Neurosurgical Phase Recognition for Aneurysm Clipping via Fine-Tuned CNN and Temporal Models

Emily Oberleitner
Stanford University
eober@stanford.edu

Veronica Wang
Stanford University
vwang2@stanford.edu

Nicole Wong
Stanford University
nicolew6@stanford.org

Abstract

Surgical phase recognition is a crucial part of surgical workflow analysis, but most prior work has focused on laparoscopic or robotic abdominal procedures rather than neurosurgery. In this paper, we implement automatic phase recognition for aneurysm neurosurgery videos in which visually similar tissue structures, subtle motions, and ambiguous phase boundaries make classification challenging.

This task is approached as a supervised classification problem over surgical video frames and temporal frame sequences. We construct a dataset of 48 aneurysm clipping videos sampled at 1fps and annotated into four surgical phases and compare frame-based and temporal architectures under both frame-level and video-level validation protocols. Using these, we compare a frame-wise ResNet-50 baseline, a CNN+LSTM temporal baseline, MS-TCN, and an OperA-based self-attention transformer model. We evaluate these models using accuracy, edit distance, F1 score, and boundary-aware accuracy. We test post-processing with Viterbi. We also compare frame-level and video-level validation splits to understand how our models are generalizing.

Our results show that our data-splitting methodology and the quality of the CNN features strongly affected performance. The ResNet-50 baseline achieves high accuracy but has poor segment-level recognition due to class imbalance, while temporal models better capture phases and phase boundaries. Fine-tuning the ResNet-50 backbone on surgical frames greatly improves temporal model performance, and the OperA transformer with Viterbi achieves the strongest results. Future work should focus on accurately detecting clipping phase boundaries and reducing overfitting due to limited dataset size.

1. Introduction

Neurosurgical procedures require an exceptional degree of precision, real-time spatial awareness, and cognitive endurance due to the delicate nature of eloquent brain structures and the high stakes of intraoperative complications. In the modern digital operating room (OR), computer-vision-assisted tools and surgical data science have emerged as foundational technologies to improve patient outcomes and optimize workflows [14]. A core component of these intelligent systems is automatic surgical phase recognition, which enables online context-aware assistance, safety auditing, and efficient post-operative video summarization for residency training [7].

Transitioning automated workflow analysis from laparoscopic surgeries to neurosurgery presents profound computer vision challenges. In laparoscopic surgeries, there are distinct anatomical landmarks and highly contrasted multi-organ environments typical of abdominal procedures, while the neurosurgical field (viewed through an operative microscope) presents additional challenges through its extreme visual homogeneity, where specialized brain tissues often appear as a continuous landscape of structurally similar features. Furthermore, real-world neurosurgical video streams are subject to significant operational artifacts: rapid microscope focus and position adjustments, periodic saline irrigation, lens obstructions, and unpredictable debris like blood or gauze. These factors induce significant visual noise, causing standard frame-wise convolutional neural networks (CNNs) to yield unstable and erroneous feature representations.

To parse long-range temporal dependencies in surgical videos, traditional architectures have relied on recurrent networks such as Long Short-Term Memory (LSTM) models [10] or multi-stage Temporal Convolutional Networks (TCNs) [1]. While these methods attempt to model sequential context, LSTMs struggle with memory decay and vanishing gradients across long-duration procedures, which in complex neurosurgery can span several hours. Conversely, while TCNs process sequences efficiently via dilated convolutions, they lack the multi-head self-attention mechanisms capable of dynamically weighting non-adjacent frame relationships over extended timelines.

In this work, we present an adaptation of the *OperA*

(Attention-Regularized Transformer) framework [4] designed specifically to manage the challenges of real-world neurosurgical data. In our project, the input is a sequence of frames from an aneurysm neurosurgery video labeled as one of 4 phases. A CNN backbone converts each frame into a learned feature vector. The sequence of feature vectors is passed into a transformer with self-attention. The output is a predicted surgical phase label for each frame or time step. Tracking these distinct phases is clinically essential; each phase represents a sharp transition in surgical risk and objective, from the macro-level navigation of initial craniotomy and tissue retraction to the micro-vascular manipulation required for proximal control and ultimate clip deployment. The core architecture decouples spatial feature extraction from global sequence modeling by passing frame embeddings to an 2-layer causally masked transformer network. To mitigate the impact of intraoperative visual noise across these delicate transitions—such as sudden bleeding or microscope adjustments during critical exposure phases—we utilize an attention regularization loss (L_{reg}) that penalizes the transformer if it assigns high attention weights to frames where the underlying spatial backbone exhibits low confidence or high prediction error. By anchoring timeline decisions strictly to high-quality landmarks and utilizing causal masking to prevent future information leakage, the model remains robust against technical artifacts while executing stable, real-time online phase tracking.

Our main contributions are: (1) formulating aneurysm neurosurgery phase recognition as a frame-level and temporal sequence-level classification problem, (2) implementing and evaluating a ResNet-50 frame-wise baseline, (3) implementing and evaluating a CNN+LSTM temporal baseline, (4) adapting and evaluating an OperA-style transformer model for aneurysm neurosurgery videos, and (5) analyzing the strengths and limitations of frame-wise and temporal models.

2. Related Works

Surgical phase recognition has been a widely studied task in surgical workflow analysis, largely due to its applications in surgical training, intraoperative assistance, and autonomous surgical tools. For instance, Maier-Hein et al. describe surgical data science as a framework for using computational methods to understand and improve surgical care [14]. They illustrate the importance and power of machine learning systems in designing the most effective algorithms for classic problems in computer vision and medical imaging. This supports the point from Garrow et al., who provide a systematic review of early machine learning approaches for surgical phase recognition, including methods such as Hidden Markov Models and Artificial Neural Networks, with a trend towards higher model complexity over time [7].

Similar to OperA's application in laparoscopic surgeries, earlier approaches such as EndoNet used convolutional neural networks instead to recognize surgical phases and tool presence from surgical video frames [20]. Frame-wise CNNs are useful for learning the visual features that correspond to a phase, but they do not model the temporal context across a procedure.

To address this limitation, later works incorporated temporal models. For instance, Twinanda et al.'s EndoLSTM built on EndoNet by adding a temporal component that incorporates information from neighboring frames [19]. This improved reported performance from 81.7% to 88.6% accuracy and from 76.5% to 84.5% F1 score, supporting the idea that surgical phase recognition benefits from temporal context rather than frame-wise classification alone. Temporal convolutional approaches such as TeCNO also improved phase recognition by applying multi-stage temporal convolutional networks to CNN frame features [3], and SV-RCNet used recurrent convolutional networks to optimize workflow recognition from surgical videos [12]. Recently, Konnai et al. applied TeCNO to robot-assisted radical prostatectomy and performed cross-surgeon validation, showing that performance can decrease when models are tested on procedures from different surgeons. This emphasizes the importance of evaluating generalization across surgical styles and motivated our decision to include videos from different surgeons in our dataset [13].

Transformer-based models have been more recently introduced as another approach to temporal surgical phase classification. Vaswani et al. introduced the transformer architecture, which uses self-attention to model relationships across a sequence [21]. This formed the basis for the OperA model we implemented, which adapts this idea to surgical phase recognition by passing ResNet-50 frame features to a transformer and adding attention regularization to encourage the model to focus on more phase-defining frames [4]. Since our project also uses ResNet-50 as a frame-wise baseline, we cite He et al. for the ResNet architecture [9]. Our project applies ideas from prior work on laparoscopic and robotic abdominal procedures to aneurysm neurosurgery, where visually similar tissue structures, subtle motions, and miscellaneous surgical tools and objects make phase recognition especially challenging.

3. Methods

We present aneurysm neurosurgery phase recognition as a classification problem over surgical videos. Each video is represented as an ordered sequence of raw frames,

$$V = \{v_1, v_2, \ldots, v_T\},$$

where v_t is the video frame at time step t. For each frame or time step, the goal is to predict a surgical phase label

$$\hat{y}_t \in \{1, 2, \ldots, C\},$$

where C is the number of aneurysm neurosurgery phases in our dataset, including brain exposure, parent vessel identification, neck identification, and clipping. Each model outputs a probability distribution on the phase classes and the predicted phase is selected as the class with the highest probability.

3.1. Feature Extraction

We use ResNet-50 [9] pre-trained on ImageNet [5] as our CNN feature extractor.

We evaluate two feature regimes as an ablation. In the frozen regime, ResNet-50 weights are kept fixed at their ImageNet-pretrained values, so features encode generic visual structure rather than surgical phase-specific semantics. In the fine-tuned regime, the final classification head is replaced with a C-class linear layer and the network is trained on labeled surgical frames before feature extraction. Temporal models are then trained on these extracted features.

The fine-tuned ResNet was trained on 10 epochs using AdamW with a differential learning rate: $\eta = 10^{-4}$ for the classification head and $\eta = 10^{-5}$ for the convolutional layers, with batch size 32. A lower learning rate for the backbone prevents overwriting pretrained low-level features while allowing the higher-level representations to adapt to the surgical domain. After fine-tuning, the backbone weights are frozen and features are re-extracted for all videos. The temporal models are then trained on these fixed features rather than end-to-end.

3.2. Frame-wise ResNet-50 Baseline

Our first baseline is a frame-wise ResNet-50 classifier. The input to this model is a single raw surgical video frame v_t, and the output is a predicted surgical phase label $\hat{y}_t$. The final fully connected layer of ResNet-50 is replaced with a classification layer that outputs C phase logits.

This baseline tests how much phase information can be learned from visual features alone. Since it classifies each frame independently, it does not use temporal information from neighboring frames or the surgical workflow.

3.3. CNN+LSTM Temporal Baseline

Our second baseline is a CNN+LSTM model. The CNN first converts each raw frame v_t into a learned feature vector $\mathbf{x}_t$. The LSTM then processes these frame features in temporal order, allowing the model to use context from previous frames when predicting the current surgical phase.

This baseline tests whether adding temporal context improves phase classification compared with frame-wise classification alone. It is a useful comparison because aneurysm surgical phases typically occur in a sequence, and the correct label for a frame may depend on what happened immediately before it.

3.4. MS-TCN

In literature, this model is often used as the standard for surgical video analysis. This architecture relies on dilated causal convolutions and processes sequences layer-by-layer, where each stage refines the predictions of the previous one. It is highly inductive to local, short-term trends and scales linearly with sequence length which makes them highly memory-efficient and excellent for real-time video analysis.

Our MS-TCN follows Farha and Gall [6] with two stages and eight dilated residual layers per stage. Each layer uses dilation 2^i for layer $i \in \{0, \ldots, 7\}$, giving a receptive field of $2^9 - 1 = 511$ frames. The first stage operates on raw features; subsequent stages refine predictions by taking the softmax output of the previous stage as input. The multi-stage loss is:

$$\mathcal{L}_{\text{MS-TCN}} = \sum_{s=1}^{S} \mathcal{L}_{\text{CE}}(\hat{Y}^{(s)}, Y)$$

where $\hat{Y}^{(s)}$ are the logits at stage s and Y are the ground-truth labels. We use 128 filters per layer and dropout $p = 0.5$.

3.5. OperA Transformer Model

Our final model is based on OperA, an attention-regularized transformer model for surgical phase recognition. Like the CNN+LSTM baseline, OperA first uses a CNN to extract features from each frame. However, instead of using an LSTM to process the sequence, OperA uses a transformer with self-attention.

Self-attention allows the model to compare frames across the sequence and learn temporal patterns in the surgery. This is important because a single frame may contain visually ambiguous features, while earlier frames can provide more context about the current phase.

We implement the transformer with 2 layers, 4 attention heads, model dimension $d = 256$, and dropout $p = 0.5$. Features are projected from 2048 to 256 dimensions before the attention layers. Per-frame logits are produced by a linear head applied to the output of the final attention layer.

3.6. Causal Masking

OperA uses causal masking so that the model cannot attend to future frames when predicting the current phase. This means that the prediction at time t can only use frames from time steps 1 through t. This ensures that the model is suitable for real-time phase recognition, where future video frames would not be available.

$$\text{Mask}_{ij} = \begin{cases} 0 & \text{if } i \geq j \\ -\infty & \text{otherwise} \end{cases}$$

3.7. Attention Regularization

OperA also includes an attention regularization term. Surgical videos often contain noisy or uninformative frames due to motion, blur, bleeding, obstruction, or camera adjustments. Thus, attention regularization encourages the transformer to focus on more phase-defining frames.

The idea is that if the CNN backbone has high prediction error on a frame, that frame may be less reliable, so OperA penalizes the transformer when it assigns high attention to frames with high CNN error. The total loss is

$$L = L_{CE} + \lambda L_{reg},$$

where L_{CE} is the cross-entropy classification loss, L_{reg} is the attention regularization loss, and λ adjusts the strength of the regularization term. $\mathcal{L}_{\text{reg}} \in \{\mathcal{L}_{\text{ent}}, \mathcal{L}_{\text{smooth}}\}$ depending on the chosen variant, and $\epsilon = 10^{-8}$ for numerical stability. λ = 0.01 for both variants.

The cross-entropy loss is

$$L_{CE} = -\frac{1}{N} \sum_{i=1}^{N} \log p_{i,y_i},$$

where N is the number of training examples, y_i is the ground-truth phase label, and p_{i,y_i} is the predicted probability assigned to the correct class.

This loss penalizes the model when it assigns low probability to the correct surgical phase.

However, in our study attention regularization with $\lambda = 0.01$ reduced accuracy (entropy: 51.9%, smoothness: 41.3%) compared to the unregularized baseline (53.3%), so it was not used in the fine-tuned experiments.

3.8. Regularization and Optimization

All models are optimized with AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay 10^{-3}). We use a linear warmup for 5 epochs followed by cosine annealing:

$$\eta_t = \eta_{\text{min}} + \frac{1}{2}(\eta_{\text{max}} - \eta_{\text{min}}) \left(1 + \cos\left(\frac{\pi(t - t_{\text{warm}})}{T - t_{\text{warm}}}\right)\right)$$

with $\eta_{\text{max}} = 5 \times 10^{-4}$ and $\eta_{\text{min}} = 10^{-6}$. Gradient norms are clipped to 1.0. During training, Gaussian noise ($\sigma = 0.1$) is added to input features as data augmentation.

Models were trained for 100 epochs and the checkpoint with lowest validation loss was selected.

3.9. Post-Processing

Viterbi Decoding Given per-frame softmax probabilities $P \in \mathbb{R}^{T \times C}$, Viterbi decoding finds the most likely label sequence under a first-order Markov model:

$$\hat{Y}^* = \arg\max_{Y} \prod_{t=1}^{T} P(y_t|\mathbf{x}_t) \cdot P(y_t|y_{t-1})$$

We compare two transition matrices: (1) a *default surgical prior* encoding forward-only transitions with $P(\text{stay}) = 0.95$, $P(\text{advance}) = 0.04$, $P(\text{skip}) = 0.009$, $P(\text{backward}) = 0.001$; and (2) a *learned* transition matrix estimated from empirical frame-to-frame transition counts in the training set. The learned matrix outperforms the prior across all thresholds.

Temporal Smoothing As a lightweight alternative, we apply a mean filter of window size 15 over per-frame class probabilities and take the argmax. This suppresses brief flickering at minimal computational cost.

3.10. Tolerance Window

The dataset was difficult to label because phase changes are often ambiguous, and annotators sometimes needed the temporal context of the full video to determine where one phase ended and another began. Due to this, we added a tolerance window of 5 seconds (5 frames) around phase boundaries, excluding frames within this region from accuracy computation to avoid penalizing predictions near inherently ambiguous transitions. 5 frames is a small enough window to exclude only the genuinely ambiguous boundary region while still evaluating the majority of each phase.

3.11. Implementation Details and Codebase

Because the original OperA code was not publicly available, we reimplemented the model ourselves based on the architecture and methodology described in the OperA paper. Our implementation includes the data loading pipeline, frame preprocessing, ResNet-50 frame-wise baseline, MS-TCN, CNN+LSTM temporal baseline, transformer-based OperA model, training loop, loss computation, and evaluation scripts.

For the OperA implementation, we reproduced the main components described in the paper: CNN-based frame feature extraction, transformer self-attention for temporal modeling, causal masking for online prediction, and attention regularization. We adapted the model to our aneurysm neurosurgery dataset by replacing the original cholecystectomy phase labels with our aneurysm surgery phase labels.

We implement all models in PyTorch [16] with torchvision [18] for image preprocessing. Video decoding uses OpenCV [2], data manipulation uses NumPy [8] and Pandas

[15], and figures are generated with Matplotlib [11]. t-SNE visualizations use scikit-learn [17].

4. Dataset and Features

Our dataset consists of 48 videos from aneurysm neurosurgery procedures. Each video was sampled at 1fps, and the frames were manually annotated in CVAT with labels corresponding to the major phases of the operation, being brain exposure, parent vessel identification, aneurysm neck identification, and clipping. The input data for our models comes from these sampled frames, and each frame is paired with a surgical phase label. Table 1 shows the class distribution, which is moderately imbalanced with Clipping being the most common phase (36.8%) and Brain Exposure the least common (18.1%). Not all videos contain all four phases.

Phase	Frames	Proportion
Brain Exposure	7,354	18.1%
Parent Vessel ID	10,222	25.1%
Dome & Neck ID	8,172	20.1%
Clipping	14,977	36.8%
Total	40,725	100%

Table 1. Frame-level class distribution across 48 videos.

4.1. Data Split

In total, our surgical videos range from 1 to 30 minutes in length, with a total of 41,604 frames. We split the data using an 80-20 train-validation split. For our initial experiments, we used a frame-level split, where individual frames were randomly divided into training and validation sets. However, because this can cause data leakage, this split can overestimate accuracy. We tested video-level split for both testing and training but found that there may not be enough data for accurate training. We then considered using **64 frame-level sliding windows for training while the validation was performed on complete video sequences.** This gives us a stronger evaluation method because small inconsistencies such as one specific video not having a phase will not affect validation results as much. This gave us many more training batches (leading to more gradient updates per epoch) and fixed the issue of variable-length sequences, as otherwise we would have to use excessive padding.

4.2. Preprocessing

After downsampling to 1fps, each frame is resized to 224x224 pixels (from 1920x1080) and normalized with ImageNet mean and standard deviation. For the frame-wise ResNet-50 baseline, each sampled frame is treated as one training example. For the temporal models, we used temporally ordered sequences of CNN frame features. We evaluated two feature regimes: (1) frozen ResNet-50 features using unmodified ImageNet weights and (2) fine-tuned ResNet-50 features trained on labeled surgical frames. t-SNE visualization of frozen features showed no meaningful cluster structure across the four surgical phases, motivating fine-tuning.

We did not apply image-level data augmentation in our main experiments because many standard augmentations could distort phase-defining visual features, such as the presence and position of clips, tools, blood vessels, or the aneurysm neck. Our bottleneck also was not lack of frames but lack of phase transitions which augmentation would not fix.

The features used by our models are learned directly from the image data. We do not manually extract features such as HOG, PCA, Fourier features, or optical flow.

5. Experiments/Results/Discussion

5.1. Evaluation Metrics

We evaluate with four complementary metrics:

1. **Frame accuracy**: $\frac{1}{T}\sum_t \mathbf{1}[\hat{y}_t = y_t]$. Standard but dominated by frequent classes.

2. **Edit distance**: $1 - \frac{\text{Levenshtein}(\hat{S},S)}{\max(|\hat{S}|,|S|)}$, where $\hat{S}$ and S are the predicted and ground-truth sequences of *segments* (contiguous runs of the same phase). A score of 1.0 indicates perfect segment-level ordering; 0.0 indicates complete disorder.

3. **Segmental F1@k**: Following [6], a predicted segment is considered a true positive if its IoU with a ground-truth segment of the same class exceeds threshold $k \in \{0.1, 0.25, 0.5\}$. This penalizes both missed phases and temporally misaligned predictions.

4. **Boundary-aware accuracy**: Frame accuracy computed only on frames more than 5 seconds from any phase transition boundary, to measure performance on unambiguous central frames.

5.2. Hyperparameter Selection

Hyperparameters were chosen based on validation performance and established best practices. The learning rate $\eta_{\max} = 5 \times 10^{-4}$ was selected as the highest rate that did not cause training instability (observed as oscillating loss). Warmup over 5 epochs was introduced after observing that models trained without warmup failed to converge from random initialization when large batch losses appeared in the first epoch. Dropout $p = 0.5$ and weight decay 10^{-3} were chosen to address early overfitting: without regularization, all models reached their best validation loss within the first 10 epochs and degraded thereafter. Batch size 4 was constrained by GPU memory (16 GB T4) when processing variable-length sequences up to several hundred frames.

5.3. Importance of Validation Split

We first establish that split methodology significantly affects reported results. As established in Section 4.1, we use 64-frame sliding windows for training and complete held-out video sequences for validation. As shown in Figure 1, the frame-level split yields lower validation loss and higher accuracy than the video-level split because the temporal context from each video leaks into training. All subsequent experiments use video-level splits.

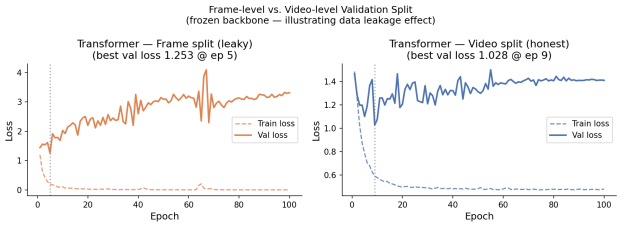

Figure 1. Transformer validation loss under a leaky frame-level split vs. an honest video-level split, both using **frozen** (inaccurate) ImageNet features. The frame-level split yields artificially lower validation loss due to frames from the same video appearing in both train and val sets.

5.4. Frozen vs Fine-Tuned Features

We cannot understate the impact of retraining the CNN encoder on surgical features rather than frozen ImageNet features. Figure 2 reports loss after retraining on a fine-tuned CNN. Comparing this to Figure 1, we see massive improvement.

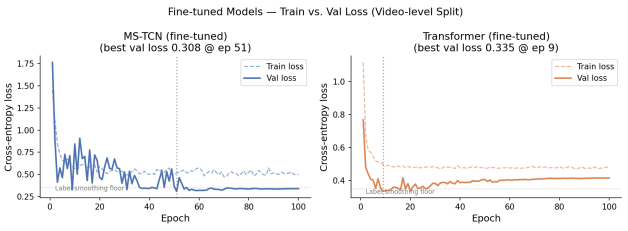

Figure 2. Train and validation loss for MS-TCN and Transformer trained on fine-tuned backbone features under the video-level split. Both models converge near the label smoothing floor of 0.349, indicating the loss function is saturated rather than the models underfitting.)

Note, validation loss lower than train loss is expected here for two reasons. (1) label smoothing is applied only to training loss, making train loss artificially higher since even correct predictions are penalized for being too confident. (2) dropout is active during training but disabled during evaluation, so the val forward pass uses the full network capacity.

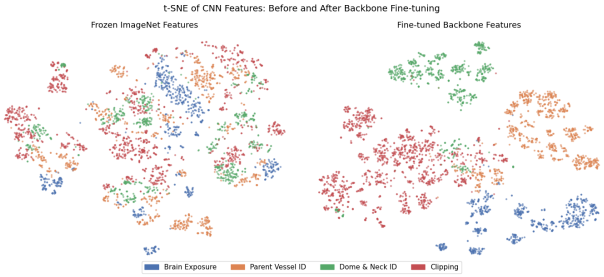

Figure 3. t-SNE visualization of CNN features extracted from 4,000 randomly sampled frames using frozen ImageNet weights (top) and fine-tuned backbone weights (bottom), colored by surgical phase. Frozen features show no meaningful cluster separation, explaining the poor performance of temporal models trained on them. Fine-tuned features form distinct clusters per phase, providing the discriminative signal necessary for accurate phase recognition.

5.5. Main Results

Table 2 summarizes results for all models. We report metrics at the best validation loss epoch, with Viterbi post-processing applied where available. We refer to our causal transformer with attention regularization as the OperA-based transformer model. In Table 2, this model is listed as Transformer + Viterbi.

Model	Feat.	Acc	Edit	F1@10	F1@50
CNN Baseline	Frozen	0.956	—	0.000	0.000
LSTM	Frozen	0.116	0.024	0.018	0.004
Transformer	Frozen	0.533	0.015	0.016	0.004
MS-TCN + Viterbi	Frozen	0.525	0.467	0.553	0.340
MS-TCN + Vit. (learned)	Frozen	0.503	0.583	0.585	0.390
tMS-TCN + Viterbi	Finetune	0.957	0.680	0.809	0.714
Transformer + Vit.	Finetune	**0.936**	**0.810**	**0.895**	**0.789**

Table 2. Main results at best validation loss epoch. Edit distance and F1 use Viterbi decoding where available. "Feat." = feature regime.

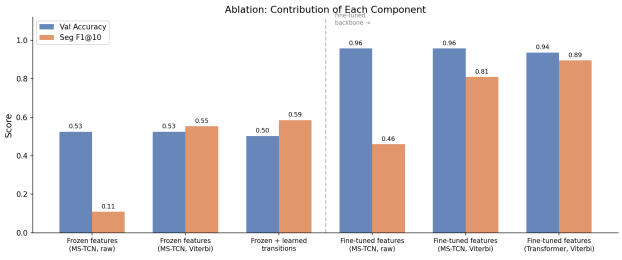

Figure 4. Ablation study showing the contribution of each component to validation accuracy and segmental F1@10. Each bar represents the best validation epoch for that configuration. The largest single gain comes from switching to fine-tuned backbone features (right of the dashed line), outweighing all architectural and post-processing improvements combined.

CNN Baseline As seen in Table 2, the frame-level CNN baseline achieves 95.6% frame accuracy but a segmental F1@10 of 0.000. This reflects the baseline predicting a single dominant class for most videos—frame accuracy is high due to class imbalance but the model produces no meaningful segment-level output. This showcases the inadequacy of frame accuracy as the sole metric for phase recognition.

Temporal Models on Frozen Features LSTM, Transformer, and MS-TCN trained on frozen ImageNet features all achieve 50–53% frame accuracy, near the majority-class baseline. Segmental F1 is near zero for LSTM and Transformer without Viterbi post-processing, as these models produce temporally incoherent predictions. MS-TCN with Viterbi decoding is the exception, reaching F1@10 of 0.553–0.585 even with frozen features, demonstrating MS-TCN's stronger inductive bias for temporal structure.

Impact of Backbone Fine-Tuning Fine-tuning ResNet50 on surgical frames is the single largest performance jump in our experiments. Both MS-TCN and the Transformer improve dramatically, with frame accuracy increasing from ∼53% to 95.7% and 93.6%, and Viterbi F1@10 from 0.553 to 0.809 and 0.895 respectively. This result confirms that frozen ImageNet features lack the discriminative structure needed to distinguish surgical phases.

Viterbi and Temporal Smoothing Figure 5 shows that Viterbi decoding substantially improves segmental F1 in all configurations. For the fine-tuned Transformer, the learned transition matrix achieves a peak Viterbi F1@10 of 0.944. Temporal mean smoothing is competitive with Viterbi on MS-TCN but significantly weaker on the Transformer, suggesting the Transformer's raw probability outputs have higher entropy and benefit from the stronger sequential constraint imposed by Viterbi.

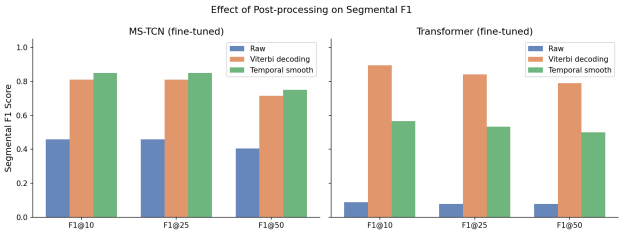

Figure 5. Segmental F1 at three overlap thresholds (10%, 25%, 50%) for raw predictions, Viterbi decoding, and temporal smoothing applied to MS-TCN and the Transformer trained on fine-tuned features. Viterbi decoding consistently improves over raw predictions by enforcing valid phase orderings; temporal smoothing is competitive on MS-TCN but weaker on the Transformer.

5.6. Per-Class Analysis

Figure 6 shows per-class accuracy and F1 for both fine-tuned models. Brain Exposure, Parent Vessel ID, and Dome & Neck ID are all learned well ($\geq$ 0.90 accuracy). Clipping is the exception: while frame-level accuracy is high, segmental F1 for Clipping is disproportionately low (0.134 for MS-TCN, 0.323 for Transformer). This reflects a systematic misalignment between predicted and ground-truth clipping segments: the model correctly identifies frames as clipping but mis-identifies the start and end boundaries. Clipping is visually similar to Dome & Neck ID (the clip is positioned at the neck), and the fine motor clip application can be brief, contributing to poor boundary detection.

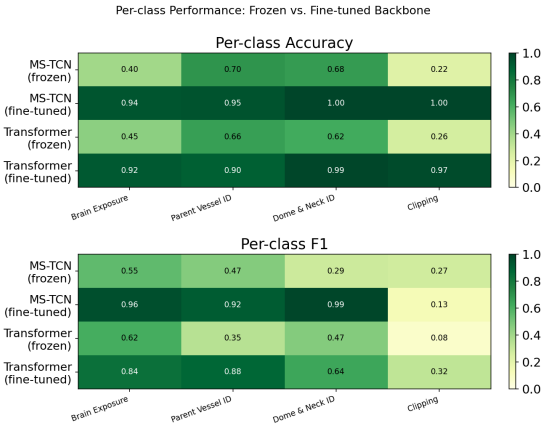

Figure 6. Per-class accuracy (top) and F1 score (bottom) for frozen and fine-tuned models. Fine-tuned features improve all classes substantially. Clipping shows high accuracy but consistently low F1 across all models, indicating the model identifies clipping frames correctly but fails to align predicted segment boundaries with ground truth.

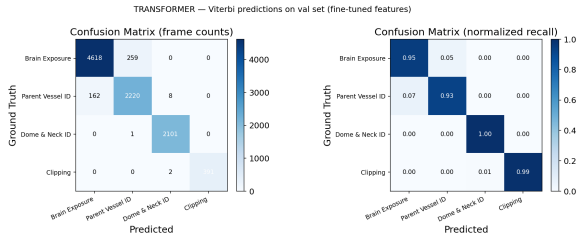

Figure 7. Confusion matrix (normalized by row) for the fine-tuned Transformer using Viterbi decoding on the held-out validation set. Rows are ground truth phases, columns are predicted phases. Diagonal entries show per-class recall. Clipping shows high recall but is frequently predicted during Dome and Neck Identification frames, explaining its low segmental F1.

5.7. Failure Cases

The phase timeline in Figure 8 illustrates common failure modes. The most frequent error is early termination of Clip-

ping—the model reverts to Dome & Neck ID before the surgeon finishes Clipping. As discussed in Section 5.6, Clipping is visually similar to Dome & Neck ID, contributing to the oscillation errors shown here. Viterbi decoding reduces but does not eliminate these errors; the transition matrix assigns low but nonzero probability to backward transitions, allowing the model to revert when evidence is strong.

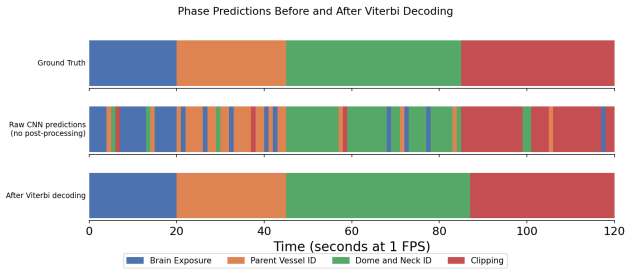

Figure 8. Phase timeline for a 120-second video clip. Ground truth (top) shows the ideal phase sequence. Raw CNN predictions (middle) exhibit temporal flickering — brief, physically implausible phase switches — due to frame-level classification without temporal context. After Viterbi decoding (bottom), the sequence is constrained to forward-only transitions, eliminating flickering and producing a smooth segmentation that closely follows the ground truth.

5.8. Overfitting Analysis

As shown in Figure 9, the Transformer reaches peak segmental F1@10 around epoch 10–20 before overfitting, while MS-TCN peaks later around epoch 50, suggesting MS-TCN is slower to overfit due to its stronger temporal inductive bias. We partially mitigate this through label smoothing, dropout ($p = 0.5$), weight decay (10^{-3}), gradient clipping, and feature noise augmentation. Data augmentation at the feature level (Gaussian noise with $\sigma = 0.1$) was added to improve generalization but its isolated contribution was not ablated. The primary solution would be more labeled videos; 48 videos is small relative to published surgical phase recognition datasets (Cholec80 [20] has 80 videos, and more recent datasets have 200+).

6. Conclusion/Future Work

We presented a two-stage pipeline for surgical phase recognition in aneurysm clipping, combining a fine-tuned ResNet50 feature extractor with MS-TCN and causal transformer temporal models, and Viterbi post-processing. Our main finding is that task-specific feature learning is the dominant factor: fine-tuning the visual backbone increased segmental F1@10 from 0.585 to 0.895—a larger gain than any architectural choice among temporal models. The causal transformer with a learned Viterbi transition matrix achieves the best overall performance (F1@10 = 0.944, F1@50 = 0.789), competitive with state-of-the-art systems

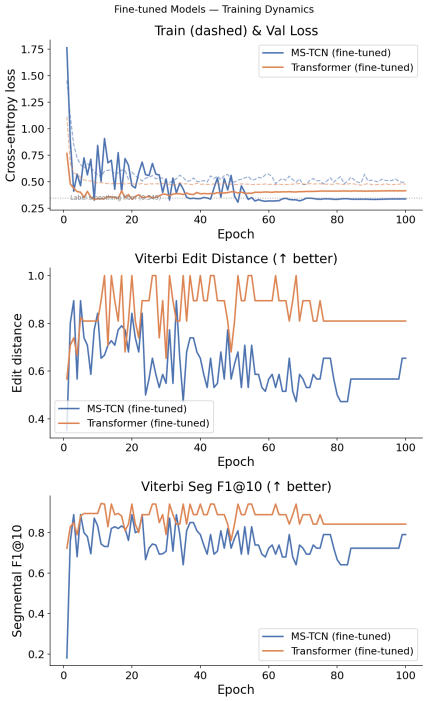

Figure 9. Training dynamics for fine-tuned MS-TCN and Transformer. Top: train (dashed) and validation loss, with the label smoothing floor marked. Middle: Viterbi edit distance over epochs. Bottom: Viterbi segmental F1@10 over epochs. Both models peak within the first 10–50 epochs before overfitting, with the Transformer achieving higher segmental F1 despite similar loss values.

on larger surgical datasets.

Clipping phase segmentation remains an open challenge, with segmental F1 substantially below the other phases due to short duration, visual similarity to the preceding phase, and subtle boundary cues. Future work should explore boundary-aware loss functions that explicitly penalize misaligned segment endpoints, or incorporate tool detection as an auxiliary supervision signal (clip applier presence is a reliable indicator of the Clipping phase).

Longer term, the primary bottleneck is dataset size. With 48 videos, the model begins overfitting within 10–50 epochs depending on configuration. Pseudo-labeling of unannotated surgical footage, leave-one-out cross-validation across surgeons, and data-sharing across institutions are natural avenues. Online inference is another important direction: the causal transformer already supports it, but latency at 1 FPS has not been characterized on clinical hardware.

References

[1] S. Bai, J. Z. Kolter, and V. Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018. 1

[2] G. Bradski. OpenCV library, 2000. 4

[3] D. Czempiel, M. Paschali, M. Keicher, W. Simson, H. Feussner, S. S. Park, and N. Navab. TeCNO: Surgical phase recognition with multi-stage temporal convolutional networks. In *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2020. 2

[4] D. Czempiel, M. Paschali, D. Ostler, S. S. Park, H. Feussner, and N. Navab. OperA: Attention-regularized transformers for surgical phase recognition. In *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2021. 2

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 3

[6] Y. A. Farha and J. Gall. MS-TCN: Multi-stage temporal convolutional network for action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3, 5

[7] C. R. Garrow, K.-F. Kowalewski, L. Li, M. Wagner, M. W. Schmidt, S. Engelhardt, D. A. Hashimoto, H. G. Kenngott, S. Bodenstedt, S. Speidel, B. P. Müller-Stich, and F. Nickel. Machine learning for surgical phase recognition: A systematic review. *Annals of Surgery*, 273(4):684–693, 2021. 1, 2

[8] C. R. Harris, K. J. Millman, S. J. van der Walt, et al. Array programming with NumPy. *Nature*, 585:357–362, 2020. 4

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 3

[10] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. 1

[11] J. D. Hunter. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. 5

[12] Y. Jin, Q. Dou, H. Chen, L. Yu, J. Qin, C.-W. Fu, and P.-A. Heng. SV-RCNet: Workflow recognition from surgical videos using recurrent convolutional network. *IEEE Transactions on Medical Imaging*, 37(5):1114–1126, 2018. 2

[13] Y. Konnai, K. Fukumoto, M. Takeuchi, R. Takeuchi, S. Fujiwara, Y. Yasumizu, N. Tanaka, T. Takeda, K. Matsumoto, T. Kosaka, H. Kawakubo, Y. Kitagawa, and M. Oya. Artificial-intelligence-based surgical phase recognition in robot-assisted radical prostatectomy and cross-surgeon validation. *Annals of Surgical Oncology*, 33:1870–1877, 2026. 2

[14] L. Maier-Hein, M. Eisenmann, C. Feldmann, H. Feussner, G. Forestier, S. Giannarou, B. Gibaud, G. D. Hager, M. Hashizume, D. Katic, H. Kenngott, R. Kikinis, M. Kranzfelder, A. Malpani, K. März, B. Müuller-Stich, N. Navab, T. Neumuth, N. Padoy, A. Park, C. Pugh, N. Schoch, D. Stoyanov, R. Taylor, M. Wagner, S. S. Vedula, P. Jannin*, and S. Speidel*. Surgical data science: A consensus perspective, 2018. 1, 2

[15] W. McKinney. Data structures for statistical computing in Python. In *Proceedings of the 9th Python in Science Conference*, pages 56–61, 2010. 5

[16] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*. 2019. 4

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, et al. Scikit-learn: Machine learning in Python. In *Journal of Machine Learning Research*, volume 12, pages 2825–2830, 2011. 5

[18] PyTorch Contributors. Torchvision: Pytorch's computer vision library, 2016. 4

[19] A. P. Twinanda. *Vision-based Approaches for Surgical Activity Recognition Using Laparoscopic and RGBD Videos*. PhD thesis, University of Strasbourg, 2017. 2

[20] A. P. Twinanda, S. Shehata, D. Mutter, J. Marescaux, M. de Mathelin, and N. Padoy. EndoNet: A deep architecture for recognition tasks on laparoscopic videos. *IEEE Transactions on Medical Imaging*, 36(1):86–97, 2017. 2, 8

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 2