Attention Gate

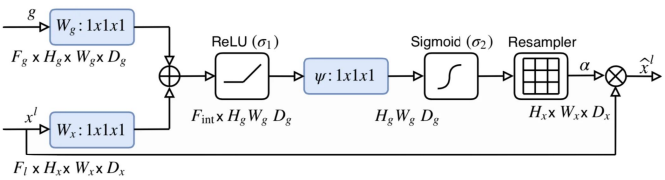


Fig. 3: Detailed Attention Gate

The main dataset used in this study is the publicly available Kvasir-SEG collection, which contains images of gastrointestinal polyps and their corresponding segmentation masks. These masks were manually annotated by a medical doctor and later verified by an experienced gastroenterologist [12]. The dataset contains 1,000 polyp images, with dimensions ranging from 332×487 to 1920×1072 pixels. We resized all Kvasir-SEG images to 320×320 to unify the image dimensions and reduce memory usage. While this resizing inevitably causes some loss of resolution, the effect is consistent across all experiments and therefore does not bias comparisons between loss functions.

To evaluate performance in different environments, we also used the CVC-ClinicDB and CVC-ColonDB datasets. CVC-ClinicDB images are originally 384×288 pixels, while CVC-ColonDB images are 574×500 pixels. Due to GPU memory constraints and our chosen hyperparameters (notably batch size), we resized CVC-ColonDB images to 320×320.

All other hyperparameters, such as number of epochs, learning rate, batch size, random seed, and optimizer, were kept constant. By fixing the architecture and varying only the

loss function, we aimed to isolate and better understand the influence of the loss function on performance.

To address the limited amount of data in medical imaging, we applied common data augmentation techniques. We reserved 20% of the training data for validation in all experiments.

Performance was assessed using standard metrics: Accuracy, Dice Coefficient, Precision, Recall, F1 Score, Intersection over Union (IoU), Boundary F1 Score, and Boundary IoU. Multiple configurations were tested for each loss function, and results were compared both quantitatively and statistically. Statistical tests, including the Wilcoxon signed-rank test and effect size calculations, were used to determine whether observed performance differences were significant.

All experiments were implemented in Python using the PyTorch framework with CUDA acceleration on a Kaggle notebook equipped with an NVIDIA P100 GPU. Supporting libraries included NumPy and Pandas for data handling, and Matplotlib/Seaborn for visualization. Code and configuration files were tracked via GitHub for transparency and reproducibility.

IV. Experimental Setup

A. Image Preprocessing

Before training, all input images were normalized to maintain consistent intensity ranges. Binary segmentation masks were encoded as 0 (background) and 1 (foreground). Images were resized when necessary to meet the input size requirements of the U-Net architecture and to fit within the available GPU memory limits. This resizing was applied consistently across all datasets used in the study.

B. Data Augmentation

To improve generalization and reduce overfitting, several data augmentation techniques were applied to the training set. These transformations were chosen to simulate realistic variations in medical imaging conditions without altering the semantic meaning of the images. Specifically, we applied horizontal and vertical flips, random rotations, and adjustments to brightness and contrast. These augmentations were applied probabilistically during training so that the model would encounter a diverse set of input variations across epochs.

C. Hyperparameters

All experiments were conducted using a consistent set of hyperparameters to ensure comparability between results. The random seed was fixed at 42 for reproducibility. Training was carried out for 30 epochs with a batch size of 10, and the learning rate was set to 3×10^{-4}. For experiments involving combined loss functions, the weights assigned to each component were fixed across runs to isolate the effect of the loss type itself. No learning rate scheduling was applied during training.

D. Execution Setup

We evaluated a total of 11 different loss function configurations, including both single losses and hybrid combinations. The combinations of loss functions can be found in I. The dataset was split into 80% for training and 20% for validation, ensuring that all metrics were calculated exclusively on the validation set. No early stopping criteria were employed, allowing each configuration to run for the full number of epochs. Performance metrics were computed for each image at every epoch to capture fine-grained changes in performance over time.

TABLE I: Loss Function Combinations Used in Experiments

Loss 1	**Loss 2**	**Loss 3**
CE		
Dice		
CE	Dice	
CE	Adaptive Dice	
CE	Focal	
CE	Tversky	
CE	Adaptive Dice	Focal
CE	Adaptive Dice	Tversky
CE	Dice	Focal
CE	Dice	Tversky
CE	Adaptive Dice	Boundary

E. Statistical Analysis

To rigorously evaluate performance differences among the various loss function combinations, we conducted pairwise statistical tests on every metrics. Only epochs greater than or equal to 25 were considered in the analysis. For each model, we first computed the per-image metric values and then calculated the average for each metric per epoch. The top-performing model for each metric was identified based on its best epoch, and this model was compared against all other models to assess performance differences in the same epoch for fairness.

For the statistical comparisons, we used either the paired t-test or the Wilcoxon signed-rank test, depending on whether normality of the metric differences could be assumed. The Wilcoxon signed-rank test is a non-parametric test suitable for paired data when normality cannot be guaranteed, while the paired t-test is applied when the differences are approximately normally distributed. For each comparison, effect sizes and 95% confidence intervals were computed to quantify the magnitude and reliability of the observed differences. Statistical significance was determined using a threshold of $p < 0.05$, with significant results indicating that the top model performs meaningfully better than the comparison model.

Finally, results were visualized using boxplots and paired line plots to clearly show per-image metric distributions and differences between models. This approach ensures that both statistical significance and practical relevance of performance differences are transparently presented, which is especially important in medical imaging tasks where minimizing false negatives is critical.

V. Results

All detailed tables and outputs for our experiments are provided in Appendix C. In this section, we summarize and highlight key results for two primary metrics: **Dice** and **Recall**. Dice coefficient is a standard metric for image segmentation that measures the overlap between predicted and ground truth masks, while Recall is particularly important in medical applications because it captures false negatives, which we aim to minimize to avoid missing critical findings.

Tables V and VIII show pairwise statistical tests on the Kvasir-SEG dataset for Dice and Recall metrics, respectively. Tables XII and XV report results on CVC-ClinicDB, and Tables XIX and XXII show results on CVC-ColonDB. Each table lists the top-performing model compared with other models, the statistical test used, p-values, significance, effect sizes, and confidence intervals.

VI. Analysis and Discussion

In this section, we analyze the results presented in the previous section and discuss key insights from the pairwise statistical tests across the three datasets: Kvasir-SEG, CVC-ClinicDB, and CVC-ColonDB. We focus on the Dice and Recall metrics, which are crucial for evaluating image segmentation performance, particularly in medical imaging tasks where false negatives must be minimized.

The visualization results in Figure 4 show generated segmentation images alongside their corresponding statistical distributions. The generated images represent the model's segmentation output overlaid on the original endoscopic images, displaying the predicted polyp or lesion boundaries. For each metric's box plot, the leftmost bar represents the top-performing model for that specific metric on that dataset, providing a clear visual comparison of performance distributions across different loss functions.

For the Kvasir-SEG dataset, the top-performing model for Dice was `ce adaptive`, while for Recall it was `ce dice focal`. The `ce adaptive` model significantly outperformed standard CE-based models and several variants, with high effect sizes and confidence intervals consistently above zero. In terms of Recall, `ce dice focal` consistently achieved the highest scores, with very strong effect sizes, indicating a reliable improvement in capturing true positives. These results suggest that adaptive weighting strategies improve segmentation overlap, while Dice-focused focal losses enhance sensitivity to foreground regions.

For the CVC-ClinicDB dataset, `ce adaptive dice focal` and `ce adaptive dice tversky` achieved the best Dice and Recall performance, respectively. The effect sizes for Dice range from moderate to strong, indicating notable improvements over other models. Recall improvements were even more pronounced, with several comparisons showing very high effect sizes, highlighting the effectiveness of the Tversky-based adaptive loss in reducing false negatives. These results underscore the importance of using combined adaptive and Dice/Tversky losses for highly sensitive medical segmentation tasks.

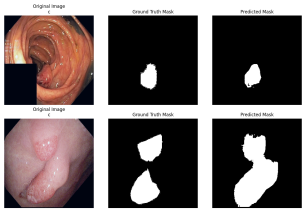
For the CVC-ColonDB dataset, similar trends were observed, with `ce adaptive dice tversky` and `ce tversky` achieving top Dice and Recall scores, respectively. Effect sizes for Dice were moderate to strong, confirming that adaptive Tversky variants significantly improve segmentation quality. Recall effect sizes were very high, often approaching one, indicating near-complete reduction of false negatives relative to baseline models. This trend reinforces the value of Tversky-based losses in highly imbalanced segmentation problems, where foreground regions are small and missing them is critical.

A notable observation from the CVC-ColonDB Dice box plot reveals that the calculated Dice metrics are significantly spread apart with high standard deviation and interquartile range (IQR). This wide distribution likely explains the background-only generation phenomenon observed in some model outputs, where models default to predicting no foreground pixels when uncertainty is high. The CVC-ColonDB dataset presents unique challenges as it contains substantially smaller foreground regions compared to Kvasir-SEG and CVC-ClinicDB datasets, making accurate segmentation more difficult due to the extreme class imbalance. This smaller target area requires models to be more precise in their predictions, leading to increased variability in performance metrics.

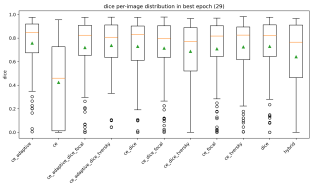
The analysis of top-performing Recall models reveals an interesting behavioral pattern where these models tend to classify some non-foreground areas as foreground regions. This tendency stems from the models' strategy to minimize false negative errors by predicting slightly larger segmentation areas than the ground truth. While this approach may reduce the Recall metric's numerical value in some cases, it represents a clinically favorable trade-off, as covering a broader area ensures that potential lesions are not missed, even at the cost of including some healthy tissue in the segmentation boundary.

Across all datasets, a few consistent patterns emerge. Adaptive losses generally improve Dice scores, as models with adaptive weighting outperform static cross-entropy or focal losses. Models incorporating Dice or Tversky-based losses are more sensitive to foreground pixels, reducing false negatives and improving Recall. Comparisons with high effect sizes almost always correspond to statistically significant p-values, indicating reliable improvements rather than random fluctuations. These findings suggest that carefully designed adaptive loss functions that combine CE, Dice, and Tversky components can balance segmentation accuracy and sensitivity, which is crucial in medical applications where missing lesions or polyps could have serious consequences.

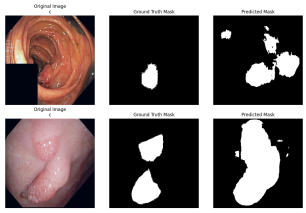
The strong performance of Dice- and Tversky-based losses in Recall highlights their practical importance for medical imaging, as minimizing false negatives directly translates to fewer missed detections in clinical practice. Therefore, models trained with these losses are preferable when the cost of missing a lesion is higher than that of false positives, which can be manually reviewed by clinicians.



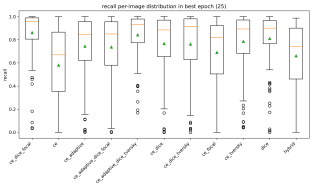
(a) Kvasir - Dice Scores



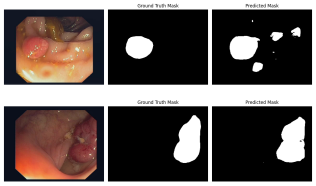
(b) Kvasir - Dice Box Plot



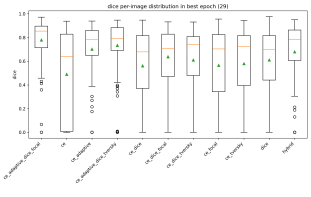
(c) Kvasir - Recall Scores



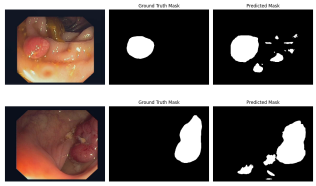
(d) Kvasir - Recall Box Plot



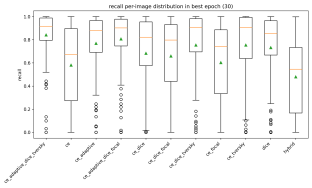
(e) CVC-ClinicDB - Dice Scores



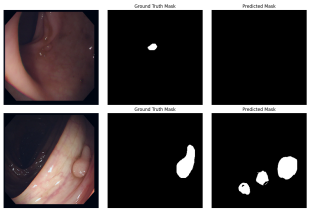
(f) CVC-ClinicDB - Dice Box Plot



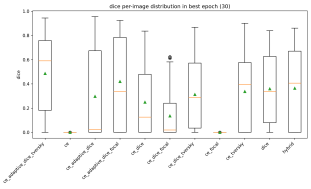
(g) CVC-ClinicDB - Recall Scores



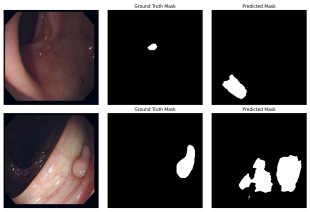
(h) CVC-ClinicDB - Recall Box Plot



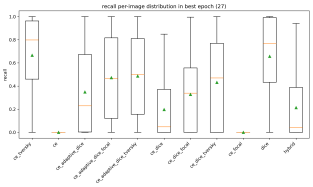
(i) CVC-ColonDB - Dice Scores



(j) CVC-ColonDB - Dice Box Plot



(k) CVC-ColonDB - Recall Scores



(l) CVC-ColonDB - Recall Box Plot

Fig. 4: Performance comparison across three datasets showing Dice and Recall metrics. Each row represents a dataset (Kvasir, CVC-ClinicDB, CVC-ColonDB) with columns showing: (1) Dice scores, (2) Dice box plots, (3) Recall scores, (4) Recall box plots.

VII. Limitations

This study faced several constraints that influenced the scope and depth of our experiments. First, our available computing resources limited the size and complexity of the datasets we could process within a reasonable timeframe. Our original plan was to conduct our experiments on the BraTS datasets for brain tumor segmentation. However, these images are 3D, making them much more complex than 2D images. Consequently, we restricted our evaluation to the 2D image datasets Kvasir-SEG, ClinicDB, and ColonDB for gastrointestinal polyp segmentation.

Furthermore, to address GPU memory limitations and maintain feasible training times, the ColonDB input images were downsized. While this standardization reduced computational load, it may have resulted in a loss of fine-grained visual detail, potentially affecting segmentation accuracy. Since polyps in gastrointestinal scans tend to be relatively small, compared to the entire scan, the loss of fine-grained visual detail could negatively impact our model's performance.

Lastly, the constraints of the shortened academic semester noticeably impacted the scope of our project. Our evaluation was confined to gastrointestinal polyp datasets as aforementioned, and therefore our key takeaways from our loss function exploration may not directly generalize to other medical image segmentation domains without additional validation. With more time, we would have aimed to attempt generalization to other domains.

VIII. Future Work

Future research should aim to expand the scope of this work by exploring larger and more complex datasets to improve generalization. This could include 3D datasets, higher-resolution images, and multi-center clinical datasets that better capture the variability found in real-world medical imaging. Beyond dataset size, experimenting with alternative model architectures, such as transformer-based segmentation networks may enhance performance further. Additionally, testing with different hyperparameters and further investigation of other loss functions could also be applied to further refine training setups and yield valuable insights. Finally, addressing current computational and memory constraints, through either optimized implementations or running on more powerful GPUs, would enable the use of larger datasets and more complex models, broadening the applicability of this research.

References

[1] N. Abraham and N. M. Khan, "A Novel Focal Tversky Loss Function with Improved Attention U-Net for Lesion Segmentation," Dept. of Electrical and Computer Engineering, Ryerson University, Toronto, ON, Canada.

[2] M. Montazerolghaem, Y. Sun, G. Sasso, and A. Haworth, "U-Net Architecture for Prostate Segmentation: The Impact of Loss Function on System Performance," School of Physics, The University of Sydney, Sydney, NSW, Australia; Radiation Oncology Department, Auckland City Hospital, Auckland, New Zealand; Faculty of Medical and Health Sciences, University of Auckland, Auckland, New Zealand.

[3] M. Yeung, E. Sala, C.-B. Schönlieb, and L. Rundo, "Unified Focal loss: Generalising Dice and cross entropy-based losses to handle class imbalanced medical image segmentation," University of Cambridge, UK; University of Salerno, Italy.

[4] M. Yeung, E. Sala, C.-B. Schönlieb, and L. Rundo, "Focus U-Net: A Novel Dual Attention-Gated CNN for Polyp Segmentation during Colonoscopy," University of Cambridge, UK.

[5] L. F. Sánchez-Peralta, A. Picón, J. A. Antequera-Barroso, J. F. Ortega-Morán, F. M. Sánchez-Margallo, and J. B. Pagador, "Eigenloss: Combined PCA-Based Loss Function for Polyp Segmentation," Jesús Usón Minimally Invasive Surgery Centre, Spain; TECNALIA, Spain.

[6] W. Zhang, Y. Chen, Z. Long, H. Chen, Y. Zhang, Z. Zhou, W. Chen, and X. Le, "Focal difficult-to-predict pixels dice loss for mitigating data imbalance in medical image segmentation," various institutions in China.

[7] S. Jadon, "A Survey of Loss Functions for Semantic Segmentation," IEEE Member.

[8] R. Bourday, I. Aattouchi, and M. Ait Kerroum, "A Comparative Study of Deep Learning Loss Functions: A Polyp Segmentation Case Study," in *Studies in Computational Intelligence*, vol. 1145, Feb. 2024.

[9] G. Batchkala and S. Ali, "Real-Time Polyp Segmentation Using U-Net with IoU Loss," Dept. of Computer Science, University of Oxford, Oxford, UK; Institute of Biomedical Engineering, University of Oxford, Oxford, UK.

[10] S. Kato and K. Hotta, "Adaptive t-vMF Dice Loss for Multi-class Medical Image Segmentation," Dept. of Electrical and Electronic Engineering, Meijo University, Japan.

[11] O. Oktay et al., "Attention U-Net: Learning Where to Look for the Pancreas," arXiv preprint arXiv:1804.03999, 2018.

[12] D. Jha, P. H. Smedsrud, M. A. Riegler, P. Halvorsen, T. de Lange, D. Johansen, and H. D. Johansen, "Kvasir-SEG: A Segmented Polyp Dataset," SimulaMet, Norway; UIT The Arctic University of Norway; University of Oslo; Oslo University Hospital; Oslo Metropolitan University; Kristiania University College.

IX. Appendices

A. Appendix A: Code

To encourage reproducibility, the complete source code for our experiments, including model implementation, training scripts, and evaluation notebooks, is available at: https://github.com/SHaruto0/Medical-Image-Segmentation-Loss-Function-Exploration.

B. Appendix B: Team Contributions

Haruto (Team Lead)

- Implemented U-Net and 3d U-Net (before switch from brain tumor to polyp segmentation)
- Implemented the final Attention U-Net model for polyp segmentation
- Responsible for compiling the results (tables, figures)
- Ran experiments on Kaggle notebook equipped with an NVIDIA P100 GPU. on personal account
- Wrote sections of the report
- Prepared class presentation

Ronin

- Implemented U-Net and 3d U-Net (before switch from brain tumor to polyp segmentation)
- Responsible for the literature review
- Ran experiments on Kaggle notebook equipped with an NVIDIA P100 GPU. on personal account
- Wrote sections of the report
- Prepared class presentation

C. Appendix C: Supplementary Results

This appendix contains supplementary figures referenced in the Results Section. The most relevant results are shown in the main text; additional examples and visualizations are provided here for completeness.

The following tables summarize additional experimental results not included in the main body. For interpretation, refer to the captions provided with each table.

Top Model	Other Model	Test	p-value	Significant	Effect Size	CI Low	CI High
ce_adaptive_dice_focal	ce	Wilcoxon	1.747e-13	✓	0.647	0.0248	0.0406
ce_adaptive_dice_focal	ce_adaptive	Wilcoxon	1.956e-02	✓	0.167	0.0019	0.0136
ce_adaptive_dice_focal	ce_adaptive_dice_tversky	Wilcoxon	1.823e-02	✓	0.227	-0.0001	0.0130
ce_adaptive_dice_focal	ce_dice	Wilcoxon	2.174e-03	✓	0.311	-0.0015	0.0102
ce_adaptive_dice_focal	ce_dice_focal	Wilcoxon	2.314e-01		0.076	-0.0026	0.0088
ce_adaptive_dice_focal	ce_dice_tversky	Wilcoxon	5.103e-10	✓	0.483	0.0136	0.0250
ce_adaptive_dice_focal	ce_focal	Wilcoxon	3.471e-03	✓	0.261	0.0013	0.0154
ce_adaptive_dice_focal	ce_tversky	Wilcoxon	9.706e-03	✓	0.200	-0.0007	0.0138
ce_adaptive_dice_focal	dice	Wilcoxon	8.461e-02		0.133	-0.0014	0.0126
ce_adaptive_dice_focal	hybrid	Wilcoxon	1.855e-02	✓	0.169	0.0012	0.0157

TABLE II: Pairwise statistical tests on Kvasir-SEG for metric accuracy - top model (ce_adaptive_dice_focal) in its best epoch (28).

Top Model	Other Model	Test	p-value	Significant	Effect Size	CI Low	CI High
ce_adaptive	ce	paired t-test	1.142e-23	✓	1.151	0.0561	0.0770
ce_adaptive	ce_adaptive_dice_focal	paired t-test	4.440e-01		0.070	-0.0060	0.0135
ce_adaptive	ce_adaptive_dice_tversky	Wilcoxon	6.629e-07	✓	0.400	0.0113	0.0296
ce_adaptive	ce_dice	Wilcoxon	5.279e-02		0.117	0.0001	0.0171
ce_adaptive	ce_dice_focal	Wilcoxon	2.406e-02	✓	0.267	0.0020	0.0185
ce_adaptive	ce_dice_tversky	Wilcoxon	3.119e-09	✓	0.517	0.0203	0.0378
ce_adaptive	ce_focal	paired t-test	1.911e-06	✓	0.457	0.0131	0.0303
ce_adaptive	ce_tversky	Wilcoxon	5.053e-07	✓	0.400	0.0132	0.0308
ce_adaptive	dice	paired t-test	1.739e-04	✓	0.354	0.0070	0.0215
ce_adaptive	hybrid	paired t-test	1.619e-04	✓	0.356	0.0102	0.0312

TABLE III: Pairwise statistical tests on Kvasir-SEG for metric bf1 - top model (ce_adaptive) in its best epoch (29).

Top Model	Other Model	Test	p-value	Significant	Effect Size	CI Low	CI High
ce_adaptive	ce	paired t-test	4.423e-23	✓	1.128	0.0304	0.0419
ce_adaptive	ce_adaptive_dice_focal	paired t-test	5.757e-01		0.051	-0.0040	0.0072
ce_adaptive	ce_adaptive_dice_tversky	Wilcoxon	8.675e-07	✓	0.400	0.0062	0.0165
ce_adaptive	ce_dice	Wilcoxon	5.882e-02		0.117	-0.0002	0.0095
ce_adaptive	ce_dice_focal	Wilcoxon	2.754e-02	✓	0.267	0.0010	0.0102
ce_adaptive	ce_dice_tversky	Wilcoxon	2.835e-09	✓	0.517	0.0112	0.0211
ce_adaptive	ce_focal	Wilcoxon	3.771e-06	✓	0.417	0.0073	0.0169
ce_adaptive	ce_tversky	Wilcoxon	4.592e-07	✓	0.400	0.0074	0.0173
ce_adaptive	dice	paired t-test	3.117e-04	✓	0.339	0.0036	0.0118
ce_adaptive	hybrid	paired t-test	2.988e-04	✓	0.340	0.0052	0.0171

TABLE IV: Pairwise statistical tests on Kvasir-SEG for metric biou - top model (ce_adaptive) in its best epoch (29).

Top Model	Other Model	Test	p-value	Significant	Effect Size	CI Low	CI High
ce_adaptive	ce	Wilcoxon	2.871e-19	✓	0.850	0.2818	0.3847
ce_adaptive	ce_adaptive_dice_focal	Wilcoxon	1.875e-02	✓	0.100	0.0063	0.0677
ce_adaptive	ce_adaptive_dice_tversky	Wilcoxon	1.298e-01		0.183	-0.0143	0.0530
ce_adaptive	ce_dice	Wilcoxon	5.988e-02		0.167	-0.0052	0.0601
ce_adaptive	ce_dice_focal	Wilcoxon	2.681e-02	✓	0.150	0.0094	0.0765
ce_adaptive	ce_dice_tversky	Wilcoxon	3.109e-05	✓	0.317	0.0339	0.1043
ce_adaptive	ce_focal	Wilcoxon	4.698e-03	✓	0.250	0.0173	0.0781
ce_adaptive	ce_tversky	Wilcoxon	1.482e-02	✓	0.233	0.0054	0.0579
ce_adaptive	dice	Wilcoxon	7.262e-02		0.133	0.0039	0.0503
ce_adaptive	hybrid	Wilcoxon	2.533e-07	✓	0.383	0.0751	0.1557

TABLE V: Pairwise statistical tests on Kvasir-SEG for metric dice - top model (ce_adaptive) in its best epoch (29).

Top Model	Other Model	Test	p-value	Significant	Effect Size	CI Low	CI High
ce_adaptive	ce	Wilcoxon	2.871e-19	✓	0.850	0.2818	0.3847
ce_adaptive	ce_adaptive_dice_focal	Wilcoxon	1.875e-02	✓	0.100	0.0063	0.0677
ce_adaptive	ce_adaptive_dice_tversky	Wilcoxon	1.298e-01		0.183	-0.0143	0.0530
ce_adaptive	ce_dice	Wilcoxon	5.988e-02		0.167	-0.0052	0.0601
ce_adaptive	ce_dice_focal	Wilcoxon	2.681e-02	✓	0.150	0.0094	0.0765
ce_adaptive	ce_dice_tversky	Wilcoxon	3.109e-05	✓	0.317	0.0339	0.1043
ce_adaptive	ce_focal	Wilcoxon	4.698e-03	✓	0.250	0.0173	0.0781
ce_adaptive	ce_tversky	Wilcoxon	1.482e-02	✓	0.233	0.0054	0.0579
ce_adaptive	dice	Wilcoxon	7.262e-02		0.133	0.0039	0.0503
ce_adaptive	hybrid	Wilcoxon	2.533e-07	✓	0.383	0.0751	0.1557

TABLE VI: Pairwise statistical tests on Kvasir-SEG for metric f1 - top model (ce_adaptive) in its best epoch (29).

Top Model	Other Model	Test	p-value	Significant	Effect Size	CI Low	CI High
hybrid	ce	Wilcoxon	6.684e-06	✓	0.228	0.1297	0.2710
hybrid	ce_adaptive	Wilcoxon	2.063e-13	✓	0.800	0.0240	0.1108
hybrid	ce_adaptive_dice_focal	Wilcoxon	1.292e-10	✓	0.573	0.0294	0.0953
hybrid	ce_adaptive_dice_tversky	Wilcoxon	7.494e-16	✓	0.883	0.1040	0.2026
hybrid	ce_dice	Wilcoxon	1.025e-13	✓	0.731	0.0496	0.1353
hybrid	ce_dice_focal	Wilcoxon	9.855e-14	✓	0.748	0.0441	0.1343
hybrid	ce_dice_tversky	Wilcoxon	1.148e-14	✓	0.800	0.1167	0.2252
hybrid	ce_focal	Wilcoxon	6.231e-11	✓	0.607	0.0272	0.1180
hybrid	ce_tversky	Wilcoxon	1.236e-16	✓	0.883	0.0959	0.1858
hybrid	dice	Wilcoxon	2.080e-15	✓	0.815	0.0818	0.1773

TABLE VII: Pairwise statistical tests on Kvasir-SEG for metric precision - top model (hybrid) in its best epoch (29).

Top Model	Other Model	Test	p-value	Significant	Effect Size	CI Low	CI High
ce_dice_focal	ce	Wilcoxon	3.083e-19	✓	0.867	0.2305	0.3316
ce_dice_focal	ce_adaptive	Wilcoxon	6.851e-15	✓	0.783	0.0866	0.1483
ce_dice_focal	ce_adaptive_dice_focal	Wilcoxon	7.676e-15	✓	0.765	0.0886	0.1617
ce_dice_focal	ce_adaptive_dice_tversky	Wilcoxon	3.284e-04	✓	0.368	-0.0068	0.0460
ce_dice_focal	ce_dice	Wilcoxon	5.711e-11	✓	0.633	0.0571	0.1333
ce_dice_focal	ce_dice_tversky	Wilcoxon	3.272e-10	✓	0.492	0.0633	0.1357
ce_dice_focal	ce_focal	Wilcoxon	9.482e-16	✓	0.800	0.1269	0.2161
ce_dice_focal	ce_tversky	Wilcoxon	1.135e-08	✓	0.542	0.0409	0.1114
ce_dice_focal	dice	Wilcoxon	2.485e-12	✓	0.717	0.0290	0.0735
ce_dice_focal	hybrid	Wilcoxon	6.420e-19	✓	0.883	0.1625	0.2395

TABLE VIII: Pairwise statistical tests on Kvasir-SEG for metric recall - top model (ce_dice_focal) in its best epoch (25).

Top Model	Other Model	Test	p-value	Significant	Effect Size	CI Low	CI High
ce_adaptive_dice_focal	ce	Wilcoxon	6.320e-07	✓	0.447	0.0064	0.0168
ce_adaptive_dice_focal	ce_adaptive	Wilcoxon	1.846e-02	✓	0.197	0.0006	0.0090
ce_adaptive_dice_focal	ce_adaptive_dice_tversky	Wilcoxon	2.983e-03	✓	0.236	0.0016	0.0155
ce_adaptive_dice_focal	ce_dice	Wilcoxon	2.057e-08	✓	0.528	0.0061	0.0160
ce_adaptive_dice_focal	ce_dice_focal	Wilcoxon	2.257e-09	✓	0.561	0.0061	0.0153
ce_adaptive_dice_focal	ce_dice_tversky	Wilcoxon	5.600e-13	✓	0.659	0.0139	0.0271
ce_adaptive_dice_focal	ce_focal	Wilcoxon	4.651e-12	✓	0.639	0.0112	0.0199
ce_adaptive_dice_focal	ce_tversky	Wilcoxon	1.783e-11	✓	0.642	0.0130	0.0253
ce_adaptive_dice_focal	dice	Wilcoxon	1.333e-06	✓	0.398	0.0063	0.0190
ce_adaptive_dice_focal	hybrid	Wilcoxon	2.757e-04	✓	0.279	0.0040	0.0133

TABLE IX: Pairwise statistical tests on CVC-ClinicDB for metric accuracy - top model (ce_adaptive_dice_focal) in its best epoch (28).

Top Model	Other Model	Test	p-value	Significant	Effect Size	CI Low	CI High
ce_adaptive_dice_focal	ce	Wilcoxon	1.167e-10	✓	0.530	0.0325	0.0581
ce_adaptive_dice_focal	ce_adaptive	Wilcoxon	2.329e-09	✓	0.554	0.0209	0.0426
ce_adaptive_dice_focal	ce_adaptive_dice_tversky	Wilcoxon	3.411e-03	✓	0.145	0.0097	0.0339
ce_adaptive_dice_focal	ce_dice	Wilcoxon	8.814e-08	✓	0.421	0.0264	0.0509
ce_adaptive_dice_focal	ce_dice_focal	Wilcoxon	2.080e-09	✓	0.466	0.0280	0.0532
ce_adaptive_dice_focal	ce_dice_tversky	Wilcoxon	8.895e-10	✓	0.496	0.0300	0.0543
ce_adaptive_dice_focal	ce_focal	Wilcoxon	2.059e-13	✓	0.694	0.0362	0.0609
ce_adaptive_dice_focal	ce_tversky	Wilcoxon	2.099e-09	✓	0.436	0.0311	0.0565
ce_adaptive_dice_focal	dice	Wilcoxon	1.930e-08	✓	0.426	0.0280	0.0532
ce_adaptive_dice_focal	hybrid	Wilcoxon	8.839e-11	✓	0.661	0.0264	0.0497

TABLE X: Pairwise statistical tests on CVC-ClinicDB for metric bf1 - top model (ce_adaptive_dice_focal) in its best epoch (27).

Top Model	Other Model	Test	p-value	Significant	Effect Size	CI Low	CI High
ce_adaptive_dice_focal	ce	Wilcoxon	1.226e-10	✓	0.530	0.0175	0.0317
ce_adaptive_dice_focal	ce_adaptive	Wilcoxon	2.556e-09	✓	0.554	0.0114	0.0235
ce_adaptive_dice_focal	ce_adaptive_dice_tversky	Wilcoxon	3.301e-03	✓	0.145	0.0054	0.0187
ce_adaptive_dice_focal	ce_dice	Wilcoxon	8.105e-08	✓	0.421	0.0144	0.0279
ce_adaptive_dice_focal	ce_dice_focal	Wilcoxon	2.213e-09	✓	0.466	0.0151	0.0291
ce_adaptive_dice_focal	ce_dice_tversky	Wilcoxon	1.042e-09	✓	0.496	0.0163	0.0297
ce_adaptive_dice_focal	ce_focal	Wilcoxon	2.220e-13	✓	0.694	0.0194	0.0331
ce_adaptive_dice_focal	ce_tversky	Wilcoxon	1.883e-09	✓	0.436	0.0168	0.0309
ce_adaptive_dice_focal	dice	Wilcoxon	1.742e-08	✓	0.426	0.0153	0.0292
ce_adaptive_dice_focal	hybrid	Wilcoxon	1.098e-10	✓	0.661	0.0141	0.0270

TABLE XI: Pairwise statistical tests on CVC-ClinicDB for metric biou - top model (ce_adaptive_dice_focal) in its best epoch (27).

Top Model	Other Model	Test	p-value	Significant	Effect Size	CI Low	CI High
ce_adaptive_dice_focal	ce	Wilcoxon	3.113e-14	✓	0.561	0.2254	0.3527
ce_adaptive_dice_focal	ce_adaptive	Wilcoxon	2.858e-07	✓	0.382	0.0478	0.1067
ce_adaptive_dice_focal	ce_adaptive_dice_tversky	Wilcoxon	1.010e-04	✓	0.317	0.0170	0.0734
ce_adaptive_dice_focal	ce_dice	Wilcoxon	2.327e-13	✓	0.512	0.1649	0.2716
ce_adaptive_dice_focal	ce_dice_focal	Wilcoxon	3.549e-11	✓	0.577	0.0980	0.1855
ce_adaptive_dice_focal	ce_dice_tversky	Wilcoxon	2.417e-13	✓	0.610	0.1230	0.2157
ce_adaptive_dice_focal	ce_focal	Wilcoxon	1.039e-11	✓	0.528	0.1556	0.2716
ce_adaptive_dice_focal	ce_tversky	Wilcoxon	6.537e-11	✓	0.561	0.1429	0.2554
ce_adaptive_dice_focal	dice	Wilcoxon	3.372e-11	✓	0.561	0.1202	0.2168
ce_adaptive_dice_focal	hybrid	Wilcoxon	1.770e-05	✓	0.301	0.0575	0.1417

TABLE XII: Pairwise statistical tests on CVC-ClinicDB for metric dice - top model (ce_adaptive_dice_focal) in its best epoch (29).

Top Model	Other Model	Test	p-value	Significant	Effect Size	CI Low	CI High
ce_adaptive_dice_focal	ce	Wilcoxon	3.143e-14	✓	0.570	0.2254	0.3527
ce_adaptive_dice_focal	ce_adaptive	Wilcoxon	2.876e-07	✓	0.388	0.0478	0.1067
ce_adaptive_dice_focal	ce_adaptive_dice_tversky	Wilcoxon	1.015e-04	✓	0.311	0.0170	0.0734
ce_adaptive_dice_focal	ce_dice	Wilcoxon	2.370e-13	✓	0.517	0.1649	0.2716
ce_adaptive_dice_focal	ce_dice_focal	Wilcoxon	3.640e-11	✓	0.570	0.0980	0.1855
ce_adaptive_dice_focal	ce_dice_tversky	Wilcoxon	2.439e-13	✓	0.607	0.1230	0.2157
ce_adaptive_dice_focal	ce_focal	Wilcoxon	1.048e-11	✓	0.525	0.1556	0.2716
ce_adaptive_dice_focal	ce_tversky	Wilcoxon	6.703e-11	✓	0.554	0.1429	0.2554
ce_adaptive_dice_focal	dice	Wilcoxon	3.400e-11	✓	0.557	0.1202	0.2168
ce_adaptive_dice_focal	hybrid	Wilcoxon	1.780e-05	✓	0.306	0.0575	0.1417

TABLE XIII: Pairwise statistical tests on CVC-ClinicDB for metric f1 - top model (ce_adaptive_dice_focal) in its best epoch (29).

Top Model	Other Model	Test	p-value	Significant	Effect Size	CI Low	CI High
ce_adaptive_dice_focal	ce	Wilcoxon	1.021e-15	✓	0.678	0.2256	0.3509
ce_adaptive_dice_focal	ce_adaptive	Wilcoxon	2.820e-06	✓	0.327	0.0780	0.1811
ce_adaptive_dice_focal	ce_adaptive_dice_tversky	Wilcoxon	2.405e-11	✓	0.675	0.0570	0.1354
ce_adaptive_dice_focal	ce_dice	Wilcoxon	1.043e-15	✓	0.729	0.1681	0.2802
ce_adaptive_dice_focal	ce_dice_focal	Wilcoxon	1.709e-18	✓	0.765	0.2734	0.3860
ce_adaptive_dice_focal	ce_dice_tversky	Wilcoxon	1.767e-20	✓	0.897	0.2265	0.3200
ce_adaptive_dice_focal	ce_focal	Wilcoxon	2.495e-10	✓	0.466	0.2036	0.3445
ce_adaptive_dice_focal	ce_tversky	Wilcoxon	5.782e-19	✓	0.916	0.2498	0.3513
ce_adaptive_dice_focal	dice	Wilcoxon	2.227e-14	✓	0.712	0.1559	0.2721
ce_adaptive_dice_focal	hybrid	Wilcoxon	1.536e-02	✓	-0.363	0.0039	0.1164

TABLE XIV: Pairwise statistical tests on CVC-ClinicDB for metric precision - top model (ce_adaptive_dice_focal) in its best epoch (27).

Top Model	Other Model	Test	p-value	Significant	Effect Size	CI Low	CI High
ce_adaptive_dice_tversky	ce	Wilcoxon	1.898e-19	✓	0.880	0.2062	0.3147
ce_adaptive_dice_tversky	ce_adaptive	Wilcoxon	2.368e-06	✓	0.362	0.0393	0.1052
ce_adaptive_dice_tversky	ce_adaptive_dice_focal	Wilcoxon	1.048e-03	✓	0.292	0.0069	0.0625
ce_adaptive_dice_tversky	ce_dice	Wilcoxon	3.925e-11	✓	0.579	0.1076	0.2084
ce_adaptive_dice_tversky	ce_dice_focal	Wilcoxon	4.207e-14	✓	0.656	0.1293	0.2343
ce_adaptive_dice_tversky	ce_dice_tversky	Wilcoxon	1.409e-02	✓	0.155	0.0381	0.1358
ce_adaptive_dice_tversky	ce_focal	Wilcoxon	2.419e-18	✓	0.826	0.1845	0.2932
ce_adaptive_dice_tversky	ce_tversky	Wilcoxon	4.846e-02	✓	0.054	0.0383	0.1355
ce_adaptive_dice_tversky	dice	Wilcoxon	2.927e-06	✓	0.378	0.0607	0.1580
ce_adaptive_dice_tversky	hybrid	Wilcoxon	6.641e-22	✓	1.000	0.3156	0.4077

TABLE XV: Pairwise statistical tests on CVC-ClinicDB for metric recall - top model (ce_adaptive_dice_tversky) in its best epoch (30).

Top Model	Other Model	Test	p-value	Significant	Effect Size	CI Low	CI High
ce_adaptive_dice_focal	ce	Wilcoxon	6.165e-09	✓	0.667	0.0103	0.0199
ce_adaptive_dice_focal	ce_adaptive_dice	Wilcoxon	2.539e-03	✓	0.167	0.0036	0.0179
ce_adaptive_dice_focal	ce_adaptive_dice_tversky	Wilcoxon	2.792e-01		0.189	-0.0078	0.0124
ce_adaptive_dice_focal	ce_dice	Wilcoxon	9.437e-07	✓	0.568	0.0127	0.0348
ce_adaptive_dice_focal	ce_dice_focal	Wilcoxon	2.720e-06	✓	0.534	0.0088	0.0247
ce_adaptive_dice_focal	ce_dice_tversky	Wilcoxon	6.867e-07	✓	0.627	0.0250	0.0588
ce_adaptive_dice_focal	ce_focal	Wilcoxon	1.602e-08	✓	0.600	0.0102	0.0197
ce_adaptive_dice_focal	ce_tversky	Wilcoxon	8.013e-02		0.333	-0.0104	0.0141
ce_adaptive_dice_focal	dice	paired t-test	2.746e-03	✓	0.355	0.0100	0.0458
ce_adaptive_dice_focal	hybrid	Wilcoxon	1.781e-03	✓	0.405	0.0018	0.0094

TABLE XVI: Pairwise statistical tests on CVC-ColonDB for metric accuracy - top model (ce_adaptive_dice_focal) in its best epoch (30).

Top Model	Other Model	Test	p-value	Significant	Effect Size	CI Low	CI High
ce_adaptive_dice_tversky	ce	Wilcoxon	1.756e-12	✓	1.000	0.0240	0.0417
ce_adaptive_dice_tversky	ce_adaptive_dice	Wilcoxon	2.715e-03	✓	0.448	0.0011	0.0224
ce_adaptive_dice_tversky	ce_adaptive_dice_focal	Wilcoxon	3.065e-02	✓	0.322	-0.0020	0.0179
ce_adaptive_dice_tversky	ce_dice	Wilcoxon	3.550e-06	✓	0.556	0.0133	0.0314
ce_adaptive_dice_tversky	ce_dice_focal	Wilcoxon	1.285e-08	✓	0.754	0.0174	0.0350
ce_adaptive_dice_tversky	ce_dice_tversky	Wilcoxon	9.491e-06	✓	0.556	0.0101	0.0293
ce_adaptive_dice_tversky	ce_focal	Wilcoxon	1.756e-12	✓	1.000	0.0240	0.0417
ce_adaptive_dice_tversky	ce_tversky	Wilcoxon	3.688e-04	✓	0.302	0.0108	0.0295
ce_adaptive_dice_tversky	dice	Wilcoxon	5.593e-03	✓	0.229	0.0086	0.0270
ce_adaptive_dice_tversky	hybrid	Wilcoxon	9.395e-03	✓	0.368	-0.0012	0.0160

TABLE XVII: Pairwise statistical tests on CVC-ColonDB for metric bf1 - top model (ce_adaptive_dice_tversky) in its best epoch (30).

Top Model	Other Model	Test	p-value	Significant	Effect Size	CI Low	CI High
ce_adaptive_dice_tversky	ce	Wilcoxon	1.756e-12	✓	1.000	0.0124	0.0218
ce_adaptive_dice_tversky	ce_adaptive_dice	Wilcoxon	3.374e-03	✓	0.439	0.0004	0.0117
ce_adaptive_dice_tversky	ce_adaptive_dice_focal	Wilcoxon	3.188e-02	✓	0.322	-0.0012	0.0094
ce_adaptive_dice_tversky	ce_dice	Wilcoxon	3.550e-06	✓	0.556	0.0070	0.0165
ce_adaptive_dice_tversky	ce_dice_focal	Wilcoxon	1.285e-08	✓	0.754	0.0090	0.0184
ce_adaptive_dice_tversky	ce_dice_tversky	Wilcoxon	9.491e-06	✓	0.556	0.0052	0.0155
ce_adaptive_dice_tversky	ce_focal	Wilcoxon	1.756e-12	✓	1.000	0.0124	0.0218
ce_adaptive_dice_tversky	ce_tversky	Wilcoxon	3.688e-04	✓	0.302	0.0057	0.0156
ce_adaptive_dice_tversky	dice	Wilcoxon	5.505e-03	✓	0.229	0.0046	0.0143
ce_adaptive_dice_tversky	hybrid	Wilcoxon	9.982e-03	✓	0.368	-0.0008	0.0085

TABLE XVIII: Pairwise statistical tests on CVC-ColonDB for metric biou - top model (ce_adaptive_dice_tversky) in its best epoch (30).

Top Model	Other Model	Test	p-value	Significant	Effect Size	CI Low	CI High
ce_adaptive_dice_tversky	ce	Wilcoxon	1.912e-13	✓	0.836	0.4153	0.5578
ce_adaptive_dice_tversky	ce_adaptive_dice	paired t-test	9.995e-06	✓	0.544	0.1096	0.2685
ce_adaptive_dice_tversky	ce_adaptive_dice_focal	Wilcoxon	5.034e-02		0.147	0.0074	0.1252
ce_adaptive_dice_tversky	ce_dice	Wilcoxon	4.903e-09	✓	0.680	0.1661	0.3077
ce_adaptive_dice_tversky	ce_dice_focal	Wilcoxon	1.048e-12	✓	0.760	0.2831	0.4182
ce_adaptive_dice_tversky	ce_dice_tversky	paired t-test	8.369e-06	✓	0.549	0.1014	0.2461
ce_adaptive_dice_tversky	ce_focal	Wilcoxon	1.912e-13	✓	0.836	0.4153	0.5578
ce_adaptive_dice_tversky	ce_tversky	Wilcoxon	1.944e-03	✓	0.211	0.0662	0.2318
ce_adaptive_dice_tversky	dice	paired t-test	1.281e-03	✓	0.384	0.0510	0.2011
ce_adaptive_dice_tversky	hybrid	Wilcoxon	6.703e-05	✓	0.387	0.0580	0.1867

TABLE XIX: Pairwise statistical tests on CVC-ColonDB for metric dice - top model (ce_adaptive_dice_tversky) in its best epoch (30).

Top Model	Other Model	Test	p-value	Significant	Effect Size	CI Low	CI High
ce_adaptive_dice_tversky	ce	Wilcoxon	8.488e-14	✓	1.000	0.4153	0.5578
ce_adaptive_dice_tversky	ce_adaptive_dice	paired t-test	9.995e-06	✓	0.544	0.1096	0.2685
ce_adaptive_dice_tversky	ce_adaptive_dice_focal	Wilcoxon	4.524e-02	✓	0.206	0.0074	0.1252
ce_adaptive_dice_tversky	ce_dice	Wilcoxon	5.005e-09	✓	0.714	0.1661	0.3077
ce_adaptive_dice_tversky	ce_dice_focal	Wilcoxon	7.147e-13	✓	0.884	0.2831	0.4182
ce_adaptive_dice_tversky	ce_dice_tversky	paired t-test	8.369e-06	✓	0.549	0.1014	0.2461
ce_adaptive_dice_tversky	ce_focal	Wilcoxon	8.488e-14	✓	1.000	0.4153	0.5578
ce_adaptive_dice_tversky	ce_tversky	Wilcoxon	1.403e-03	✓	0.353	0.0662	0.2318
ce_adaptive_dice_tversky	dice	paired t-test	1.281e-03	✓	0.384	0.0510	0.2011
ce_adaptive_dice_tversky	hybrid	Wilcoxon	5.733e-05	✓	0.465	0.0580	0.1867

TABLE XX: Pairwise statistical tests on CVC-ColonDB for metric f1 - top model (ce_adaptive_dice_tversky) in its best epoch (30).

Top Model	Other Model	Test	p-value	Significant	Effect Size	CI Low	CI High
ce_adaptive_dice_focal	ce	Wilcoxon	7.220e-13	✓	1.000	0.6090	0.8049
ce_adaptive_dice_focal	ce_adaptive_dice	Wilcoxon	5.309e-04	✓	0.490	0.0640	0.2604
ce_adaptive_dice_focal	ce_adaptive_dice_tversky	Wilcoxon	6.912e-06	✓	0.645	0.0502	0.2280
ce_adaptive_dice_focal	ce_dice	Wilcoxon	9.872e-08	✓	0.562	0.2318	0.4273
ce_adaptive_dice_focal	ce_dice_focal	Wilcoxon	2.189e-06	✓	0.562	0.1888	0.4089
ce_adaptive_dice_focal	ce_dice_tversky	Wilcoxon	8.756e-08	✓	0.623	0.2304	0.4433
ce_adaptive_dice_focal	ce_focal	Wilcoxon	7.220e-13	✓	1.000	0.6090	0.8049
ce_adaptive_dice_focal	ce_tversky	Wilcoxon	3.205e-07	✓	0.631	0.1781	0.3643
ce_adaptive_dice_focal	dice	Wilcoxon	2.266e-07	✓	0.569	0.2120	0.4110
ce_adaptive_dice_focal	hybrid	Wilcoxon	2.168e-03	✓	0.379	0.0299	0.2156

TABLE XXI: Pairwise statistical tests on CVC-ColonDB for metric precision - top model (ce_adaptive_dice_focal) in its best epoch (28).

Top Model	Other Model	Test	p-value	Significant	Effect Size	CI Low	CI High
ce_tversky	ce	Wilcoxon	7.196e-14	✓	1.000	0.5887	0.7437
ce_tversky	ce_adaptive_dice	Wilcoxon	9.367e-09	✓	0.622	0.2255	0.4101
ce_tversky	ce_adaptive_dice_focal	Wilcoxon	3.564e-07	✓	0.571	0.1179	0.2689
ce_tversky	ce_adaptive_dice_tversky	Wilcoxon	5.550e-07	✓	0.652	0.1062	0.2541
ce_tversky	ce_dice	Wilcoxon	1.695e-12	✓	0.884	0.3851	0.5537
ce_tversky	ce_dice_focal	Wilcoxon	4.116e-12	✓	0.855	0.2630	0.4133
ce_tversky	ce_dice_tversky	Wilcoxon	1.258e-08	✓	0.686	0.1627	0.3072
ce_tversky	ce_focal	Wilcoxon	7.196e-14	✓	1.000	0.5887	0.7437
ce_tversky	dice	Wilcoxon	9.011e-01		-0.143	-0.0664	0.0855
ce_tversky	hybrid	Wilcoxon	4.888e-13	✓	0.884	0.3815	0.5231

TABLE XXII: Pairwise statistical tests on CVC-ColonDB for metric recall - top model (ce_tversky) in its best epoch (27).