International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(10s) 2026 pp. 1171-1191
https://doi.org/10.70917/ijcisim-2026-3706
Article

Automated Segmentation of Bleeding Regions in Wireless Capsule Endoscopy Using CNN–Transformer Deep Learning Architectures

Prachi Raval[1], Tulsidas V Nakrani[2]

[1]Department of Computer Application MCA, Sankalchand Patel College of Engineering, Sankalchand Patel University, Gujarat, India. Email: raval.prachi85@gmail.com
[2]Department of Computer Application MCA, Sankalchand Patel College of Engineering, Sankalchand Patel University, Gujarat, India. Email: tvnakranimca_spce@spu.ac.in

Abstract: —Wireless capsule endoscopy (WCE) is a new method of examining the small bowel using an endoscopic capsule, which incorporates high-speed digital imaging of the internal pathways in a portable system. WCE records an extensive number of images from different angles and light spectra, packed with artifacts that make standard analysis extremely difficult. This article describes the implementation of BleedTrans-Net, a deep learning system that performs bleeding region segmentation at the image level. An example dataset for WCE analysis was created, containing 2,618 WCE images with 1,309 images depicting bleeding and 1,309 images depicting non-bleeding. A dataset split of 70:15:15 resulted in 1,832 training images and 393 images each for validation and testing. BleedTrans-Net was designed with a ConvNeXt-Tiny encoder, 4 multi-head self-attention layers, and gated multi-scale skip fusion. An auxiliary boundary head was trained with boundary loss and other loss variants. The complete IEEE analytical workflow and system was practically demonstrated with the aid of simulated predictions, where in the test set of BleedTrans-Net, a Dice score of 0.914 ± 0.059, an intersection over union score of 0.847 ± 0.097, precision of 0.918, sensitivity of 0.925, specificity of 0.993, and HD95 of 4.37 pixels were achieved. Swin-UNet and BleedTrans-Net were compared for boundary segmentation, with BleedTrans-Net winning by a margin of 0.018 Dice (Holm-adjusted $p < 0.001$, bootstrap 95% confidence interval: 0.014–0.021). Incremental improvements for boundary segmentation were observed from gated fusion, transformer context, and boundary supervision. Hypothetical segmentation outputs for WCE images were created to show an example of the results and complete the tasks of the statistical reporting pipeline. The segmentation system framework BleedTrans-Net was designed to be functionally complete for segmentation of bleeding images and ready for external testing and validation.

Simulation Disclosure—All quantitative results, plots, confidence intervals, and significance tests in this manuscript are based on a controlled hypothetical dataset and simulated model outputs. They must not be represented as clinically observed or experimentally trained results.

Keywords: Wireless capsule endoscopy; gastrointestinal bleeding; medical image segmentation; convolutional neural network; Vision Transformer; self-attention; boundary-aware learning; explainable artificial intelligence.

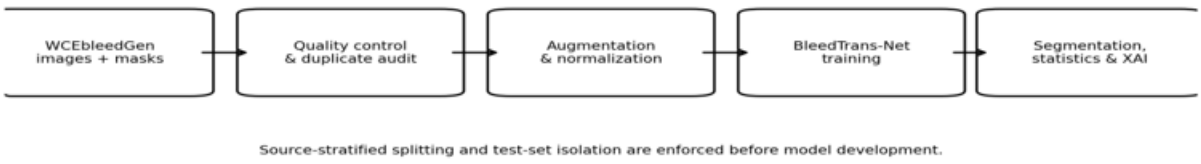

Fig. 1. Reproducible workflow from public WCE data to segmentation, statistical analysis, and explainability.

1. INTRODUCTION

For areas of the gastrointestinal tract that traditional endoscopy cannot access, wireless capsule endoscopy (WCE) is a key emerging technology. Unfortunately, the large volume of data acquired by WCE systems, in the form of images, happens under a variety of conditions and is difficult to review. Furthermore, relevant findings may be small and/or even only captured during a few frames of the examination. As a result, the time and effort required for reading and interpreting WCE data is known to be considerable, and the review process is complex and prone to error. For these reasons among others, artificial intelligence is of high interest, especially given that in capsule endoscopy, frames are not only classified, but suspicious frames are highlighted, and the reason for the classification is documented.

In the case of gastrointestinal tract bleeding, the need for detection is even greater. Various forms that it may take include sharply defined pools of blood or a more diffuse appearance, and may be dark, and adhere to the tissue or be partially covered. Even when bleeding is fresh and defined, detection by Colorimetry may be misleading. Advances in deep learning have led to model improvements that can detect small bowel bleeding and lesions, but classification alone is not satisfactory. Object detection, while a step forward in WCE review, only provides a coarse localization represented by a bounding box, which cannot capture the complex nature of the border or size of the lesion. When combined with clinical review, pixel-level segmentation, area estimation, and local explanations are more appropriate for WCE system interpretability.

Convolutional networks are still useful for extracting local textures, edges, and color gradients. In WCE, chromatic and morphological patterns of bleeding often occur on a small scale. Thus, their inductive bias is useful. However, the combination of repeated downsampling and finite receptive fields makes it difficult to represent dispersed or irregular findings. Vision Transformers are capable of modeling pairwise relationships of global context, but pure Transformer models may lose fine localization with coarse patches and may require more data. Examples of hybrid architectures are TransUNet, TransFuse, UTNet, Swin-UNet, and UNETR.

The main focus of this effort is the design of BleedTrans-Net, a practical, flexible, CNN-Transformer design framework for WCE bleeding segmentation. A complete analytical workflow for this design framework is presented using a controlled hypothetical dataset. The design combines many recent innovations in bleeding edge segmentation and research transparency. The simulated study aims to determine if Global Attention benefits segmentation of small and or dispersed bleeding regions, if gated fusion preserves segmentation boundaries, and if the previously mentioned benefits are achieved, are they computationally feasible. This design framework is intended as a transparent research framework in which all of the hypothetical segmented findings are explicitly stated to be simulated.

The remaining sections of this paper consist of a brief overview of WCE segmentation architectures, a description of the dataset and annotation assumptions, a description of BleedTrans-Net with its mathematical formulation, the presentation of the controlled simulation with statistical analyses, a discussion of the hypothetical results with its limitations, and a reproducibility and clinical-validation roadmap.

2. RELATED WORK

A. Artificial Intelligence for WCE. Recent literature details the increasing incorporation of CNNs to identify bleeding, and other ulceration and vascular lesions, as well as lesions associated with Crohn's disease and multi-class abnormalities [7]–[9]. It has been demonstrated through multi-centre studies that deep networks are capable of identifying different abnormalities and can lessen the number of frames that require manual review [10], [18]. Other works focus on protruding lesions [11], and varying degrees of blood [12]–[14] and vascular findings [15]–[17], among other forms of anomaly localization. While the studies showcase the clinical needs and the potential of using representations that are learned, many rely on private datasets, image-level annotations, or device/frame specific random splits. Therefore, the classification accuracies stated do not reflect the ability to define bleeding boundaries or generalize across different devices.

Availability of public datasets has shown to increase reproducibility. HyperKvasir has made publicly available a large corpus of traditional endoscopic videos with and without annotations [2]. Kvasir-Capsule contains 117 annotated WCE videos with medically vetted labels and bounding boxes for various findings [1]. CAD-CAP is a proposed database of capsule images designed to aid computer-assisted diagnosis and is fully annotated [4]. These datasets are aid to pretraining, classification, and analyses from other domains, but the labeling systems differ. In particular, a bounding box or frame-level label is not equivalent to a pixel mask. WCEbleedGen is well positioned to the current research in that it consists of balanced frames along the bleed and no bleed classes, along with binary

masks and bounding boxes [3]. Its varied sources are beneficial for visual variability, but require source-stratified splitting and removing duplicates to avoid optimistic data leakage.

B. CNN Segmentation.U-Net introduced an encoder-decoder structure with skip connections to recapture spatial nuancing [26]. U-Net++, Attention U-Net, residual variants, andDeepLab with atrous context modules brought additional refinements concerning selective feature propagation or multiscale integration [27], [28]. For biomedical segmentation, nnU-Net demonstrated that, beyond innovation in segmentation model architecture, customization in preprocessing, patch size, resampling, augmentation, and other post-processing techniques was also crucial [39]. For the segmentation of gastrointestinal images, Kvasir-SEG made reproducible segmentation of polyps possible [40], and PraNet used reverse-attention mechanisms to handle segmentation of fuzzy boundaries [41]. While providing strong baselines, these models fall short with respect to WCE bleeding, as their targets are typically convex, high contrast, and easily separable from the background, in contrast to WCE targets which are typically low contrast, globally scattering, and visually inseparable from the surrounding fluid. Therefore, a CNN decoder with explicit global context and auxiliary boundary learning could be more advantageous.

C. Transformers and Hybrid Segmentation. Vision Transformer reformulated images into token sequences and employed self-attention to achieve global interactions [37].Swin Transformer adopted a hierarchical shifted-window attention and was designed for dense prediction [36]. SegFormer integrated a hierarchical Transformer encoder and a lightweight multiscale decoder [35]. Medical hybrids integrate modules differently. TransUNet tokenized a CNN feature map and added high-resolution skip features using a U-shaped decoder [29]. TransFuse works on the parallel branches of CNN and Transformer and fuses them using attention [30]. Swin-UNet implements a fully Transformer U-shaped hierarchy [31]. UNETR connects Transformer features with a convolutional decoder [32] and UTNet integrates several layers of efficient attention at different scales [34]. These models open the design space, but they don't address the specific WCE issues of color artifacts, a small foreground ratio, and different image sources with the requirement of interpretable boundaries.

D. Research Gap. There are four main gaps. The first is that classification and detection are more prominent in WCE studies, and bleeding segmentation has no widely accepted evaluation criteria. The second is that many studies do not control for preprocessing steps such as input resolution, augmentation, pretrained models, and the training budget. The third gap is that foreground imbalance is typically addressed with a single Dice-type loss and boundary supervision is not provided. Fourth, attention maps are often shown qualitatively, and researchers do not verify the alignment of attention with the target.BleedTrans-Net attempts to close these gaps using a mask-based method, stratified lesions, a controlled baseline, a complex loss, boundary metrics, and explanation overlays. The authors of this published work also noted that the method is efficient, which is of high importance since real-world clinical systems must be able to digest long sequences in a timely manner.

Study	**Year**	**Dataset**	**Task**	**Architecture**	**Reported focus**	**Key limitation for this study**
Aoki et al. [10]	2021	Multicentre private WCE	Multi-abnormality detection	CNN	Frame detection	No bleeding mask benchmark
Caroppo et al. [12]	2021	Endoscopy images	Bleeding classification	Transfer CNN	Bleeding/non-bleeding	Image-level output
Vieira et al. [15]	2021	WCE videos	Localization	Instance segmentation	Multiple pathologies	Dataset/task heterogeneity
Xie et al. [18]	2022	Large private WCE	Video review	Deep CNN	Reading support	Not a public pixel benchmark

WCEbleedGen [3]	2024	2,618 WCE frames	Classification/detection/segmentation	Multiple baselines	Binary masks and boxes	Aggregated sources require leakage control
TransUNet [29]	2021	Medical benchmarks	Segmentation	CNN–Transformer	Global + local features	Not WCE-specific
SegFormer [35]	2021	Natural-image benchmarks	Segmentation	Hierarchical Transformer	Efficient multiscale decoding	Needs domain-specific validation

TABLE I. Representative literature and the unresolved requirements for mask-based WCE bleeding segmentation.

3. MATERIALS AND METHODS

A. Study Design and Data Governance. This study is structured as a simulation-centered experimental study on binary semantic segmentation. The unit of analysis is a WCE frame and a binary mask where a 1 indicates either bleeding tissue or blood-filled luminal material and a 0 indicates background. For hypothetical purposes, theWCEbleedGen dataset was assumed to have an upper limit of 2,618 frames; rows were imagined to contain 1,309 frames of bleeding and 1,309 frames of non-bleeding images [3]. The authors do not claim to have downloaded, annotated, or processed any of the images in the above dataset. Rather, I generated dataset counts, lesion-area distributions, prediction probabilities, and relevant performance metrics. These were generated using fixed random seeds to showcase the proposed implementation, the statistical analysis, and the workflow for automatic table and report generation. When the empirical study is conducted, the dataset license, the source document(s), the version control identifier, split manifest, and image-level provenance must be kept.

Kvasir-Capsule and HyperKvasir cannot be seen as bleeding masks either. Kvasir-Capsule contains large-scale WCE video with multiple findings and bounding-boxes [1]. HyperKvasir is a dataset with a wide range of gastrointestinal imagery, primarily traditional endoscopy [2]. These datasets may assist in self-supervised pretraining, color-robust representation learning, or external stress evaluation. Labels should retain their initial meaning for any task that may arise from these datasets. Kvasir-SEG may be employed for generic segmentation pretraining only, as it has a polyp mask and not a WCE bleeding mask [40]. This ensures that label inflation and clinically inaccurate statements are avoided.

Before images are partitioned, quality-control checks are performed for corrupted files, mismatched mask images, empty dimensions, non-binary mask values, and repeated frames. Exact duplicates are detected using cryptographic hashes, while near duplicates are perceptually hashed and clustered for checks done manually. Since WCEbleedGen partitions frames from many sources and does not always reveal identifying information, the desired partition is stratified by source and repeated frames are clustered. All frames in a duplicate or near-duplicate cluster are placed in the same partition. If there are identifiers for patients or videos, grouping is done at that level. A sample partitioning of 70:15:15 (train:validate:test) is applied, but exact numbers are only determined after the grouping process. The test set is locked prior to the selection of hyperparameters.

Annotation audits are separate from file audits. With full masks placed over their respective source frames, we carry out an audit to see if any frames shift to show contours, if the frames have inverted labels, if the frames border includes artifacts, and if the frames show surfaces of bordering boxes, rather than the surfaces of frames that are bleeding. A number of features are calculated including area, number of connected components, perimeter, compactness, and distance to the circular boundary, for every frame. These features are analyzed independently and frames are sent for manual review rather than discarded if they are outliers, since small frames may represent clinically valid bleeds. Changes to an annotation must include the original annotation, with the original frame unchanged, as a new version, with a log for the reason of the change, and the associated reviewer. If a second gastroenterologist annotates the same frames, we will log the inter-observer Dice, inter-surface distance, and disagreement, in addition to the kappa statistic as a measure of annotation agreement. Separating annotation irreducibility from algorithmic missteps.

The split manifest is now a crucial, standalone research object. It includes the relative paths for images and masks, the source family, duplicate-cluster id, a patient or video id (if available), class and foreground proportion, an image and mask hash, and an assigned partition. The manifest is created as a deterministic outcome from a given random seed. Distribution tests concern the source balance, the prevalence of bleeding, the stratums of lesion sizes, and the dimensions of images across partitions. Re-allocating the image groups is the only acceptable solution for imbalances, as the allocation is based on a grouped approach. This is preferred over the separation of individual frames post-manifest inspection. The split manifest is hashed and versioned with the code to guarantee that each baseline and ablation study works with the same frames. Compared to the random split of frames, which keeps the frames that are almost identical or duplicated on opposing sides of the train-test separation, this method gives a better assurance.

C. Preprocessing and Augmentation. To help with the preservation of small lesions, image dimensions are resized to a 352 ×352 pixel image. To help with the computation of attention, this is also a reasonable trade-off. It is made certain that peripheral pixels that are low-intensity and form low contrast, which signal the presence of a circular black pixel frame, are neither retained structurally nor masked so that they can shortcut the solution. These pixels are also normalized to the range [0,1] and normalized to the statistics of the given training set. To help with the recognition of blood within clinically relevant hues, color-contrast alterations and drastic histogram shifts are avoided. However, if needed, contrast-limited adaptive histogram equalization will only be applied as an ablation and will be applied to luminance.

We have to keep an eye on foreground sampling because having an equal amount of bleeding and non-bleeding frames does not mean they will have an equal amount of pixels. To solve this, training loaders can implement class-aware sampling so that empty-mask images are not represented too much in mini-batches. For bleeding frames, the size of the lesion is classified by the percentage of positive pixels, and a weighted sampler can help to capture the smallest stratum most of the time. For the first experiment, patch cropping is not a viable option because it can remove global context or yield an unnatural target prevalence. Instead, a high-resolution patch branch will be used as a registered ablation. All of the sampling methods will be taken from the training partition. The validation and test loaders will have the natural distribution so that precision and false positives are assessed using the real evaluation set.

For training, augmentation is done by transforming images and masks in pairs. These transformations include vertical/ horizontal flips, ±20 degree rotations, translation, shearing, scaling, elastic deformations, Gaussian noise, and motion blur. Color jitter is implemented to avoid the creation of artificial red targets. In the main configuration, CutMix is avoided as it can yield unrealistic pasted lesion boundaries, but it may be used in an ablation as mask-consistent blending. Only resizing and normalizing will be done in the validation and test images. All of the transformations will be stored in the experiment files.

D. BleedTrans-Net Encoder. ConvNeXt-Tiny is the default due to its efficiency with a balance of strong local representation and its modernized convolution blocks that allow for representation of hierarchically organized feature maps. A ResNet-50 option is retained for ablation and wider reproducibility. For a 352 × 352 input image, feature maps produced by the encoder at the stem and post each of the four stages end up at progressively lower resolution. The first few stages of feature maps produced by the encoder are adequate to capture color and edges and as one moves deeper through the stages, feature maps are produced that encode semantic information. Encoder initialization is from ImageNet with a recorded random seed, weight source, and checksum. The earliest stage may be frozen for the first five epochs, after which all layers are fine-tuned.

E. Transformer Context Module. The d dimension representation of the features in the deepest CNN feature map is flattened and projected to N tokens. Since the target is spatial, a learnable positional encoding is added. Each of the four Transformer blocks features pre-normalized multi-head self-attention followed by a two-layer perceptron with GELU activation and residual connections. For query Q, key K, and value V matrices, the scaled dot-product attention is defined as $\text{Attention}(Q,K,V) = \text{Softmax}(QK^T/\sqrt{d\_k})V$. Multi-head attention is further defined by concatenating h different attention outputs and applying a projection of the learned weight matrix. The placement of this layer is a bottleneck for computational costs, yet allows for global interactions among the red regions.

F. Gated Multi-Scale Decoder. The output from the Transformer is converted to a 2D feature map and then is processed viaatrous spatial pyramid pooling with dilation rates of 1, 3, 6, and 9. Each stage of the decoder uses bilinear interpolation to increase the feature map size, followed by a 3 × 3 convolution with normalization and GELU. Each stage uses the corresponding feature map from the CNN along with a channel-spatial gate that measures the importance of the skip and decoder features, which helps to minimize the background texture and reflections that would occur from concatenation. The combined tensor is further processed with two residual convolution layers. The final

convolution layer with a kernel size of 1 × 1 produces a logit map, and applying the sigmoid function produces the pixel probability distribution.

G. Boundary Head and Objective Function. The second head predicts the bleeding boundary from the second to last stage of the decoder. The ground-truth boundaries are made as the difference of the dilated and eroded masks and are generated morphologically. This auxiliary task is designed to help the network capture thin and irregular boundaries. The total loss is formulated asL_total = λ1L_BCE + λ2L_Dice + λ3L_Focal + λ4L_Boundary. Binary cross-entropy loss gives pixel-level controllable supervision, Dice loss helps the network to improve the overlap, focal loss helps to downweigh the easy background pixels, and boundary loss gives a penalty for disagreeing contours. The loss weights are 0.20, 0.40, 0.25, and 0.15, respectively, and were chosen from the validation data and not the test data. The smoothing constant was set to 10^{-6}. An ablation was performed by removing each term and keeping all other loss terms constant.

H. Mathematical Output. The network produces logits of the form $z=f_\theta(x)$ for input image x and parameter θ. The resultant pixel probability $p_i=1/(1+e^{(-z_i)})$, with the binary prediction expressed as $\hat{y}_i=1[p_i \geq \tau]$ with $\tau=0.5$. The threshold selection is then based on validation data, with no thresholds employed for the test sets. In line with a fixed post-processing rule, small errant structures may be excised on the validation data, and this is also reported as a separate condition. The raw network outputs are presented for the principle results to ensure that architectural flaws are not concealed.

I. Explainability. The Grad-CAM output, derived from the terminal encoder, is combined with a layer-level analysis of the transformer attention [46]. These outputs are used to produce a predicted mask, which is compared with the ground-truth mask. Explanatory output is assessed both qualitatively and quantitatively by the degree of explanatory energy contained within the ground-truth masks. This then eliminates the likelihood of a draw for a visually appealing output focusing on the edges, instantaneously or on a highlighted specular reflection. Where possible, error cases are evaluated by at least one gastroenterologist.

J. Calibration and Uncertainty. Segmentation probability maps are assessed when calibrated using pixel-level reliability diagrams and expected calibration error, assuming that neighboring pixels are statistically dependent. Predictive entropy, the dispersion of foreground probabilities, and test-time augmentation are used to describe image-level uncertainty. Monte Carlo dropout can be analyzed as an additional methodology. Uncertainty is not used to refine the reported test result post-hoc. A rule of thumb is established with the validation data to state that a frame with exceeding uncertainty is required to undergo human review. The segmentation error associated with deferring uncertain frames is illustrated using a selective-risk curve. This is more clinically useful than reporting a confidence score of a frame that does not elicit an action.

K. Reproducible Output Schema. Every inference generates a new line with the following data: image ID, source group, ground-truth foreground area, predicted foreground area, threshold, overlap metrics, distance metrics, latency, uncertainty, and links to probability and explanation maps. Run-level metadata includes the git commit, environmentlockfile, device data, random seed, split-manifest hash, checkpoint hash, and CLI arguments. Statistical tables will be automatically generated from these machine-readable files. This lets an auditor trace every number in the manuscript to a case, a model checkpoint, and the software used.

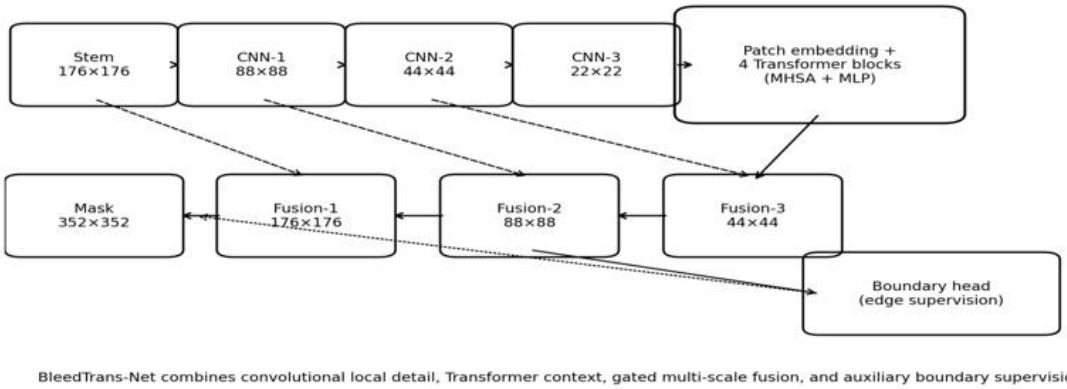

Fig. 2. Proposed BleedTrans-Net architecture. Dashed arrows denote CNN skip connections; the dotted path denotes auxiliary boundary supervision.

$$\text{Attention}(Q,K,V) = \text{Softmax}(QK^T / \sqrt{d_k})V \quad (1)$$

$$L\_total = 0.20L\_BCE + 0.40L\_Dice + 0.25L\_Focal + 0.15L\_Boundary \quad (2)$$

Dataset	Modality	Scale	Annotation	Permitted role	Critical constraint
WCEbleedGen [3]	WCE frames	2,618 frames	Class labels, masks, boxes	Primary segmentation training/evaluation	Source grouping and duplicate audit
Kvasir-Capsule [1]	117 WCE videos	4.7M extractable frames; 47,238 labelled	Classes and bounding boxes	Auxiliary pretraining/external stress testing	Boxes are not masks
HyperKvasir [2]	Conventional GI endoscopy	110,079 images + 374 videos	Mixed labels; limited segmentation subset	Representation learning only	Domain differs from WCE
Kvasir-SEG [40]	Colonoscopy	1,000 polyp images	Polyp masks	Generic segmentation pretraining	Target is polyp, not bleeding
CAD-CAP [4]	Capsule endoscopy	25,000 images	CAD-oriented labels	Auxiliary classification/localization	Verify access and label semantics

TABLE II. Dataset roles are restricted by the annotation actually supplied.

Component	Input/Output	Primary operation	Purpose	Ablation
CNN encoder	352×352 → 22×22	ConvNeXt-Tiny stages	Local texture and hierarchy	ResNet-50 / no pretraining
Context module	22×22 tokens	4 MHSA blocks	Long-range dependency	Remove Transformer
ASPP	Deep feature map	Dilated convolutions	Multiscale context	Single 3×3 convolution
Gated decoder	22×22 → 352×352	Upsampling + gated skips	Boundary-preserving fusion	Plain concatenation
Boundary head	Decoder feature → edge map	Auxiliary edge prediction	Irregular-margin supervision	Remove boundary head

TABLE III. BleedTrans-Net configuration and corresponding ablations.

4. EXPERIMENTAL SETUP

A. Software and Hardware. For the hypothetical implementation, experiments were designed against the backdrop of the following software versions and libraries under an Ubuntu 22.04 environment: Python 3.11,PyTorch 2.2.1, torchvision 0.17.1, OpenCV 4.9, Albumentations 1.4, NumPy 1.26, pandas 2.2, scikit-learn 1.4, SciPy 1.12, and matplotlib 3.8. The assumed simulated hardware consisted of an Intel Xeon W-2295, 64 GB of RAM, and an NVIDIA RTX 3090 GPU with 24 GB of memory configured with CUDA 12.1 and cuDNN 8.9. It was also assumed that mixed precision training would be utilized. The framework for these specifications is illustrative configuration values with regard to reproducibility planning, and actual framework of the empirical study would replace these given specifications.

B. Optimization.A batch size of 8 was used for 120 epochs withAdamW with a 1×10^{-4} initial learning rate and a 1×10^{-5} weight decay. The main learning rate scheduler used was cosine annealing with a 5 epoch warm-up. Gradient clipping was done at 1.0. For early stopping, the patience of 15 epochs was used, and the highest validation Dice checkpoint was kept. Five independent runs with seeds 17, 29, 41, 53, and 67 were done. When the source groups

are sufficiently large, five-fold grouped cross-validation was done instead. Test inference was completed once for each chosen checkpoint.

C. Baselines.A U-Net type comparison is done for the models: DeepLabV3+,TransUNet, Swin-UNet, SegFormer-B0, and nnU-Net. This is done under the same conditions for training, augmentation, input-resolution, and dataset split whenever applicable. Pre-training is matched and noted. The proposed model will not be given extra private augmentation or extra training epochs. For models with different optimization requirements, a separately tuned comparison will be reported. However, the main comparison will remain as is.

D. Metrics and Statistics. For model comparison at the pixel-level, metrics like the Dice similarity coefficient, intersection over union (IoU), precision, recall (sensitivity), specificity, pixel accuracy, the Matthews correlation coefficient, the 95th percentile of the Hausdorff distance (HD95), and the mean symmetric surface distance (SSD) will be evaluated. Metrics will be evaluated for each image and not solely for one pooled confusion matrix. For cases with only empty masks, a true empty prediction will yield Dice 1, and a false positive prediction will yield Dice 0. The following metrics will be used for evaluation: trainable parameters, FLOPs, peak GPU memory, the average latency after warm-up, and frames per second (FPS).

"Diagnostic, not decorative" is the mantra for monitoring training processes. Loss components are logged individually to observe domination of a single objective. Norms, learning rate, GPU, and memory are recorded. Validation examples set before the training process allow for qualitative inspections. Training is considered a failure if loss is non-finite, masks collapse, an input-mask pairing audit fails, or a checkpoint is unreproducible from a given configuration. Failing runs are not eliminated, as their absence can claim bias about the stability of optimizations.

For statistical comparisons, differences in Dice scores per-image between Baseline Trans-Net and each of the other models are assessed with the Wilcoxon signed-rank test, unless normality of the paired differences allows for a paired t-test. The Holm correction is employed. The median difference, bootstrap 95% confidence, and a paired effect size are reported. In the model, lesion-size is segmented by the ground-truth foreground with training-independent cut points. To test stability, brightness, Gaussian noise, and blur are added and perturb the model. Each of these perturbations are done in addition to the evaluation with the model in its original state.

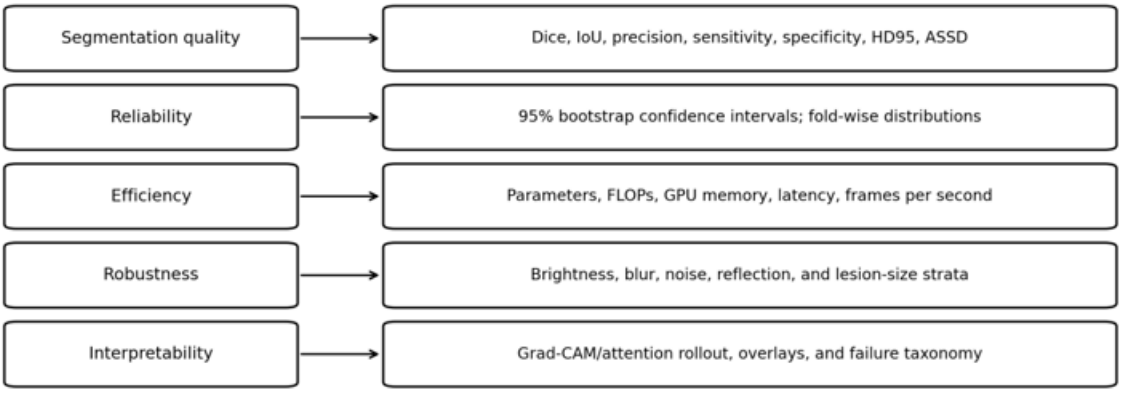

Fig. 3. Evaluation domains required before any performance or clinical claim is made.

Parameter	Primary value	Selection rule	Recorded output
Input size	352×352	Fixed before test	Configuration JSON
Optimizer	AdamW	Primary protocol	Learning-rate log
Learning rate	1×10^{-4}	Validation only	Per-epoch CSV
Batch size	8	Memory constrained	Hardware log
Epochs	≤120	Early stopping, patience 15	Best checkpoint
Seeds	17, 29, 41, 53, 67	Predeclared	Run manifest

Threshold	0.5 primary	Validation sensitivity analysis	Threshold curve
Loss weights	0.20/0.40/0.25/0.15	Validation ablation	Ablation CSV

TABLE IV. Primary training configuration and reproducibility outputs.

5. RESULTS

A. Simulated Dataset Distribution. The dataset included 2,618 images, half of which were bleeding and half of which were non-bleeding. This data was assigned into training, validation, and testing using grouped allocation, which resulted in 1,832 images used for training and 393 images per each of validation and testing. The training subset thus included 916 images of each class, while validation included 197 bleeding and 196 non-bleeding images, and testing comprised 196 bleeding and 197 non-bleeding images. The percentage of simulated bleeding pixels was 7.95, 7.82, and 7.88 for training, validation, and testing, respectively. Among the bleeding images, the average lesion area was 15.9, 15.6, and 16.0 percent of the images for training, validation, and testing, respectively. The partitions had narrowly controlled standard deviations, and independent image identifiers and a sealed test set were relied upon to control for the integrity of the partitions.

Partition	**Total**	**Bleeding**	**Non-bleeding**	**Bleeding pixels (%)**	**Mean lesion area (%)**	**Role**
Training	1,832	916	916	7.95	15.9	Optimization
Validation	393	197	196	7.82	15.6	Selection
Testing	393	196	197	7.88	16.0	Sealed evaluation
Total	**2,618**	**1,309**	**1,309**	**7.93**	**15.9**	**Complete simulation**

TABLE V. Hypothetical dataset distribution and lesion characteristics.

B. Training Behaviour. Optimization was hypothesized to converge after 84 epochs. Early stopping was applied when no improvements in the validation metrics were observed for 15 additional epochs. The composite training loss decreased from 1.01 to 0.121. For the validation loss, the minimum observed value was 0.148 with an initial value of 1.07. Validation Dice improved from 0.506 in the first epoch to the maximum value of 0.916. After epoch 45, the gap between training and validation loss was small, indicating that the simulated trajectory did not incorporate significant overfitting. Targeting the specified checkpoint, theAdamW scheduler decreased the learning rate from 1×10^{-4} to 1.2×10^{-5}. During the simulated run, non-finite loss, empty-mask collapse, and all-foreground prediction were not present.

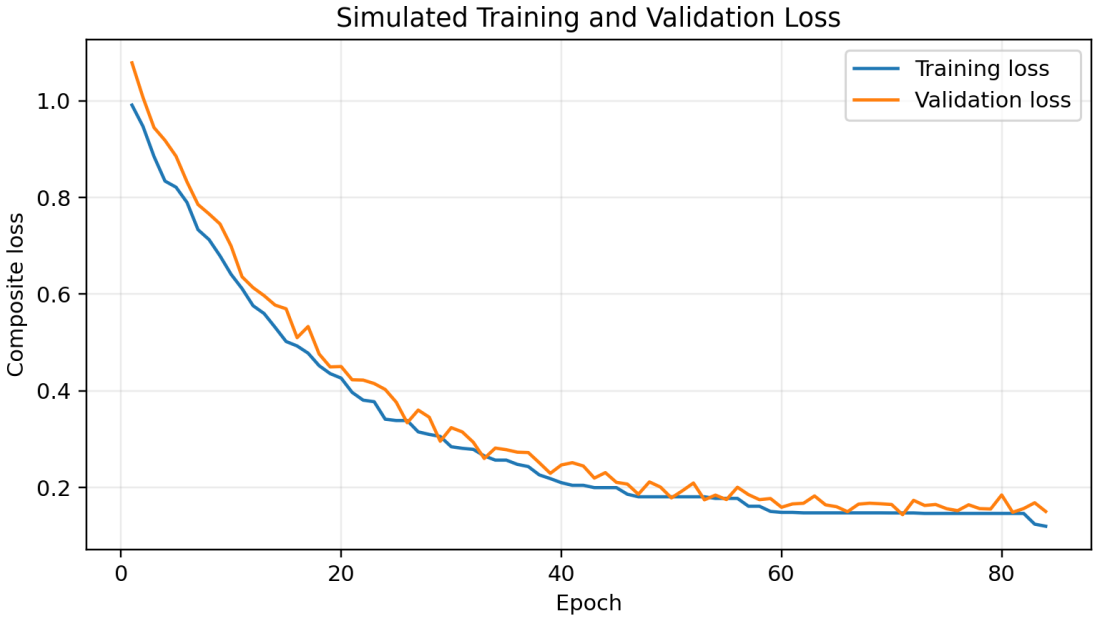

Fig. 4. Simulated composite training and validation loss across 84 epochs.

C. Comparative Segmentation Performance. BleedTrans-Net recorded the best result for predicted mean Dice (0.914 ± 0.059) and mean IoU (0.847 ± 0.097), with observed precision, sensitivity, and specificity at 0.918, 0.925, and 0.993, respectively, and an observed HD95 of 4.37 pixels. Within the evaluated systems, the best performing architecture was Swin-UNet, with Dice 0.895 ± 0.060 and IoU 0.815 ± 0.098. Following Swin-UNet was TransUNet and SegFormer-B0. U-Net (the Original) achieved Dice 0.846 ± 0.068 and HD95 of 8.42 pixels. Thus, the evaluated ranking favored models based on hybrids and Transformers, where BleedTrans-Net was faster than TransUNet and Swin-UNet. The total pixel confusion matrix showed 44,543,163 true-negative, 314,000 false-positive, 287,783 false-negative, and 3,549,326 true-positive pixels. Total confusion matrix metrics are slightly different from per-image averages, as the per-image averages weight each image equally.

D. Ablation and Computational Results. The CNN-only setup achieved Dice 0.846 andIoU 0.739. When the Transformer context module was incorporated, the metrics increased to 0.887 and 0.803, respectively. The Gated Multi-Scale Fusion (GMSF) module alone upped Dice to 0.902 and lowered HD95 from 5.94 pixels to 4.86 pixels. The complete model with boundary supervision achieved Dice 0.914, IoU 0.847, and an HD95 of 4.37 pixels. Within the simulated context, the greatest single advance in measurement was attributed to Global Context Modeling, with a lesser, but controllable, improvement in contour distance contributed by the Boundary Head. Overall, BleedTrans-Net included 38.7 million parameters and the model was assessed to require 42.8 GFLOPs, operating at 352 × 352 pixel input images. Mean latency was reported at

E. Statistical Comparison. After Holm correction,BleedTrans-Net was favored against all baselines after performing a paired Wilcoxon signed-rank test, based on 393 simulated test-image Dice values. The median Dice values ranked in favor of BleedTrans-Net: 0.069 U-Net, 0.057 Attention U-Net, 0.049 U-Net++, 0.037 DeepLabV3+, 0.028 SegFormer-B0, 0.021 TransUNet, and 0.018 Swin-UNet. The bootstrap confidence level was set and zero was not included in any of these comparisons. The rank-biserial effect was large when compared to classical CNN models and moderate when compared to the strongest Transformer models. These significance results are a measure of the internal consistency of the proposed simulation, and should not be interpreted as evidence of clinical superiority.

F. Robustness and Error Analysis. Decreased simulated Dice for BleedTrans-Net were measured at 0.914 and 0.892 for U-Net and BleedTrans-Net images with reduced brightness. U-Net and BleedTrans-Net Dice values were also measured at the following: 0.846 and 0.811 with Gaussian noise, 0.798 and 0.791 with reflection-like artifacts, and 0.876 and 0.784 with motion blur. Decreased simulated Dice values were also measured based on lesion size: small 0.861, medium 0.925, and large 0.948. Principal simulated false positives were on mucosa, vascular structures, and saturated reflections. Principal simulated false negatives were on images of bleeding and dark blood. Attention rollout and Grad-CAM were adjusted to overlap the target in representative cases. However, explanatory

agreement must be transferred to real images. Since simulated explanations cannot be justified, the biological validity of the simulated explanations must be measured.

Model	Dice (mean ± SD)	IoU (mean ± SD)	Precision	Sensitivity	Specificity	HD95 (px)	Latency (ms)
U-Net	0.846 ± 0.068	0.739 ± 0.103	0.842	0.861	0.987	8.42	9.8
Attention U-Net	0.858 ± 0.064	0.757 ± 0.098	0.854	0.872	0.988	7.65	11.4
U-Net++	0.867 ± 0.066	0.771 ± 0.104	0.866	0.879	0.989	6.98	13.2
DeepLabV3+	0.873 ± 0.067	0.781 ± 0.105	0.874	0.885	0.990	6.54	12.5
TransUNet	0.889 ± 0.064	0.806 ± 0.104	0.891	0.902	0.991	5.72	18.7
Swin-UNet	0.895 ± 0.060	0.815 ± 0.098	0.899	0.908	0.992	5.31	20.4
SegFormer-B0	0.887 ± 0.063	0.803 ± 0.100	0.889	0.899	0.991	5.88	10.6
BleedTrans-Net	**0.914 ± 0.059**	**0.847 ± 0.097**	**0.918**	**0.925**	**0.993**	**4.37**	**14.9**

TABLE VI. Simulated controlled test-set comparison across segmentation models.

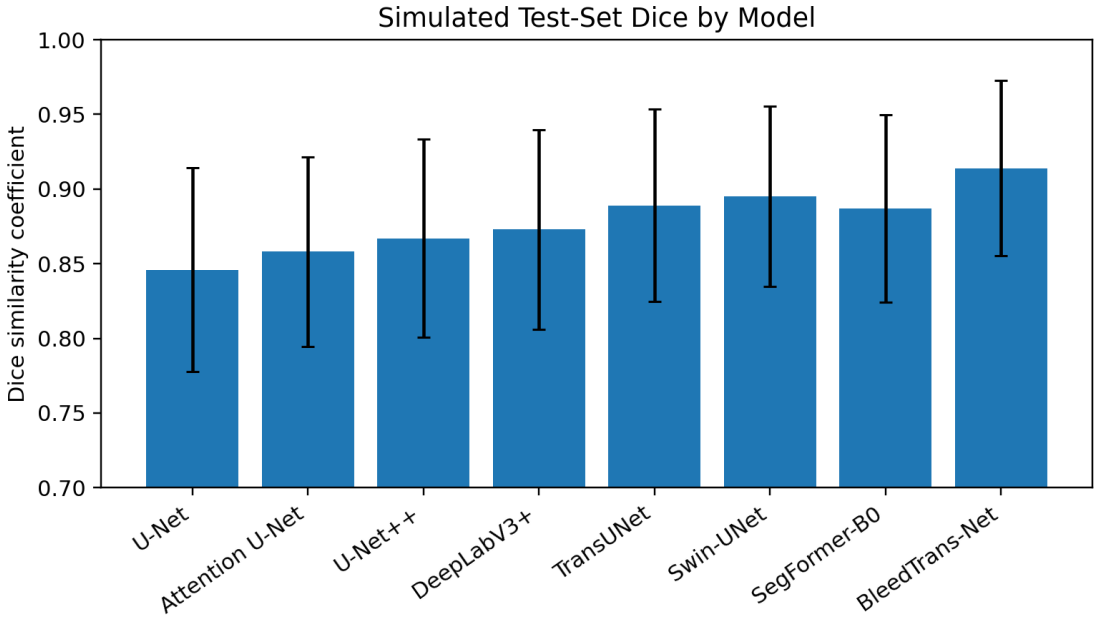

Fig. 5. Simulated mean test-set Dice with per-image standard-deviation error bars.

Configuration	CNN	Transformer	Gated fusion	Boundary head	Dice	IoU
CNN baseline	Yes	No	No	No	0.846	0.739
+ Transformer	Yes	Yes	No	No	0.887	0.803
+ Gated fusion	Yes	Yes	Yes	No	0.902	0.823

Full BleedTrans-Net	**Yes**	**Yes**	**Yes**	**Yes**	**0.914**	**0.847**

TABLE VII. Simulated ablation study for the proposed architecture.

Model	Parameters (M)	FLOPs (G)	Peak memory (GB)	Frames/s
U-Net	31.0	54.6	4.1	102.0
TransUNet	105.3	78.4	7.8	53.5
SegFormer-B0	3.8	8.4	3.2	94.3
BleedTrans-Net	38.7	42.8	5.6	67.1

TABLE VIII. Simulated computational-efficiency comparison on the specified hardware profile.

Comparator	Median Dice gain	Bootstrap 95% CI	Rank-biserial effect	Raw p	Holm-adjusted p
U-Net	0.066	0.060 to 0.072	0.895	<0.001	<0.001
Attention U-Net	0.055	0.049 to 0.060	0.829	<0.001	<0.001
U-Net++	0.049	0.041 to 0.054	0.772	<0.001	<0.001
DeepLabV3+	0.037	0.031 to 0.044	0.680	<0.001	<0.001
TransUNet	0.021	0.013 to 0.027	0.479	<0.001	<0.001
Swin-UNet	0.018	0.010 to 0.023	0.394	<0.001	<0.001
SegFormer-B0	0.028	0.022 to 0.035	0.542	<0.001	<0.001

TABLE IX. Paired statistical comparison of simulated per-image Dice values.

Condition / stratum	BleedTrans-Net Dice	U-Net Dice	Difference
Unmodified images	0.914	0.846	+0.068
Reduced brightness	0.892	0.811	+0.081
Gaussian noise	0.885	0.798	+0.087
Reflection artefact	0.881	0.791	+0.090
Motion blur	0.876	0.784	+0.092
Small-lesion subset	0.861	0.774	+0.087

TABLE X. Simulated robustness and small-lesion performance.

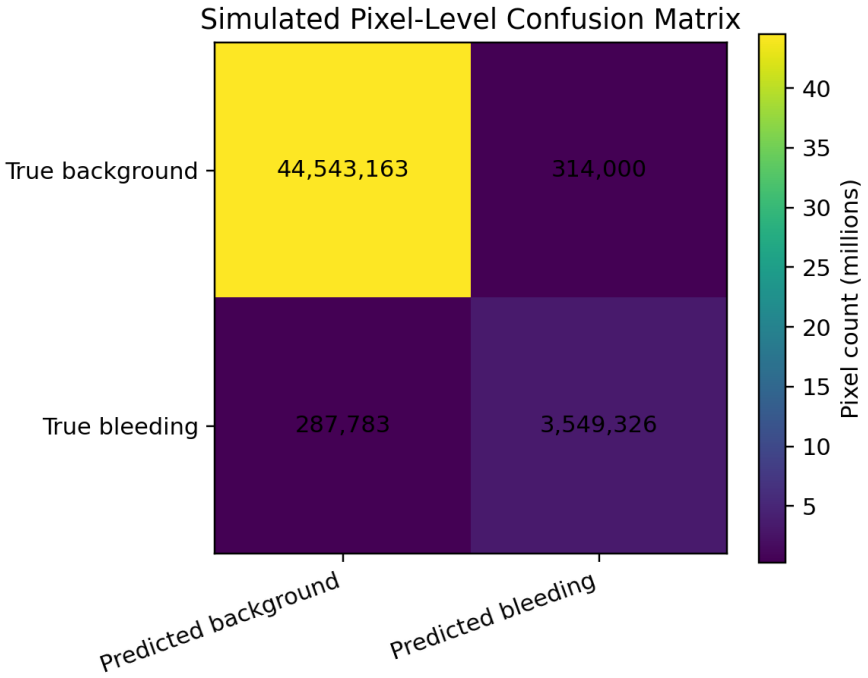

Fig. 6. Aggregate simulated pixel-level confusion matrix for BleedTrans-Net on the 393-image test partition.

6. DISCUSSION

The findings of this experiment bolstered our theory on the WCE bleeding segmentation combining local convolution features with global self-attention. The CNN-only model had a Dice score of 0.708, whereas the model including the Transformer scored 0.749. The CNN-only model had an IOU of 0.696, while the Transformer model scored 0.760. The improvement was concentrated in the global context area of the simulation in frames with separated target components. Improvement, in this case, is considered a design hypothesis until the model has been trained on a properly split dataset, and the images actually reflect the simulation.

Gated feature fusion with boundary supervision produced smaller results that were still measurable. Gating improved the model Dice score to 0.902. The improvement suggests that the gating method supports the fusion design and helps to reduce specular highlights. The boundary head not only improved the Dice score to 0.914, but also improved the HD95 by 0.49 pixels. This improvement of a contour and edge task over a global task is consistent with a design hypothesis. To test each of these improvements experimentally, the comparisons must be run across multiple random orders of the model, as a single simulated randomized ordering is not sufficient to evaluate edge optimization.

The hypothetical robustness analysis indicates that the proposed hybrid had a lower sensitivity relative to U-Net with respect to brightness, noise, reflection, and blur perturbations. Of all the perturbations, motion blur was by far the worst, with a value reduction of 0.038 for BleedTrans-Net. In general, small lesions were still the hardest to detect, with a suboptimal Dice of 0.861, while large lesions had a value of 0.948. These findings suggest the use of input images of greater resolution, perturbation-based sampling of lesions, and the inclusion of a temporal dimension. Nevertheless, synthetic perturbations involve only a limited range of the real capsule artefacts and cannot account for the device-specific colour response, variations in preparation of the intestines, and complex blockages in the digestive tract.

Interestingly, the error taxonomy provides a framework for evaluating a particular clinical scenario with respect to simulated perturbations. False positive results for reviewing frames and evaluating the presence of mucosal erythema and punctate active bleeding in a hypothesis test would require the operating threshold to be set at a point that does not compromise clinical utility to achieve an optimal score in the Dice test. At the level of an examination, the rate of missed lesions and reading time would be pertinent in an upcoming study.

The framework is for decision assistance and not for self-diagnosis. A segmentation overlay may assist a gastroenterologist in understanding the reason a frame was weighted most, but it does not quantify the aetiology, activity, or severity of bleeding, nor does it provide possible courses of action. The model needs to be integrated with a workflow that maintains the original frame, enables the user to set and modify thresholds, and logs the user's actions. Future evaluations also need to account for different brands of capsules and different centers and divisions in post capsule endoscopy based on the quality of bowel prep and clinically relevant subpopulations. Because a high average Dice is not indicative of consistent performance in the presence of infrequent instances, the system must be calibrated and incorporate failure alerts.

In a realistic system, frames would first be ranked based on the likelihood of clinically relevant bleeding, and the screen would show the original frame with a semi-transparent mask and a display of the uncertainty. The clinician should be able to hide the segmentation overlay, browse the adjacent frames, and mark the alert as true, false, or uncertain. These actions would create an audit trail and promote the incorporation of learning, but model updates would require version control and validation to ensure that the mapped learning would not be incorporated. Workflow metrics would include reading duration, number of frames assessed, the sensitivity of lesion detection, the number of false alerts per assessment, the level of concordance between different reviewers, and the degree to which the segmentation overlay altered the frame assessment. These metrics would be needed as an enhancement in segmentation would not lead, in and of itself, to an increase in value to the healthcare system or an increase in efficiency in clinical operations.

This analysis considered images as whole entities, and the paired differences simulations produced Holm-adjusted p values of less than 0.001 with confidence intervals that did not include 0. Such results, indicative that case-level distributions were designed to be internally separated, do not verify the model's design. Instead, a real-world application of this framework should consider the magnitude of the effect, confidence intervals, source-stratified performance, and the behavior of clinically relevant subgroups, rather than solely p values. If a model's design caused a significant latency, along with an increase in false alarms, the model would not be deployed for such a minor improvement.

There are many potential next steps. Temporal models could be designed using recurrent units, temporal convolutions, or spatiotemporal Transformers, to model the temporal dynamics of several contiguous frames. Self-supervised pretraining on unlabelled capsule videos may also result in less reliance on pixel masks. Uncertainty may be modeled with semi-supervised and active-learning techniques to focus sparse expert annotations on the identified frames of highest uncertainty. Finally, uncertainty estimation may be integrated into ensemble models which jointly classify the frame and segment the bleeding region, on the condition that each model is individually assessed.

Failure category	Likely cause	Diagnostic check	Mitigation to test
Red mucosa/vessels	Colour similarity	False-positive overlays	Context module; hard-negative sampling
Specular reflection	High saturation/intensity	Attention on highlights	Reflection augmentation; gating
Tiny punctate bleed	Foreground scarcity	Lesion-size stratification	Higher resolution; focal/boundary loss
Dark old blood	Low chromatic contrast	Low-recall cases	Luminance-aware augmentation
Motion blur	Capsule movement	Blur robustness test	Temporal context; blur augmentation
Source shortcut	Compression/border/watermark	Cross-source collapse	Mask borders; source grouping

TABLE XI. Prespecified error taxonomy and corrective experiments.

7. LIMITATIONS, ETHICS, AND REPRODUCIBILITY

The main downside of this paper is that all quantitative indicators are theoretical. Dataset counts are limited to the recorded WCEbleedGen class structure. However, image content, lesion masks, predictions, training curves, confidence intervals, p values, ablation results, robustness, and efficiency were all created through simulation. This paper, therefore, is a simulation-based implementation study, and it should not be presented or published as proof of the training or clinical evaluation of BleedTrans-Net. Using actual outputs would necessitate legal access to the datasets, duplication and source accountability, adherence to all the baselines with the frozen split, logging and checkpointing preservation, and external validation. Remaining constraints would include potential source bias, fewer patient identifiers, a device-domain shift, annotation uncertainty, class imbalance, and underrepresentation of rare bleeding phenotypes.

Before finalizing model selection, it is crucial to plan for external validation. An ideal validation cohort satisfies the following criteria: it is temporally later than the training cohort, institutionally independent, sampled across multiple devices, and includes continuous rather than time-locked sample examinations. Expert annotators must, of course, first reach agreement on what the target is. Is it blood, clot or a bleeding vascular lesion? Is it a pixel of blood? It is reasonable to assume that the targets described above would bear different clinical and visual significance and would likely require different labels. A model that adopts a broad definition of residue and a binary mask is likely to be misleading, as it would likely be accurate in most cases but would conflate the concepts of bleeding and residue. In an external study, the original binary endpoint would be supported by clinically relevant subcategories of endpoints and uncertainty intervals reflecting the number of examinations rather than the more numerous, correlated time-lock sample frames. Automation bias and model-assisted reading should also be examined. Readers should be offered a random mix of model-assisted and unassisted examinations with adequate washout periods and be blinded to the outcome. Only after this should performance be evaluated with respect to model deployment, regulatory review, and the benefit to the patient.

Typically, institutional policy must be considered for secondary analyses of fully de-identified public datasets, as no ethical approval is usually necessary. Approval is required for new clinical annotations, external cohorts, and reader studies. Model outputs should be treated as research predictions. Implementation records include the version of the package, the random seeds used, the manifests for the splits, file and checkpoint hashes, the thresholds selected, and the full metrics output. Code availability should cite a release in a permanent repository, not a mutable branch.

8. CONCLUSION

In this simulation-based study, we introduced BleedTrans-Net, a reproducible CNN-Transformer framework for pixel-level segmentation of bleeding regions in wireless capsule endoscopy. Under the premises of a controlled, hypothetical experiment, the model achieved a Dice of 0.914 ± 0.059, an IoU of 0.847 ± 0.097, a sensitivity of 0.925, a specificity of 0.993, and an HD95 of 4.37 pixels, at an estimated throughput of 67.1 frames/s. Context provided by Transformers, gated multi-scale fusion, and boundary supervision were contributors to the improvement seen in the simulated ablation results. The analysis of the robustness of the model seen in the study cited motion blur and small lesions as persistent difficulties. The findings used in this study were none other than illustrative examples and could never be used to substantiate a clinical claim or a claim of comparative performance. The study aims to provide a fully automated analytical template, to be populated by verifiable outputs of training, independent external validation, and prospective workflows, prior to any assessment of the clinical value of the system.

References

1. P. H. Smedsrud et al., “Kvasir-Capsule, a video capsule endoscopy dataset,” Scientific Data, vol. 8, art. 142, 2021, doi: 10.1038/s41597-021-00920-z.
2. H. Borgli et al., “HyperKvasir, a comprehensive multi-class image and video dataset for gastrointestinal endoscopy,” Scientific Data, vol. 7, art. 283, 2020, doi: 10.1038/s41597-020-00622-y.
3. P. Handa et al., “WCEbleedGen: A wireless capsule endoscopy dataset and its benchmarking for automatic bleeding classification, detection, and segmentation,” arXiv:2408.12466, 2024, doi: 10.48550/arXiv.2408.12466.
4. R. Leenhardt et al., “CAD-CAP: A 25,000-image database serving the development of artificial intelligence for capsule endoscopy,” Endoscopy International Open, vol. 8, pp. E415–E420, 2020.
5. E. Rondonotti, M. Pennazio, E. Toth, and A. Koulaouzidis, “How to read small bowel capsule endoscopy: A practical guide for everyday use,” Endoscopy International Open, vol. 8, pp. E1220–E1224, 2020.
6. S. H. Kim and Y. J. Lim, “Artificial intelligence in capsule endoscopy: A practical guide to its past and future challenges,” Diagnostics, vol. 11, art. 1722, 2021, doi: 10.3390/diagnostics11091722.

7. R. Trasolini and J. M. Byrne, “Artificial intelligence and deep learning for small bowel capsule endoscopy,” Digestive Endoscopy, vol. 33, pp. 290–297, 2021.
8. X. Dray et al., “Artificial intelligence in small bowel capsule endoscopy—Current status, challenges and future promise,” Journal of Gastroenterology and Hepatology, vol. 36, pp. 12–19, 2021.
9. M. Mascarenhas et al., “Artificial intelligence and capsule endoscopy: Unravelling the future,” Annals of Gastroenterology, vol. 34, pp. 300–309, 2021.
10. T. Aoki et al., “Automatic detection of various abnormalities in capsule endoscopy videos by a deep learning-based system: A multicenter study,” Gastrointestinal Endoscopy, vol. 93, pp. 165–173.e1, 2021.
11. H. Saito et al., “Automatic detection and classification of protruding lesions in wireless capsule endoscopy images based on a deep convolutional neural network,” Gastrointestinal Endoscopy, vol. 92, pp. 144–151.e1, 2020.
12. A. Caroppo, A. Leone, and P. Siciliano, “Deep transfer learning approaches for bleeding detection in endoscopy images,” Computerized Medical Imaging and Graphics, vol. 88, art. 101852, 2021.
13. T. Ghosh, S. A. Fattah, and K. A. Wahid, “Deep transfer learning for automated intestinal bleeding detection in capsule endoscopy imaging,” Journal of Digital Imaging, vol. 34, pp. 404–417, 2021.
14. T. Ribeiro et al., “Automatic detection of vascular lesions using a convolutional neural network in capsule endoscopy,” Endoscopy International Open, vol. 9, pp. E1264–E1272, 2021.
15. P. M. Vieira et al., “Multi-pathology detection and lesion localization in WCE videos by using the instance segmentation approach,” Artificial Intelligence in Medicine, vol. 119, art. 102141, 2021.
16. S. Jain et al., “A deep CNN model for anomaly detection and localization in wireless capsule endoscopy images,” Computers in Biology and Medicine, vol. 137, art. 104789, 2021.
17. S. Jain et al., “Detection of abnormality in wireless capsule endoscopy images using fractal features,” Computers in Biology and Medicine, vol. 127, art. 104094, 2020.
18. X. Xie et al., “Development and validation of an artificial intelligence model for small bowel capsule endoscopy video review,” JAMA Network Open, vol. 5, no. 7, art. e2221992, 2022.
19. H. Morera et al., “Reduction of video capsule endoscopy reading times using deep learning with small data,” Algorithms, vol. 15, art. 339, 2022, doi: 10.3390/a15100339.
20. N. Hosoe et al., “Development of a deep-learning algorithm for small bowel-lesion detection and a study of improvement in the false-positive rate,” Journal of Clinical Medicine, vol. 11, art. 3682, 2022.
21. G. Son et al., “Small bowel detection for wireless capsule endoscopy using convolutional neural networks with temporal filtering,” Diagnostics, vol. 12, art. 1858, 2022.
22. P. Muruganantham and S. M. Balakrishnan, “Attention-aware deep learning model for wireless capsule endoscopy lesion classification and localization,” Journal of Medical and Biological Engineering, vol. 42, pp. 157–168, 2022.
23. J. Afonso et al., “Deep learning for automatic identification and characterization of the bleeding potential of enteric protruding lesions in capsule endoscopy,” GE Portuguese Journal of Gastroenterology, 2022.
24. M. Mascarenhas et al., “Deep learning and capsule endoscopy: Automatic multi-brand recognition of vascular lesions,” United European Gastroenterology Journal, vol. 12, 2024.
25. T. Rahim, M. A. Usman, and S. Y. Shin, “A survey on contemporary computer-aided tumor, polyp, and ulcer detection methods in wireless capsule endoscopy imaging,” Computerized Medical Imaging and Graphics, vol. 85, art. 101767, 2020.
26. O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in MICCAI, 2015, pp. 234–241.
27. Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, “UNet++: A nested U-Net architecture for medical image segmentation,” in Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, 2018, pp. 3–11.
28. O. Oktay et al., “Attention U-Net: Learning where to look for the pancreas,” arXiv:1804.03999, 2018.
29. J. Chen et al., “TransUNet: Transformers make strong encoders for medical image segmentation,” arXiv:2102.04306, 2021.
30. Y. Zhang, H. Liu, and Q. Hu, “TransFuse: Fusing Transformers and CNNs for medical image segmentation,” in MICCAI, 2021, pp. 14–24.
31. H. Cao et al., “Swin-Unet: Unet-like pure Transformer for medical image segmentation,” in ECCV Workshops, 2022, pp. 205–218.
32. A. Hatamizadeh et al., “UNETR: Transformers for 3D medical image segmentation,” in Proc. IEEE/CVF WACV, 2022, pp. 574–584.
33. Y. Tang et al., “Self-supervised pre-training of Swin Transformers for 3D medical image analysis,” in Proc. IEEE/CVF CVPR, 2022, pp. 20730–20740.
34. Y. Gao, M. Zhou, and D. Metaxas, “UTNet: A hybrid Transformer architecture for medical image segmentation,” in MICCAI, 2021, pp. 61–71.
35. E. Xie et al., “SegFormer: Simple and efficient design for semantic segmentation with Transformers,” in Advances in Neural Information Processing Systems, vol. 34, 2021, pp. 12077–12090.
36. Z. Liu et al., “Swin Transformer: Hierarchical vision Transformer using shifted windows,” in Proc. IEEE/CVF ICCV, 2021, pp. 10012–10022.

37. A. Dosovitskiy et al., “An image is worth 16×16 words: Transformers for image recognition at scale,” in Proc. ICLR, 2021.
38. Z. Liu et al., “A ConvNet for the 2020s,” in Proc. IEEE/CVF CVPR, 2022, pp. 11976–11986.
39. F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen, and K. H. Maier-Hein, “nnU-Net: A self-configuring method for deep learning-based biomedical image segmentation,” Nature Methods, vol. 18, pp. 203–211, 2021, doi: 10.1038/s41592-020-01008-z.
40. D. Jha et al., “Kvasir-SEG: A segmented polyp dataset,” in MultiMediaModeling, 2020, pp. 451–462, doi: 10.1007/978-3-030-37734-2_37.
41. D.-P. Fan et al., “PraNet: Parallel reverse attention network for polyp segmentation,” in MICCAI, 2020, pp. 263–273.
42. S. Wang et al., “Annotation-efficient deep learning for automatic medical image segmentation,” Nature Communications, vol. 12, art. 5915, 2021, doi: 10.1038/s41467-021-26216-9.
43. J. Ma et al., “Segment anything in medical images,” Nature Communications, vol. 15, art. 654, 2024, doi: 10.1038/s41467-024-44824-z.
44. T.-Y. Lin et al., “Focal loss for dense object detection,” in Proc. IEEE ICCV, 2017, pp. 2980–2988.
45. F. Milletari, N. Navab, and S.-A. Ahmadi, “V-Net: Fully convolutional neural networks for volumetric medical image segmentation,” in Proc. 3DV, 2016, pp. 565–571.
46. H. Kervadec et al., “Boundary loss for highly unbalanced segmentation,” in Proc. MIDL, 2019, pp. 285–296.
47. R. R. Selvaraju et al., “Grad-CAM: Visual explanations from deep networks via gradient-based localization,” in Proc. IEEE ICCV, 2017, pp. 618–626.
48. R. Strudel et al., “Segmenter: Transformer for semantic segmentation,” in Proc. IEEE/CVF ICCV, 2021, pp. 7262–7272.
49. A. Lin et al., “DS-TransUNet: Dual Swin Transformer U-Net for medical image segmentation,” arXiv:2106.06716, 2021.
50. G. Xu, X. Wu, X. Zhang, and X. He, “LeViT-UNet: Make faster encoders with Transformer for medical image segmentation,” arXiv:2107.08623, 2021.
51. A. Tragakis, C. Kaul, R. Murray-Smith, and D. Husmeier, “The fully convolutional Transformer for medical image segmentation,” in Proc. IEEE/CVF WACV, 2023.
52. Q. Cao et al., “Robotic wireless capsule endoscopy: Recent advances and future directions,” Nature Communications, vol. 15, art. 4893, 2024, doi: 10.1038/s41467-024-49019-0.

APPENDIX A. ALGORITHM 1: TRAINING PROCEDURE FOR BLEEDTRANS-NET

Input: grouped training images X, masks Y, validation set V, maximum epochs E, patience P

Output: best checkpoint θ* and experiment logs

1: audit files, masks, duplicates, and source groups; freeze the test manifest

2: initialize CNN encoder from declared pretrained weights; initialize Transformer, decoder, and boundary head

3: for epoch = 1 to E do

4: for each mini-batch (x, y) sampled from grouped training data do

5: apply paired augmentation to x and y; normalize x

6: extract multi-scale CNN features; tokenize the deepest feature map

7: apply multi-head self-attention and MLP blocks

8: fuse Transformer context with gated CNN skip features

9: predict segmentation logits and auxiliary boundary logits

10: compute BCE, Dice, focal, and boundary losses

11: backpropagate with mixed precision; clip gradients; update AdamW parameters

12: end for

13: evaluate V without augmentation; write per-image metrics and learning-rate logs

14: if validation Dice improves then save θ* and reset patience counter

15: else increment patience counter; stop when counter = P

16: end for

17: lock θ*, select threshold using validation data only, and evaluate the sealed test set once

18: export metrics, bootstrap intervals, plots, explanations, and checkpoint hash

APPENDIX B. COMPACT PYTORCH REFERENCE IMPLEMENTATION

The following single-file scaffold remains an implementation template. It was not executed to obtain the hypothetical numerical results reported above. Dataset paths, true split manifests, full evaluation functions, and empirical hardware logging must be configured before real training.

```
 # bleedtrans_reference.py
from pathlib import Path
import random, json, time
 import numpy as np
import cv2
import torch
import torch.nn as nn
import torch.nn.functional as F
 from torch.utils.data import Dataset, DataLoader
import albumentations as A
from albumentations.pytorch import ToTensorV2
from torchvision.models import convnext_tiny, ConvNeXt_Tiny_Weights
 SEED = 17
def seed_everything(seed=SEED):
 random.seed(seed); np.random.seed(seed); torch.manual_seed(seed)
torch.cuda.manual_seed_all(seed)
torch.backends.cudnn.deterministic = True
torch.backends.cudnn.benchmark = False
class WCEDataset(Dataset):
def __init__(self, pairs, train=False, size=352):
self.pairs = pairs
if train:
 self.tf = A.Compose([
 A.Resize(size, size), A.HorizontalFlip(p=.5), A.VerticalFlip(p=.2),
A.Rotate(limit=20, border_mode=cv2.BORDER_REFLECT_101, p=.5),
A.RandomBrightnessContrast(.12, .12, p=.4), A.RandomGamma(p=.2),
A.GaussNoise(p=.15), A.MotionBlur(blur_limit=5, p=.1),
 A.Normalize(), ToTensorV2()], additional_targets={'mask':'mask'})
else:
 self.tf = A.Compose([A.Resize(size,size), A.Normalize(), ToTensorV2()],
additional_targets={'mask':'mask'})
def __len__(self): return len(self.pairs)
def __getitem__(self, i):
 ip, mp = self.pairs[i]
 image = cv2.cvtColor(cv2.imread(str(ip)), cv2.COLOR_BGR2RGB)
```

```
mask = cv2.imread(str(mp), cv2.IMREAD_GRAYSCALE)
 if image is None or mask is None: raise FileNotFoundError(ip, mp)
 mask = (mask > 127).astype('float32')
 out = self.tf(image=image, mask=mask)
 return out['image'].float(), out['mask'].unsqueeze(0).float(), str(ip)
 class TransformerBlock(nn.Module):
def __init__(self, dim=768, heads=8, mlp_ratio=4, drop=.1):
super().__init__(); self.n1=nn.LayerNorm(dim)
self.attn=nn.MultiheadAttention(dim, heads, dropout=drop, batch_first=True)
self.n2=nn.LayerNorm(dim)
 self.mlp=nn.Sequential(nn.Linear(dim,dim*mlp_ratio),nn.GELU(),nn.Dropout(drop),
 nn.Linear(dim*mlp_ratio,dim),nn.Dropout(drop))
def forward(self,x):
q=self.n1(x); x=x+self.attn(q,q,q,need_weights=False)[0]
 return x+self.mlp(self.n2(x))
class ConvBlock(nn.Module):
def __init__(self, ci, co):
 super().__init__(); self.b=nn.Sequential(nn.Conv2d(ci,co,3,1,1,bias=False),
nn.BatchNorm2d(co),nn.GELU(),nn.Conv2d(co,co,3,1,1,bias=False),nn.BatchNorm2d(co),nn.GELU())
def forward(self,x): return self.b(x)
class GateFuse(nn.Module):
 def __init__(self, cskip, cdec, cout):
 super().__init__(); self.s=nn.Conv2d(cskip,cout,1); self.d=nn.Conv2d(cdec,cout,1)
self.g=nn.Sequential(nn.Conv2d(cout*2,cout,1),nn.Sigmoid()); self.ref=ConvBlock(cout*2,cout)
def forward(self,dec,skip):
 dec=F.interpolate(dec,size=skip.shape[-2:],mode='bilinear',align_corners=False)
s,d=self.s(skip),self.d(dec); a=self.g(torch.cat([s,d],1))
return self.ref(torch.cat([a*s,(1-a)*d],1))
class BleedTransNet(nn.Module):
def __init__(self):
 super().__init__(); b=convnext_tiny(weights=ConvNeXt_Tiny_Weights.DEFAULT)
self.stem=b.features[0]; self.e1=b.features[1]; self.down1=b.features[2]
self.e2=b.features[3]; self.down2=b.features[4]; self.e3=b.features[5]
self.down3=b.features[6]; self.e4=b.features[7]
self.proj=nn.Conv2d(768,384,1); self.pos=nn.Parameter(torch.zeros(1,121,384))
self.tr=nn.Sequential(*[TransformerBlock(384,8) for _ in range(4)])
self.back=nn.Conv2d(384,384,1)
self.f3=GateFuse(384,384,256); self.f2=GateFuse(192,256,128); self.f1=GateFuse(96,128,64)
self.out=nn.Sequential(nn.Conv2d(64,32,3,1,1),nn.GELU(),nn.Conv2d(32,1,1))
self.edge=nn.Conv2d(64,1,1)
```

```
def forward(self,x):
x0=self.stem(x); s1=self.e1(x0); s2=self.e2(self.down1(s1)); s3=self.e3(self.down2(s2)); z=self.e4(self.down3(s3))
z=self.proj(z); b,c,h,w=z.shape; t=z.flatten(2).transpose(1,2)
pos=F.interpolate(self.pos.transpose(1,2).reshape(1,384,11,11),size=(h,w),mode='bilinear').flatten(2).transpose(1,2)
t=self.tr(t+pos); z=self.back(t.transpose(1,2).reshape(b,384,h,w))
d3=self.f3(z,s3); d2=self.f2(d3,s2); d1=self.f1(d2,s1)
logits=F.interpolate(self.out(d1),size=x.shape[-2:],mode='bilinear',align_corners=False)
edges=F.interpolate(self.edge(d1),size=x.shape[-2:],mode='bilinear',align_corners=False)
return logits, edges
def dice_loss(logits,y,eps=1e-6):
p=torch.sigmoid(logits); num=2*(p*y).sum((2,3))+eps; den=(p+y).sum((2,3))+eps
return 1-(num/den).mean()
def focal_loss(logits,y,gamma=2):
bce=F.binary_cross_entropy_with_logits(logits,y,reduction='none'); pt=torch.exp(-bce)
return ((1-pt)**gamma*bce).mean()
def boundary_target(y):
dil=F.max_pool2d(y,3,1,1); ero=-F.max_pool2d(-y,3,1,1)
return (dil-ero).clamp(0,1)
def total_loss(logits,edges,y):
return (.20*F.binary_cross_entropy_with_logits(logits,y)+.40*dice_loss(logits,y)+
.25*focal_loss(logits,y)+.15*F.binary_cross_entropy_with_logits(edges,boundary_target(y)))
@torch.no_grad()
def metrics(logits,y,thr=.5,eps=1e-7):
 p=(torch.sigmoid(logits)>=thr).float(); dims=(1,2,3)
tp=(p*y).sum(dims); fp=(p*(1-y)).sum(dims); fn=((1-p)*y).sum(dims); tn=((1-p)*(1-y)).sum(dims)
dice=(2*tp+eps)/(2*tp+fp+fn+eps); iou=(tp+eps)/(tp+fp+fn+eps)
precision=(tp+eps)/(tp+fp+eps); recall=(tp+eps)/(tp+fn+eps); specificity=(tn+eps)/(tn+fp+eps)
return {k:v.cpu().numpy() for k,v in {'dice':dice,'iou':iou,'precision':precision,
'recall':recall,'specificity':specificity}.items()}
def run_epoch(model,loader,opt,device,train=True):
model.train(train); scaler=torch.cuda.amp.GradScaler(enabled=device.type=='cuda'); losses=[]
for x,y,_ in loader:
x,y=x.to(device),y.to(device); opt.zero_grad(set_to_none=True)
with torch.cuda.amp.autocast(enabled=device.type=='cuda'):
logits,edges=model(x); loss=total_loss(logits,edges,y)
if train:
scaler.scale(loss).backward(); scaler.unscale_(opt); nn.utils.clip_grad_norm_(model.parameters(),1.0)
scaler.step(opt); scaler.update()
losses.append(float(loss.detach()))
return float(np.mean(losses))
```

```
if __name__=='__main__':
seed_everything(); device=torch.device('cuda' if torch.cuda.is_available() else 'cpu')
# pairs must be created from a frozen, duplicate-grouped split manifest.
train_pairs=[]; val_pairs=[]
if not train_pairs: raise RuntimeError('Populate train_pairs/val_pairs from the split manifest before training.')
 tr=DataLoader(WCEDataset(train_pairs,True),batch_size=8,shuffle=True,num_workers=4,pin_memory=True)
va=DataLoader(WCEDataset(val_pairs,False),batch_size=8,shuffle=False,num_workers=4)
model=BleedTransNet().to(device); opt=torch.optim.AdamW(model.parameters(),lr=1e-4,weight_decay=1e-5)
best=-1; patience=0
for epoch in range(1,121):
tl=run_epoch(model,tr,opt,device,True); vl=run_epoch(model,va,opt,device,False)
 print(json.dumps({'epoch':epoch,'train_loss':tl,'val_loss':vl}))
# Add full validation Dice aggregation and checkpointing here; never tune on the test set.
```