Event-Level Detection of Surgical Instrument Handovers in Videos with Interpretable Vision Models

Supplementary Material

This document provides additional material supporting the main paper. It includes a detailed description of the training procedure, implementation details of the evaluated models, and additional qualitative results illustrating model behavior. These supplementary analyses complement the experimental results presented in the main manuscript.

A. Training Procedure

Algorithm 1 summarizes the training procedure of the proposed framework.

B. Implementation Details

This section provides details on the models used in this work.

B.1. Multi-Task ViT-LSTM

The Multi-Task ViT-LSTM uses a Pytorch Image Models checkpoint of a ViT-Large/14 backbone pretrained with the DINOv2 self-supervised method [27] on the LVD-142M dataset. Table 3 contains the detailed model architecture and training setup.

The models are trained on a single NVIDIA RTX 6000 Ada GPU. Each epoch processes one third of the dataset. A weighted random sampler is used with fixed class sampling probabilities of 0.6, 0.2, and 0.2 for *assistant idle*, *assistant receives*, and *assistant gives*, respectively. The AdamW optimizer is used. Regarding the tranformations, *JitteredCenterCrop* crops a fixed-size fraction of the image around the centre, but randomly jitters the crop centre within a specified horizontal and vertical range. More specifically, it crops 40% of the image width and 71.1% of the height around the centre, randomly shifting the crop by up to $\pm 3\%$ horizontally and $\pm 5\%$ vertically before ensuring the crop remains within the image boundaries. For testing, no shifting is applied. *ColorJitter* is applied with brightness ± 0.2, contrast ± 0.1, saturation ± 0.2, and hue ± 0.05. The probability of *HorizontalFlip* is 0.5.

B.2. VideoMamba

As a comparison model, we employ a VideoMamba backbone [23] pretrained on Kinetics-400 at 224×224 resolution with 8 input frames. The backbone's classification head is replaced by a custom projection head consisting of four linear layers with LayerNorm, GELU activations, and dropout (rate 0.3), mapping from the 576-dimensional CLS token to the three handover classes.

Algorithm 1: Training on windowed surgical video

Input: Video frames $\{I_t\}_{t=1}^{N}$; sampled frames T=8; frame stride s_f=4; sequence stride s_s=2; labels $y \in \{0, 1, 2\}$

Output: Trained parameters of ViT backbone f_{ViT}, projection g, LSTM f_{LSTM}, detection head, direction head

foreach *training iteration* **do**

Sample sequence start indices $\{t_i\}_{i=1}^{B}$;

// Construct temporal input sequences

$X \leftarrow \{ I_{t_i + k s_f} \}_{i=1,k=0}^{B,T-1}$;

// $X \in \mathbb{R}^{B \times T \times H \times W \times 3}$

Apply training augmentation to all frames in X;

// Spatial encoding (per frame)

$F \leftarrow f_{\mathrm{ViT}}(X)$; // frame-wise ViT features

// Projection to temporal embedding space

$E \leftarrow g(F)$; // $E \in \mathbb{R}^{B \times T \times D}$

// Temporal aggregation

$z \leftarrow f_{\mathrm{LSTM}}(E)$; // final hidden state $z \in \mathbb{R}^{B \times H}$

// Predictions

$\hat{p}_{\mathrm{det}} \leftarrow \sigma(\mathrm{Head}_{\mathrm{det}}(z))$;

$\hat{\mathbf{p}}_{\mathrm{dir}} \leftarrow \mathrm{softmax}(\mathrm{Head}_{\mathrm{dir}}(z))$;

// Targets

$y_{\mathrm{det}} \leftarrow \mathbb{1}[y \neq 2]$;

$m \leftarrow y_{\mathrm{det}}$;

$y_{\mathrm{dir}} \leftarrow y$;

// Multi-task loss

$\mathcal{L}_{\mathrm{det}} \leftarrow \mathrm{WBCE}(\hat{p}_{\mathrm{det}}, y_{\mathrm{det}})$;

$\mathcal{L}_{\mathrm{dir}} \leftarrow \mathrm{WCE}(\hat{\mathbf{p}}_{\mathrm{dir}}[m], y_{\mathrm{dir}}[m])$;

$\mathcal{L} \leftarrow \lambda_{\mathrm{det}} \mathcal{L}_{\mathrm{det}} + \lambda_{\mathrm{dir}} \mathcal{L}_{\mathrm{dir}}$;

Update parameters via backpropagation on $\mathcal{L}$;

We selectively fine-tune the last 12 of 24 Mamba blocks along with the final normalization layer, keeping the remaining backbone frozen. The backbone is trained with a learning rate of 5×10^{-5} and the projection head at 1×10^{-4}, both using AdamW with a weight decay of 5×10^{-4} and a cosine annealing schedule with warm restarts every 10 epochs. Classification is performed for the center frame of each clip via a relaxed majority-vote labeling over a 5-frame window during training. More details are provided in

Table 3. Implementation details of ViT-LSTM model.

Quantity	Value
Image Input Size	518 x 518
Feature Projection Dimension	64
Backbone Learning Rate	3×10^{-6}
Backbone Weight Decay	1×10^{-4}
Backbone Layers Frozen	18 of 24
Backbone Ouput Dropout Rate	0.3
LSTM Learning Rate	1×10^{-5}
LSTM Weight Decay	1×10^{-5}
LSTM Hidden Size	64
LSTM Hidden Layers	1
LSTM Output Dropout Rate	0.4
Batch Size	8
LR Scheduler	5% Linear Warmup + Cosine Annealing
Gradient Accumulation Steps	2
Effective Batch Size	16
Max Gradient Norm	1.0
Loss Weighting	$w_{pos} = 1.5$ $\lambda_{\text{det}} = 2.5$ $\lambda_{\text{dir}} = 1$
Number of Epochs (Incl. Early Stopping)	10
Training Augmentations	JitteredCenterCrop ColorJitter HorizontalFlip
Test Augmentations	JitteredCenterCrop

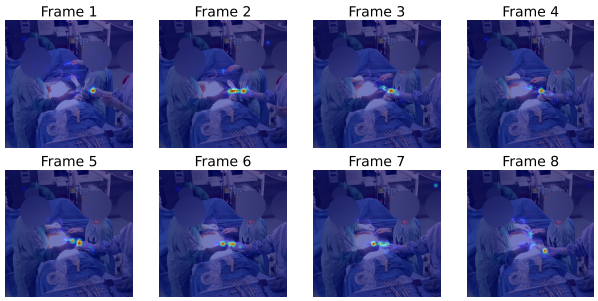
Table 4.

C. Additional Figures

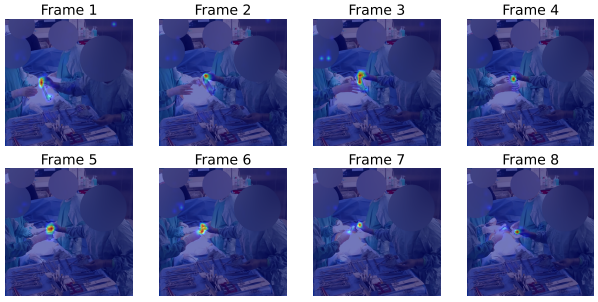
Additional qualitative examples of Layer-CAM explanations are provided in Fig. 6 to illustrate the spatial regions contributing to handover detection.

Table 4. Implementation details of the VideoMamba comparison model.

Quantity	Value
Backbone Variant	VideoMamba-Middle
Pretraining Dataset	Kinetics-400
Image Input Size	512 × 512
Number of Input Frames	8
Backbone Embedding Dimension	576
Backbone Layers Frozen	12 of 24
Backbone Learning Rate	5×10^{-5}
Projection Head Learning Rate	1×10^{-4}
Weight Decay	5×10^{-4}
Projection Head Architecture	576–256–128–128–1
Projection Head Dropout Rate	0.3
Drop Path Rate (Backbone)	0.1
LR Scheduler	CosineAnnealingWarmRestarts (T_0=10)
Training Label Strategy	Majority vote (5-frame window)
Loss Function	BCEWithLogits (weighted)
Training Augmentations	JitteredCenterCrop ColorJitter HorizontalFlip
Test Augmentations	JitteredCenterCrop



(a) Training Set



(b) Test Set

Figure 6. Layer-CAM explanation maps illustrating the contribution of each frame to the handover detection prediction. The explanation maps are generated using the second-to-last and third-to-last layers of the model.