Supplementary Material fors SurgXBench: Explainable Vision-Language Model Benchmark for Surgery

Anonymous WACV Applications Track submission

Paper ID 805

A. Post-hoc XAI Visualization Techniques - Mathematical Formulations

This section provides the detailed mathematical formulations for the post-hoc XAI visualization methods described in Section 3.2 of the main paper.

A.1. Grad-CAM for ResNet VLM

For VLMs that use ResNet-style backbones [4–6], the Grad-CAM formulation is defined as follows. Let $I \in \mathbb{R}^{H' \times W' \times 3}$ be the input image, and let $A \in \mathbb{R}^{K \times H \times W}$ denote the output of a selected convolutional layer in the image encoder, where K is the number of channels and $H \times W$ is the spatial resolution. Let $A_k(i,j)$ denote the activation at spatial position (i,j) in channel k. Let $\mathcal{P} = \{P_1, P_2, \ldots, P_N\}$ be a set contain N candidate natural language prompts. The vision-language model computes an embedding for the image $\phi_{\text{img}}(I)$ and for each text prompt $\phi_{\text{text}}(P_k)$, and calculates a similarity score S_{P_k} (typically cosine similarity) for each prompt in $\mathcal{P}$:

$$S_{P_k} = \frac{\phi_{\text{img}}(I)^\top \phi_{\text{text}}(P_k)}{\|\phi_{\text{img}}(I)\| \cdot \|\phi_{\text{text}}(P_k)\|}.$$

The predicted prompt $\hat{P}$ is then selected as the one with the highest similarity score:

$$\hat{P} = \arg\max_{P_k \in \mathcal{P}} S_{P_k},$$

and let $S_{\hat{P}}$ denote the scalar similarity score corresponding to the predicted prompt $\hat{P}$. Grad-CAM computes the gradient of the score S_c with respect to the feature map $A_k(i,j)$, and then applies global average pooling over spatial dimension:

$$\alpha_k = \frac{1}{H \cdot W} \sum_{i=1}^{H} \sum_{j=1}^{W} \frac{\partial S_{\hat{P}}}{\partial A_k(i,j)}.$$

The gradient tells how much each feature map channel A_k contributes to the model's prediction $S_{\hat{c}}$. it does this by averaging the gradient of predicted score with respect to all locations (i,j) in that channel and use it act as a weight that reflects the importance of that channel. Based on it, to compute the Grad-CAM heatmap, we first obtain a weighted combination of the activation maps using channel-wise importance weights α_k, followed by a ReLU. Finally, we apply bilinearly upsampling to resize L to the size of original image.

$$L_{\text{Grad-CAM}} = \text{Upsample}_{\text{bilinear}}\left(\text{ReLU}\left(\sum_{k=1}^{K} \alpha_k \cdot A_k\right)\right). \tag{1}$$

A.2. Gradient-Based Attention Rollout for CLIP-ViT

Following the method from [1], in transformer model, we leverage the self-attention matrices and their gradients to identify input image patches that contribute most to the similarity between an image and a given text prompt. Let $\mathcal{T}^{(l)} \in \mathbb{R}^{N \times N}$ denote the self-attention matrix from the l-th transformer block in the ViT encoder, where N is the number of tokens, a special token added at the beginning of the input, and its output representation is used to summarize the entire image for classification tasks. We extract $\mathcal{T}^{(l)}$ after the softmax operation during the forward pass.

We compute the gradient of the similarity score with respect to the attention map:

$$\nabla \mathcal{T}^{(l)} = \frac{\partial S_{\hat{P}}}{\partial \mathcal{T}^{(l)}},$$

where $S_{\hat{P}}$ is the predicted similarity score corresponding to the selected text prompt in a zero-shot setting. To capture influence on the similarity score, we compute the

gradient-weighted attention map:

$$\tilde{\mathcal{T}}^{(l)} = \text{ReLU}\left(\nabla\mathcal{T}^{(l)} \odot \mathcal{T}^{(l)}\right),$$

where $\odot$ denotes the elementwise (Hadamard) product. This highlights token-to-token interactions that both receive high attention and significantly impact the similarity score. To propagate relevance through the transformer layers, we recursively update a relevance matrix $\mathbf{R}^{(l)} \in \mathbb{R}^{N\times N}$ as follows:

$$\mathbf{R}^{(L)} = I, \quad \mathbf{R}^{(l)} = \mathbf{R}^{(l+1)} + \tilde{\mathcal{T}}^{(l)}\mathbf{R}^{(l+1)},$$

starting from the final layer L, where I is the $N \times N$ identity matrix. This recursive formulation accumulates relevance flow across all layers. Let each patch token indexed by $k \in \{1, \ldots, N-1\}$, after propagating to the input layer, we extract the relevance scores from the `[CLS]` token to each image patch token:

$$r_k = \mathbf{R}^{(0)}_{\text{CLS},\,k}, \quad \text{for } k = 1, \ldots, N-1,$$

where r_k denotes the relevance of the k-th image patch token to the final similarity score.

Finally, we reshape and upsample the vector $[r_1, \ldots, r_{N-1}] \in \mathbb{R}^{N-1}$ into the input image size using bilinear interpolation. The final heatmap can be constructed as following:

$$L_{\text{Grad-CAM-ViT}} = \text{Upsample}_{\text{bilinear}}\left(\text{reshape}\left([r_1, \ldots, r_{N-1}]\right)\right), \tag{2}$$

A.3. Grad-CAM for Multimodal Large Language Models

For large language models, we adopt a method inspired by [7]. Since the predicted classes are generated as natural language, we apply Grad-CAM to specific tokens in the output sequence. To obtain a Grad-CAM heatmap for a given token t_j, we isolate its corresponding logit z_j and compute the gradient with respect to the visual feature maps A_k from the last layer of the language model's decoder. Specifically, we compute:

$$G_k = \frac{\partial z_j}{\partial A_k}.$$

The remaining steps follow the standard Grad-CAM procedure described earlier.

A.3.1. CLEANN: Causal Learning for Attention Networks

Unlike contrastive learning, where taking the derivative of the similarity score with respect to visual tokens or feature maps can provide straightforward visual explainability, large vision-language models (LVLMs) involve more complex reasoning and interaction processes. As discussed in [7], Grad-CAM in LVLMs may indicate where the model focuses during generation, but it does not fully capture the underlying reasoning. As a result, our attention analysis procedure for LLaVA serves more as a soft interpretability measure. To better support our investigation, we adopt the Causal Graph Analysis method CLEANN [2, 3], which provides a more principled view of the model's internal reasoning. For target token t and relevant token set $\mathcal{T}$, the method tests conditional independence between token pairs $X, Y \in \mathcal{T}$ given conditioning set $\mathcal{Z} \subset \mathcal{T}$ using their attention distributions, where $\mathbf{a}_X$ represents the attention vector (row) of token X in the attention matrix. This approach enables the discovery of how visual attention patterns causally influence language generation beyond simple correlation in multi-modal transformer architectures.

B. Camera Motion Correction - Mathematical Formulation

This section provides the detailed mathematical formulation for the camera motion correction method described in the main paper.

B.1. Camera Motion Model

We model the global camera motion as a linear combination of four basic operations: Pan (P), Tilt (T), Zoom (Z), and Roll (R). Each operation corresponds to a known optical flow prototype vector at any given pixel coordinate (x, y) relative to the image center:

Pan (p): $p = \begin{pmatrix}-1\\0\end{pmatrix}$, **Tilt (t):** $t = \begin{pmatrix}0\\-1\end{pmatrix}$, **Zoom (z):** $z = \begin{pmatrix}-x\\-y\end{pmatrix}$, **Roll (r):** $r = \begin{pmatrix}y\\-x\end{pmatrix}$.

The predicted flow from the camera, f_{cam}, is a weighted sum of these prototypes:

$$f_{\text{cam}} = P \cdot p + T \cdot t + Z \cdot z + R \cdot r$$

B.2. Bidirectional Optimization

To find the actual camera motion parameters robustly, we first use RAFT to compute total flow in both directions—the forward flow (f_{fwd}) and the backward flow (f_{bwd})—and solve a least-squares problem over the valid pixel domain $\Omega \subset \mathbb{R}^2$. The optimization goal is to find the parameters P, T, Z, R that minimize the total squared error $E = E_{\text{fwd}} + E_{\text{bwd}}$, which is the sum of the error contributions from both directions.

The forward error component is:

$$E_{\text{fwd}} = \sum_{(x,y)\in\Omega} \|f_{\text{cam}}(x,y) - f_{\text{fwd}}(x,y)\|^2$$
$$= \sum_{(x,y)\in\Omega} \|(P\cdot p + T\cdot t + Z\cdot z + R\cdot r) - f_{\text{fwd}}(x,y)\|^2 \quad (3)$$

The backward error component is:

$$E_{\text{bwd}} = \sum_{(x,y)\in\Omega} \|f_{\text{cam}}(x,y) + f_{\text{bwd}}(x,y)\|^2,$$

where the backward error assumes that camera motion should be consistent in both temporal directions, with the backward flow representing the negative of the forward camera motion.

Solving this enhanced problem yields the best-fit parameters for the camera's movement. With these parameters, we construct the pure camera flow field, f_{cam}. Finally, we correct the original forward flow by subtracting this calculated camera motion, leaving only the motion of the independent objects:

$$f_{\text{corrected}} = f_{\text{fwd}} - f_{\text{cam}}$$

C. Implementation Details

C.1. Parameters

All heatmaps from post-hoc XAI methods are thresholded using percentile-based Grad-CAM, where $\tau = 0.8$ selects the top 20% high-attention regions. For RAFT-derived binary masks, we use a motion threshold of $\gamma = 0.8$. Across CLIP, BLIP, SurgVLP, HecVL, and PeskaVLP, a class is predicted if its similarity score exceeds the 90th percentile ($\delta = 0.9$). For the triplet recognition task, due to the large number of class combinations, we select the top 5 classes with highest similarity scores as predictions. CLEANN is implemented with key parameters: attention threshold $\kappa = 0.01$ to filter relevant tokens; p-value threshold $\alpha = 10^{-5}$ to enforce strong statistical significance in conditional independence tests; sample size $n = 128$ for reliable inference; and a cap of 50 image tokens per analysis to balance efficiency and coverage. Rather than analyzing all attention heads, we select the top-5 heads that prioritize visual over textual content. Head importance is computed as head_importance $= \frac{\max(\text{image\_attention})}{\max(\text{text\_attention})}$ where higher values indicate greater visual focus for the target prediction.

C.2. Prompts Design

To align with VLMs' prompt-based input format, we construct template-based prompts. For instrument classification, we use: *"What surgical tools do you see? Choose from: Grasper, Bipolar, Hook, Scissors, Clipper, Irrigator, Bag"* covering all instrument classes. For triplet reasoning, we design a detailed prompt that guides the model to identify surgical actions by providing instrument capabilities and target options. The prompt instructs the model to look at the surgical image carefully and identify all visible actions, providing specific lists of what each instrument can do (e.g., grasper can grasp, retract, dissect, or manipulate tissue) and all possible surgical targets (cystic_plate, gallbladder, omentum, etc.). We constrain the output to a three-word format: "instrument verb target".

D. Metrics

D.1. Standard Instrument Classification Metrics

Let TP, FP, and FN denote the number of true positives, false positives, and false negatives across all instrument classes. The metrics are computed as: Precision $=$ TP$/$(TP $+$ FP), Recall $=$ TP$/$(TP $+$ FN), $F1 = 2 \cdot$ Precision $\cdot$ Recall$/$(Precision $+$ Recall).

D.2. Standard Triplet Recognition Metrics:

We adopt the standard triplet metrics from [**?**]. Each predicted action is represented as a triplet of class labels $\langle \textit{instrument}, \textit{verb}, \textit{target}\rangle = (s, v, o)$, with the corresponding ground-truth triplet denoted as (s^*, v^*, o^*). Binary match indicators are defined as:

$$\text{Match}(s, v, o; s^*, v^*, o^*) = (\mathbb{1}_{s=s^*},\ \mathbb{1}_{v=v^*},\ \mathbb{1}_{o=o^*}).$$

The standard evaluation metrics focus on three specific matching conditions: IVT (instrument–verb–target), IV (instrument–verb), and IT (instrument–target). These metrics provide a finer understanding of the model's ability to recognize structured surgical actions, which are inherently instrument-centric; the verb and target gain full semantic meaning only in the context of the acting instrument.

E. Additional Results

E.1. Instrument classification

Addition results include pericison, recall and box plot of attentiona quality metrics for instrument classification result are provided in figure 1

E.2. Occlusion Analysis

Additional results for the occlusion analysis are provided in Figure 2 showing successful tool alignment under occluded conditions. The left panel shows cases where occlusion blocks irrelevant areas, yet the correct

IVT(instrument-verb-target) match is no longer among the top-5 predictions. The right panel shows cases where relevant regions are occluded, but the correct prediction still appears in the top-5. Prompts highlighted in green indicate triplets that achieve IVT coverage with the ground truth, while gray prompts represent cases where the instrument was not correctly identified. The green bounding box on the images represents the instrument-relevant area in the triplets.

E.3. Causal Graph Analysis

Two additional results for triplet recognition are shown in Figure 3. These figures highlight same findings discussed in Section 5 of the main paper, causal graphs generated using the CLEANN method (radius 3, top-5 visually important attention heads). Generated responses are in token format where each box contains the causal graph of each head for each token. Green circles indicate visual tokens that are in or close to the relevant area, and white boxes represent text tokens that were previously generated.

E.4. Text Token Ratio Analysis

Follow the same implementation details, we conducted a text token ratio analysis using a CLEANN-based causal selection approach. This method identifies tokens that directly influence target generation through conditional independence testing and computes text–visual ratios among these causal dependencies. The results reveal substantial text over-reliance: for example, tokens such as `cutting` exhibited 77.6% text dominance, while others like `c` (from "cystic") and `sc` (from "scissors") also showed notable text bias. These findings confirm that surgical vision–language models often depend heavily on linguistic context rather than visual evidence, even when processing visually grounded terms. Detailed results are summarized in Table 1.

Table 1. Causal Analysis (CLEANN) of Text Dominance in Token Generation

Subword Token	Text Dom.	Visual Dom.
`cut`	0.428	0.572
`yst`†	0.333	0.667
`c`†	0.690	0.310
`plate`	0.240	0.760
`re`†	0.495	0.505
`cutting`	0.776	0.224
`sc`†	0.612	0.388
`gr`†	0.294	0.706

† Subword fragment from tokenization (e.g., `yst` from "cystic", `re` from "retract", `gr` from "grasper", `sc` from "scissors")

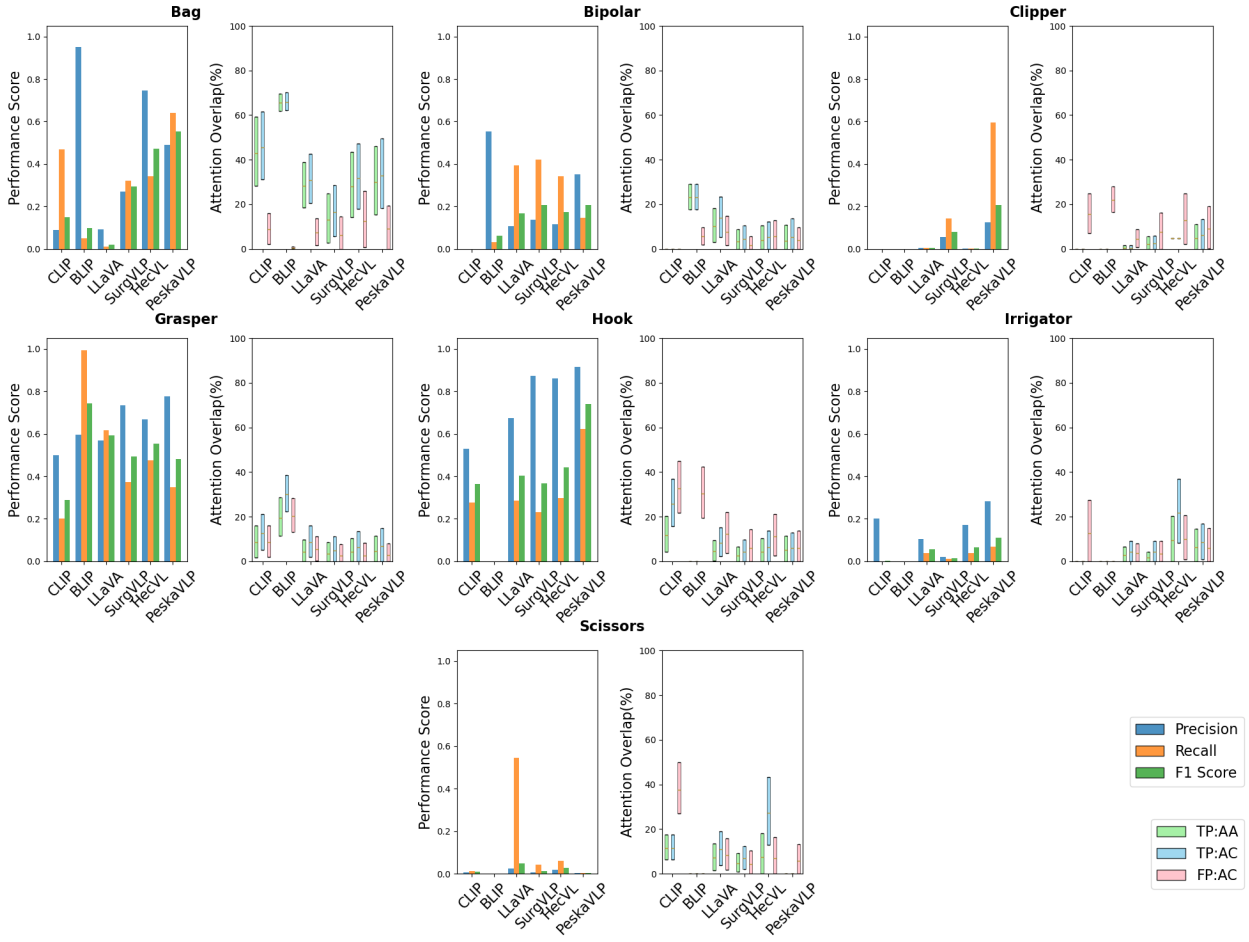

Figure 1. Surgical instrument detection performance and attention alignment across seven instrument types from the **Cholec80BBox** dataset. For each instrument, the left subplot shows classification performance (Precision, Recall, F1 Score) and the right subplot displays attention overlap AA and AC score between heatmaps and ground truth regions as box plots for True Positive and False Positive predictions.

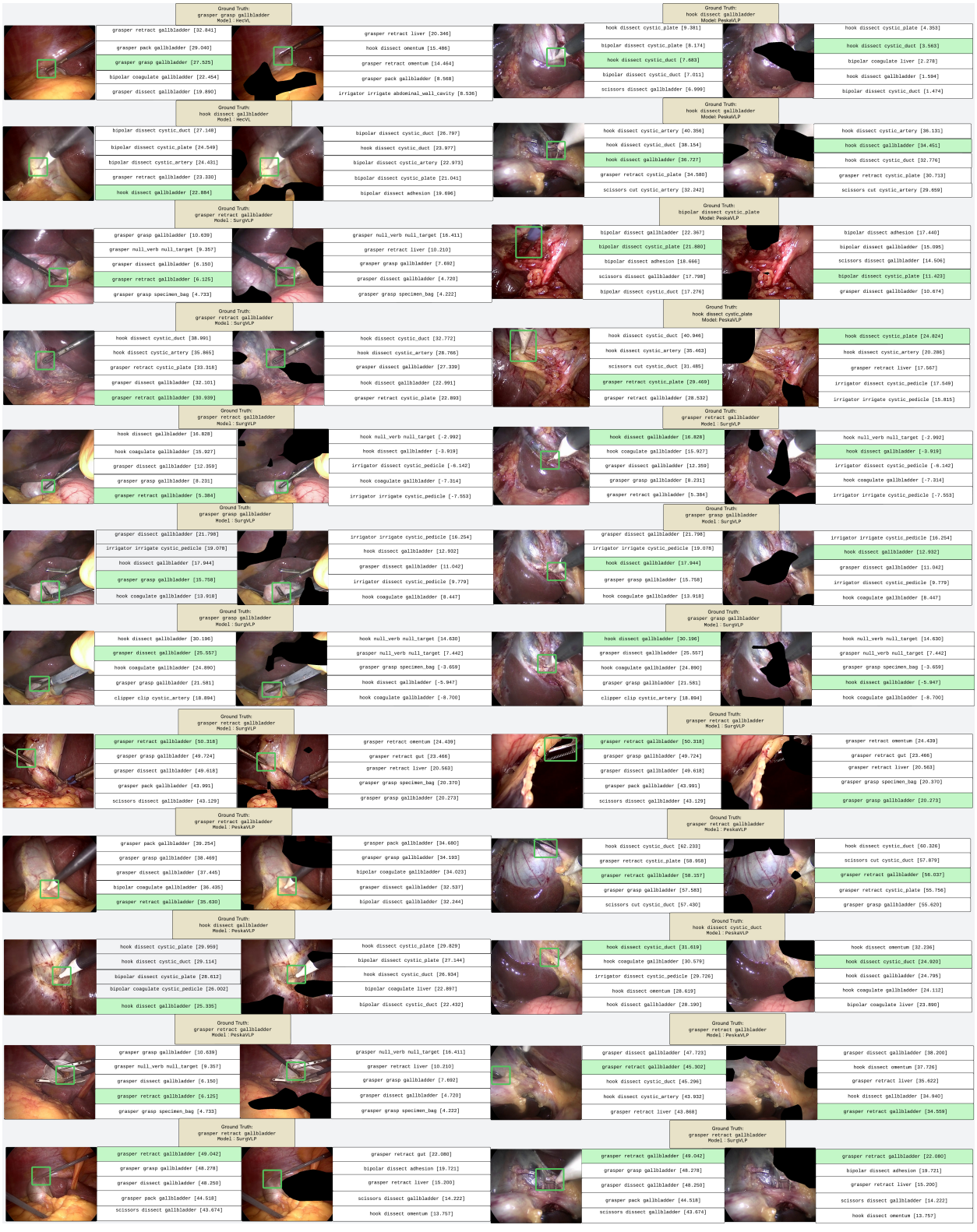

Figure 2. Occlusion analysis results for SurgVLP, HecVLP, and PeskaVLP.

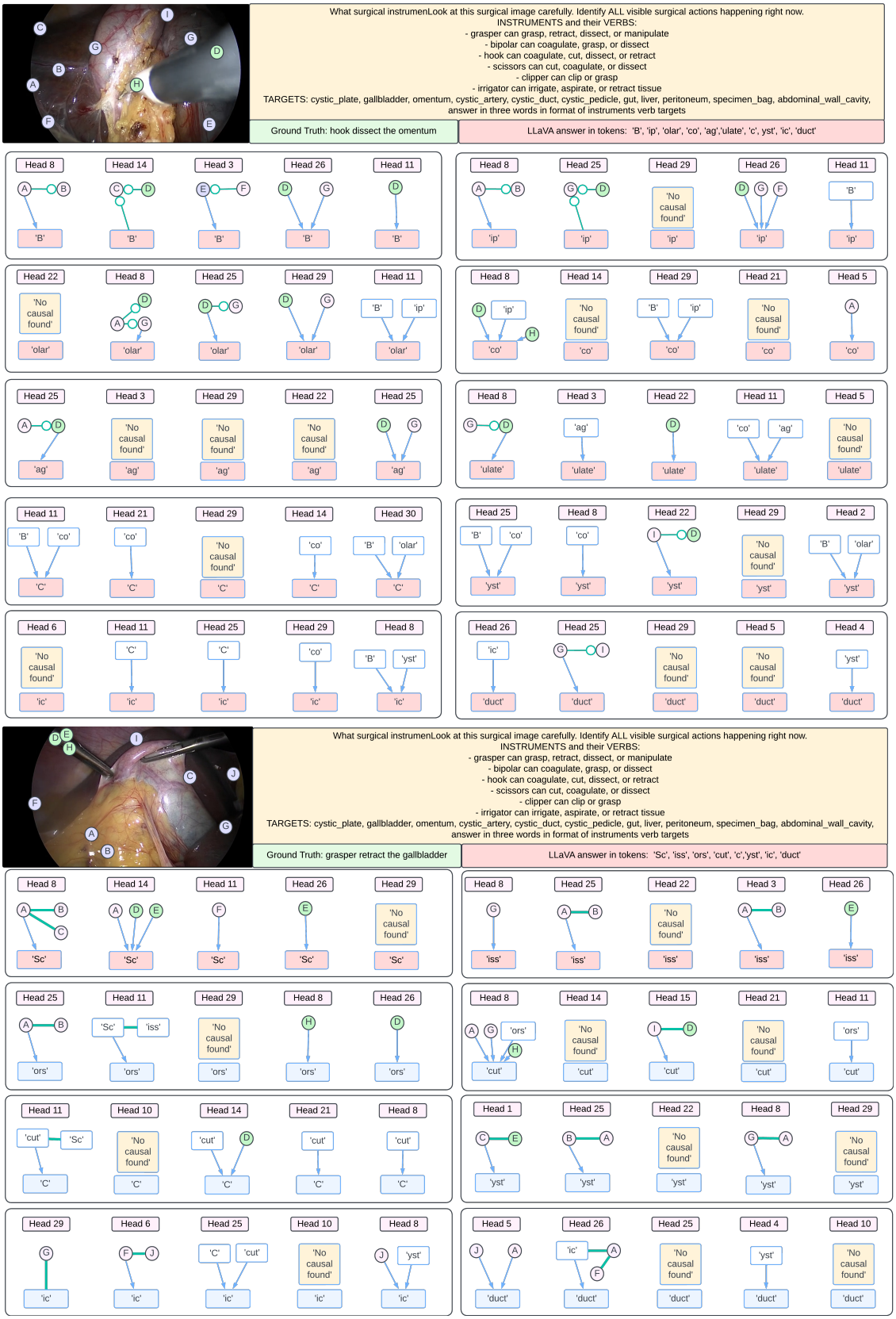

Figure 3. This figure shows triplet recognition examples from LLaVA

References

[1] Hila Chefer, Shir Gur, and Lior Wolf. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 397–406, 2021.

[2] R. Y. Rohekar, Y. Gurwicz, and S. Nisimov. Causal interpretation of self-attention in pre-trained transformers. In *Advances in Neural Information Processing Systems*, volume 36, pages 31450–31465, 2023.

[3] G. B. M. Stan, E. Aflalo, R. Y. Rohekar, et al. Lvlm-interpret: An interpretability tool for large vision-language models. *arXiv preprint arXiv:2404.03118*, 2024.

[4] Kun Yuan, Vinkle Srivastav, Nassir Navab, and Nicolas Padoy. Hecvl: Hierarchical video-language pretraining for zero-shot surgical phase recognition. *arXiv preprint arXiv:2405.10075*, 2024.

[5] Kun Yuan, Vinkle Srivastav, Nassir Navab, and Nicolas Padoy. Procedure-aware surgical video-language pretraining with hierarchical knowledge augmentation. In *Proceedings of the 38th Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[6] Kun Yuan, Vinkle Srivastav, Tong Yu, Joël L. Lavanchy, Jacques Marescaux, Pietro Mascagni, Nassir Navab, and Nicolas Padoy. Learning multi-modal representations by watching hundreds of surgical video lectures. *arXiv preprint arXiv:2307.15220*, 2023.

[7] X. Zhang, Y. Quan, C. Shen, X. Yuan, S. Yan, L. Xie, W. Wang, C. Gu, H. Tang, and J. Ye. From redundancy to relevance: Information flow in lvlms across reasoning tasks. *arXiv preprint arXiv:2406.06579*, 2024.