Chain-of-Look Spatial Reasoning for Dense Surgical Instrument Counting
Supplementary Materials

1. Analysis on Visual Chain Reasoning via Neighboring Loss

We demonstrate that the neighboring loss enforces a Chain-of-Look mechanism within the model's reasoning process. Figure 1 visualizes the self-attention scores of the query proposals in the Cross-Modality Decoder. At the first decoder layer (Layer 0), where the model primarily captures low-level spatial cues, we observe that removing the neighboring loss results in higher attention entropy, with focus spread across non-adjacent queries. In contrast, applying $\mathcal{L}_{\text{neigh}}$ constrains each query to attend mainly to its immediate predecessors and successors, forming a snake-like chained structure.

At the final decoder layer (Layer 5), the attention maps show that this chained behavior also shapes high-level semantic reasoning. For instruments that are densely clustered (labels 5–8), the model leverages the most visible and confident queries as anchors, reflected by the pronounced dark attention band, to improve the representation of uncertain and ambiguous queries. These structured interactions suppress hallucinations and ultimately improve counting accuracy.

2. Implementation Details

We train the model for 30 epochs with a learning rate of 1×10^{-4} using the Adam optimizer and a weight decay of 1×10^{-4}, which is reduced by a factor of ten after the 10^{th} epoch. Training is performed with a batch size of 4 on a single NVIDIA RTX 3090 GPU. The multi-loss weights are set as follows: $\lambda_{\text{loc}} = 10$, $\lambda_{\text{neigh}} = 100$, and $\lambda_{\text{cls}} = 1$. The number of CSL prompts used is 64, and the confidence threshold σ is set to 0.26. The rest of the training setup, including data pre-processing and augmentation strategies, follows the original CountGD [1] configuration.

3. CSL Prompt Design and Implementation

The modified Feature Enhancer takes two different Class Specific Learnable (CSL) token instances, both initialized with the same text but diversified with Gaussian noise (Fig. 2). The first set of tokens are prepended to the concatenated set of visual exemplar and text tokens. We treat the CSL token as tunable text prompts, hence they are prepended to the latent tokens derived from text. These tokens, along with the image token, are used in the Bidirectional Attention module, where image-text and text-image cross-attention are computed.

The second set of CSL token is prepended to the output of the bidirectional module and fed into the self-attention layer, where the self-attention between the text tokens is calculated.

We introduce two sets of tokens to represent distinct functional roles: one captures the relation between text and image, while the other addresses text-specific nuances. The fused feature embedding $\mathbf{F}$ is constructed by concatenating these components:

$$\mathbf{F}_{encoder} = [\mathbf{T}_{\text{CSL}}\,;\,\mathbf{T}_{\text{text}}\,;\,\mathbf{T}_{\text{vis}}] \in \mathbb{R}^{(l+2h)\times d}, \quad (1)$$

where $\mathbf{T}_{\text{CSL}} \in \mathbb{R}^{l\times d}$ denotes the l CSL tokens, while $\mathbf{T}_{\text{text}}$ and $\mathbf{T}_{\text{vis}}$ (both $\in \mathbb{R}^{h\times d}$) represent the text and visual exemplar tokens, respectively. Following standard prompt-tuning methodology, we discard $\mathbf{T}_{\text{CSL}}$ before passing the sequence to the decoder.

$$\mathbf{F}_{decoder} = [\mathbf{T}_{\text{text}}\,;\,\mathbf{T}_{\text{vis}}] \in \mathbb{R}^{(l+h)\times d} \quad (2)$$

3.1. Value Function for Hungarian Matching

We utilize the CountGD formulation to define the Hungarian matching value function $v(i,k)$ between prediction i and ground-truth k. Given $\alpha = 0.25, \gamma = 2$:

$$\begin{aligned} v(i,k) &= \underbrace{\|\mathbf{b}_i - \mathbf{b}_k\|_1}_{\textbf{bbox cost}} + \underbrace{\sum_{j=1}^{C} \tilde{y}_{kj}\,[\mathcal{L}_{\text{pos}}(p_{ij}) - \mathcal{L}_{\text{neg}}(p_{ij})]}_{\textbf{cls cost}}, \\ \text{s.t.}\quad & \mathcal{L}_{\text{pos}}(p) = -\alpha(1-p)^{\gamma}\log(p+\varepsilon), \\ & \mathcal{L}_{\text{neg}}(p) = -(1-\alpha)p^{\gamma}\log(1-p+\varepsilon), \end{aligned} \quad (3)$$

where $\mathbf{b}$ represents box center coordinates, $\tilde{y}_{kj}$ is the normalized target label for class j, and p_{ij} is the predicted probability.

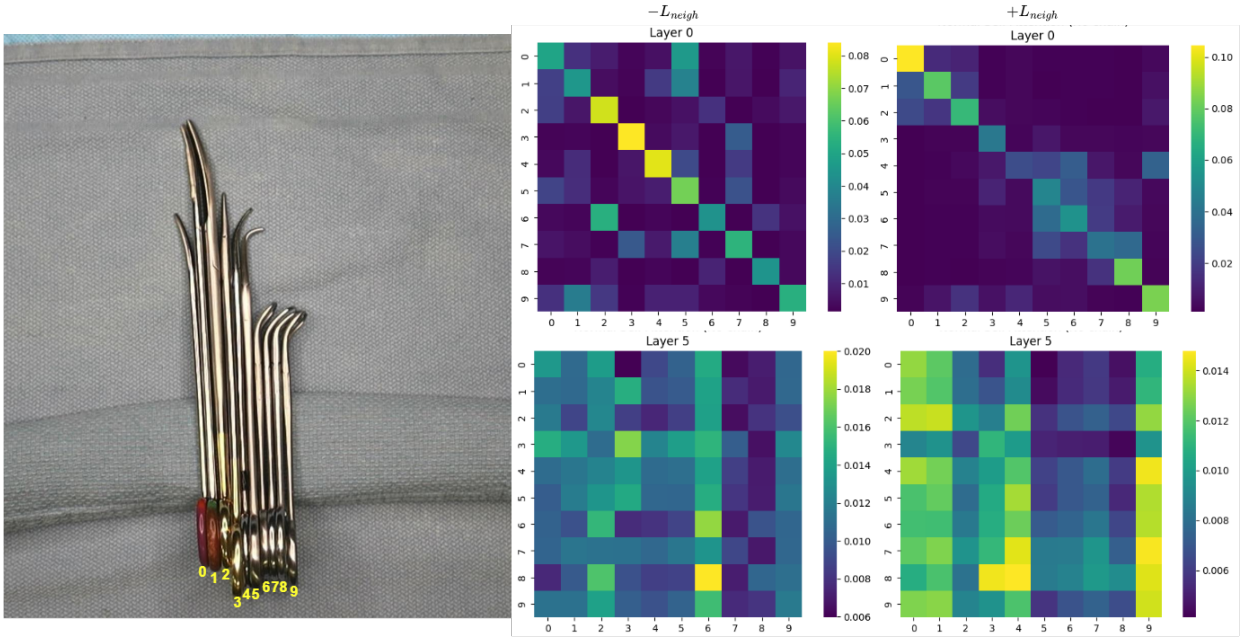

Figure 1. **Analysis on Chain-of-Look Visual Reasoning via Spatial Neighboring Loss.** Left: original surgical image. Right: attention maps from different decoder layers. "$-L_{\text{neigh}}$" denotes models trained without the Neighboring Loss, whereas "$+L_{\text{neigh}}$" indicates models trained with it. The visualizations show the self-attention outputs of the Cross-Modality Decoder, where each query corresponds to one surgical instrument (indexed 0–9). Queries and their associated attention distributions are ordered left-to-right according to the instrument labels in the original image. For each setting, we display attention maps from the first decoder layer (Layer 0), which primarily captures low-level spatial relationships, and from the final decoder layer (Layer 5), which reflects higher-level semantic focus.

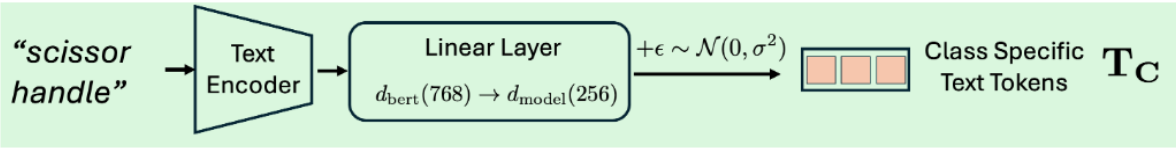

Figure 2. CSL Prompts Initialization with BERT Text Encoder

4. Inference Speed

Our model is lightweight and achieves fast inference, running over ***100*** $\times$ ***faster*** than manual human counting. The experiments were performed across a range of scenarios, with the number of surgical instruments varying from 7 to 49 per trial. In each trial, two individuals performed manual counts, followed by a count using the mobile application. Figure 3 illustrates the contrast in performance between the traditional method and the app-based approach, highlighting the real-time efficiency gains enabled by the proposed system.

5. Extended Ablation Results

5.1. CSL Prompts Placement: Appending vs. Prepending

Previous studies [6] have highlighted how prompt placement affect transformer models. Our analysis (Table. 1) reveals that prompt placement significantly impacts performance, with prepending yielding 32% better MAE than appending. Attention analysis shows prepended prompts maintain 3× stronger coupling with image features, while gradient flow analysis indicates on average 177× stronger supervision signals, explaining the performance disparity.

5.1.1. Metrics Definitions

CSL Token Gradient Norms: We measured the gradient norms for both types of CSL tokens-text and fusion-across all six encoder layers. For each type, the gradients were av-

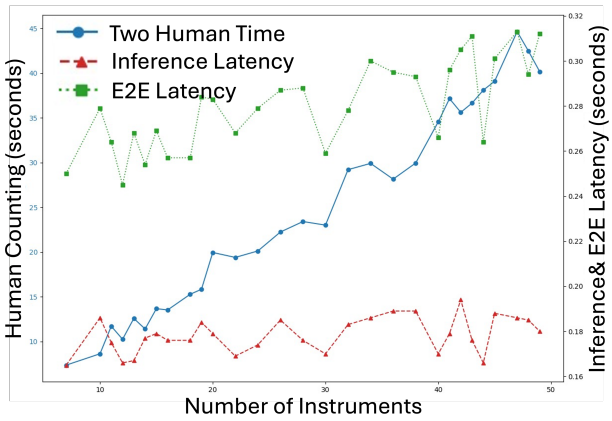

Figure 3. Time comparison between human counting and our model

Placement	MAE ↓	RMSE ↓	CSL Token Grad ↑	Vision Attention Weights ↑
Append	1.30	1.98	Total: 0.044	Mean: 0.00225
			Avg: 0.0037	Std: 0.00325
Prepend	0.88	1.27	Total: 7.876	Mean: 0.00643
			Avg: 0.656	Std: 0.00498

Table 1. **Prompt Placement** Performance comparison across CSL prompt placements. See Section 5.1.1 for metric definitions.

Initialization Type	MAE	RMSE
Random	1.32	1.96
Semantic ("scissor handle")	0.88	1.96

Table 2. **Prompt Initialization Strategy** Prepending task-specific initialized CSL prompts yields better performance compared to random initialization.

eraged over the layers using a single training batch to assess their relative contribution during backpropagation.

Vision Multi-Head Attention Weights at Fusion Module: To analyze the influence of prompts on visual attention, we extracted attention weights from the fusion module. Prompt-related weights were stacked and averaged across all four attention heads, followed by averaging over the batch dimension. This process was repeated for each of the six encoder layers, and the resulting layer-wise averages were further averaged to obtain a final mean value. The same training batch was used for both prompt placement configurations to ensure consistency.

We further investigate the impact of prompt initialization when prompts are prepended. Table 2 shows that using task-specific initialization leads to notable performance improvements.

5.2. LoRA versus CSL Tokens

We explore whether adding explicit spatial conditioning using learnable prompt tokens offers benefits over weight adaptation methods in our instrument counting task. To test this, we inserted LoRA [4] adapters at the fusion and text encoder layers, mirroring the placement of CSL tokens. The adapter configuration (α=32, Rank =16) was chosen to match the parameter count of the CSL tokens, allowing for a fair comparison.

Method	MAE ↓	RMSE ↓	Mean L2 (matched) ↓	Mean IoU (matched) ↑
LoRA	5.63	7.66	10.42	0.028
CSL Tokens	0.88	1.27	6.38	0.290

Table 3. **Counting & Localization Metrics: LoRA vs. CSL Tokens.** The Mean IoU is the average IoU of all the matched bounding boxes in the test set.

LoRA Parameters

- Rank=16, α=32
- LoRA Params Per Layer : 16 (Rank) × 256 (In Features) + 16 (Rank) × 256 (Out Features) = 8,192 parameters
- Text Encoder layers: 6 layers × 2 LoRA modules × 8,192 Per Layer ≈ 98K parameters
- Fusion layers: 6 layers × 2 LoRA modules × 8,192 ≈ 98K parameters
- Total LoRA Parameters ≈ 196K parameters

CSL Parameters

- Text Encoder layers: 64 (CSL Token) × 256 (feature dim) × 6 layers ≈ 98K parameters
- Fusion layers: 64 (CSL Token) × 256 (feature dim) × 6 layers ≈ 98K parameters
- Total CSL Parameters ≈ 196K parameters

Our comparison between CSL Tokens and LoRA shows that token-level spatial conditioning leads to superior object detection performance despite LoRA's parameter efficiency. In our experiments, LoRA struggled with instrument localization, often missing center points and producing inaccurate bounding boxes (Figure. 4). This suggests that LoRA's weight-space adaptation may lack the direct spatial conditioning beneficial for precise object localization in our setting. In contrast, the contrastive learning capability shown with CSL tokens (Section 6) appears to improve spatial reasoning that goes beyond parameter efficiency considerations. Our findings suggest that for this spatially sensitive detection task, explicit spatial conditioning through prompt tokens may provide capabilities that our constrained low-rank weight modification approach could not achieve.

5.3. Multi-Loss Weight Selection

Our method incorporates three distinct loss functions: Cross-Entropy Loss ($\mathcal{L}_{cls}$), Distance Loss ($\mathcal{L}_{loc}$), and Neighboring Loss ($\mathcal{L}_{neigh}$). We measured gradient norms across key shared model layers (encoder, decoder, fusion, text) for three loss weighting configurations to validate our λ selection strategy.

As shown in Table. 4 there exists a high imbalance in gradient magnitudes between the cross-entropy (CE) loss

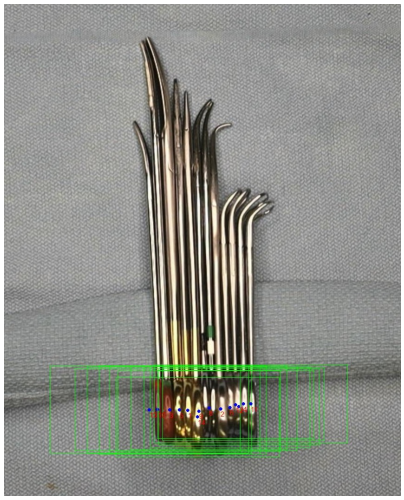

Figure 4. Predicted bounding boxes using the LoRA method. Boxes are noticeably oversized and misaligned.

$(\lambda_{cls}, \lambda_{loc}, \lambda_{neigh})$	$\mathcal{L}_{cls}$ **Grad**	$\mathcal{L}_{loc}$ **Grad**	$\mathcal{L}_{neigh}$ **Grad**	**MAE**	**RMSE**
(1, 1, 1)	0.915	0.0005	0.0002	3.23	4.20
(1, 10, 100)	0.737	0.0007	0.0003	2.35	3.59
(1, 10, 100)	**28.50**	**0.055**	**0.106**	**0.88**	**1.27**

Table 4. **Gradient Magnitude Analysis** Multi-Loss scaling factor selection.

and auxiliary losses, with the latter exhibiting gradients 1,000–4,000× weaker under equal weighting. As a result, auxiliary objectives are effectively ignored during training. By introducing a loss weighting configuration of λ = (1, 10, 100), we observe a substantial increase in auxiliary contribution (0.56% vs. 0.09%) while preserving CE dominance. This leads to a 30× increase in total gradient activity, enabling more expressive multi-objective optimization.

6. CSL Prompts Effect and Contrastive Feature Learning

When trained without CSL prompts, the model's attention is spread across and less focused on the handle regions, as illustrated in Fig. 5(b–c). In contrast, CSL tokens learn contrastive features, as shown in Fig. 5(d), where the attention on the handle is minimal. This complementary negation helps the text tokens to attend to the handle regions more precisely. We experimented with varying numbers of CSL prompts {16, 32, 64, 128}, and found that 64 prompts produced the best performance based on the MAE metric.

7. Determining Instrument Orientation for Neighboring Loss

Firstly, we extract the center points from the predicted bounding boxes. Using these center points, we compute the difference between the maximum and minimum coordinates along the x- and y-axes. The axis with the largest difference is considered the dominant orientation of the instruments.

$$1 - \text{int}\left(\left(P_x^{\max} - P_x^{\min}\right) > \left(P_y^{\max} - P_y^{\min}\right)\right) = \begin{cases} 0 & \text{for } x\text{-axis} \\ 1 & \text{for } y\text{-axis}, \end{cases} \tag{4}$$

where:

$$\begin{aligned} P_{\text{pred}} &= \{(x_i, y_i) \mid \text{point } i \text{ is predicted}\}, \\ P_x &= \{x_i \mid (x_i, y_i) \in P_{\text{predicted}}\} \quad \text{with } P_x \subseteq [0, W], \\ P_y &= \{y_i \mid (x_i, y_i) \in P_{\text{predicted}}\} \quad \text{with } P_y \subseteq [0, H], \end{aligned} \tag{5}$$

W and H denote the weight and height of the image.

8. Evaluation Metrics

8.1. Counting Metrics

8.1.1. MAE, RMSE

We use the standard Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE) to measure as evaluation metrics.

$$\begin{aligned} \text{MAE} &= \frac{1}{N}\sum_{i=1}^{N}|N_P - N_G|, \\ \text{RMSE} &= \sqrt{\frac{1}{N}\sum_{i=1}^{N}(N_P - N_G)^2}, \end{aligned} \tag{6}$$

where N is the number of image samples, N_P is the predicted count and N_G is the ground truth count for image N_i.

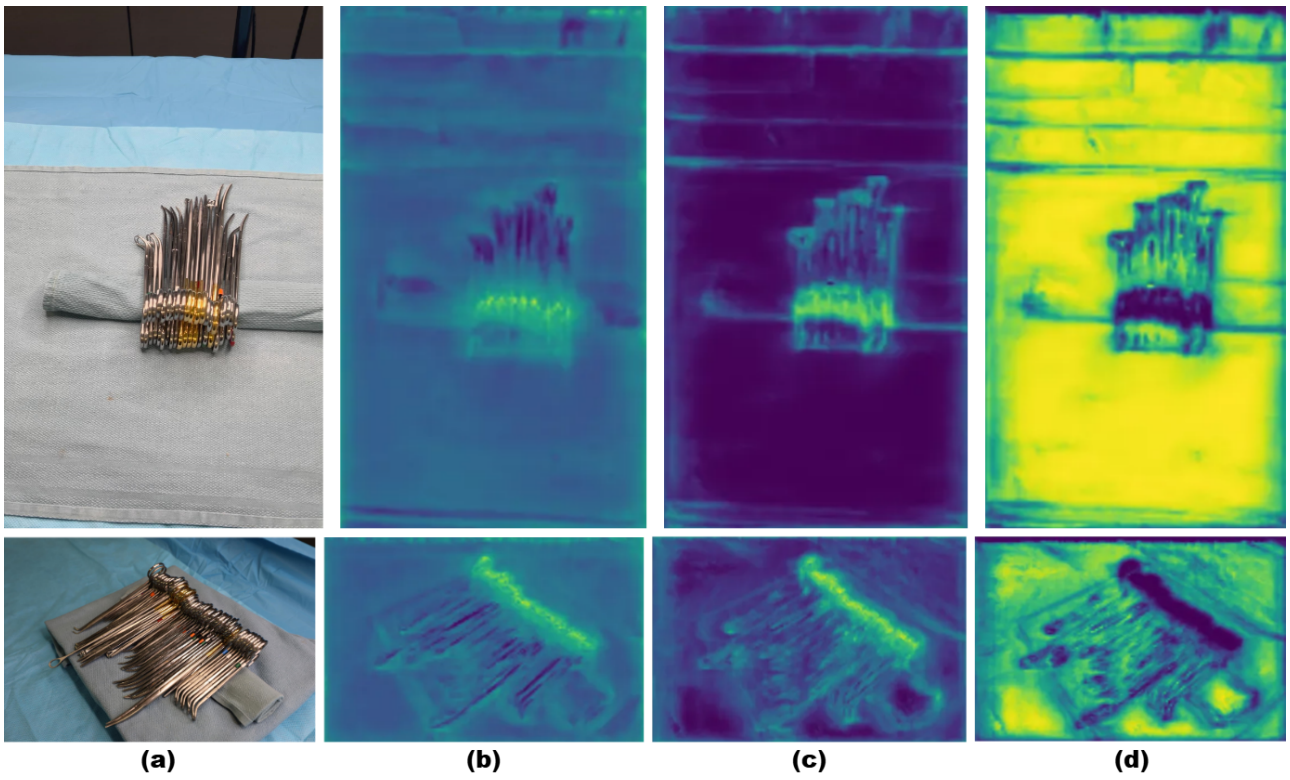

Figure 5. **a)** Original Input Image **b)** Image-Text Attention Map extracted from the Feature Fusion Block - Without CSL Prompts **c)** Image-Text Attention Map when trained with CSL Prompts **d)** Image-CSL Token Attention Map

8.1.2. Grid Average Mean Absolute Error

We also measure the Grid Average Mean Absolute Error (GAME) [3] to evaluate the spatial accuracy of the predicted counts within each image. GAME quantifies how well the counting predictions are localized across subdivided regions of the image. Given that surgical instruments in our dataset are typically concentrated within a limited spatial area, the GAME scores tend to decrease as the grid resolution parameter L increases (Table. 5). This is due to the presence of numerous grids containing no instruments, which contribute zero error to the overall score. Of the 228 images in our test set, only 98 include instance-level annotations suitable for spatial evaluation. Therefore, the GAME scores and localization metrics were calculated exclusively on this subset.

Moreover, we also use detection-related counting metrics such as precision, recall and F1-score as defined in Equation. 7.

Method	GAME-L1 ↓	GAME-L2 ↓	GAME-L3 ↓
CountGD [1]	1.01	0.41	0.14
REC [2]	0.60	0.25	0.08
DQ-DETR [5]	0.68	0.25	0.07
CoLSR (Ours)	**0.54**	**0.23**	**0.07**

Table 5. GAME scores (L1, L2, L3) for different methods.

8.2. Localization Metrics

Since the number of predicted instrument locations may not match the ground truth (GT) annotations, computing localization accuracy is non-trivial. To address this, we first filter predictions by selecting only those whose center points fall within any GT bounding box. These filtered predictions are then matched to GT points. In cases where multiple predictions fall within the same GT box, we apply the Hungarian algorithm using L2 distance as the cost function to perform one-to-one matching.

Unmatched predictions are treated as missed detections, while matched pairs are used to compute localization metrics. Specifically, for each image, we calculate the mean L2 distance (average localization error), the median L2 distance (typical error at the 50th percentile), and the 95th percentile of L2 distances (representing the worst 5% of matched localizations). To obtain a single dataset-level metric, we take the mean of these three values across all images (Table. 6). A similar procedure is applied for computing the Mean IoU reported in the Table. 3.

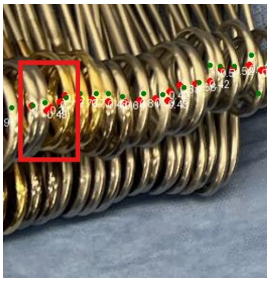

Figure 6. Example of duplicate points highlighted

Steps for a single input :

$$
\begin{aligned}
&P_{\text{filtered}} = p \in P_{\text{pred}} \mid \exists b \in B_{GT} \text{ such that } p \in b \\
&M^* = \underset{M}{\text{argmin}} \sum_{(p,g)\in M} |p - g|_2 \\
&d_i = |p_i - g_i|_2 \\
&\bar{d} = \frac{1}{N} \sum_{i=1}^{N} d_i \qquad (7) \\
&\text{Median Error} = \text{median}(d_1, d_2, \ldots, d_N) \\
&\text{95th Percentile Error} = P_{95}(d_1, d_2, \ldots, d_N) \\
&\text{True Positive (TP)} = N \\
&\text{False Positive} = len(P_{\text{pred}}) - TP \\
&\text{False Negative} = len(G_{GT}) - TP
\end{aligned}
$$

where :

P_{pred} : The set of all predicted center points.

B_{GT} : The set of all ground truth (GT) bounding boxes.

G_{GT} : The set of all GT center points.

P_{filtered} : The set of all filtered center point prediction.

M^* : The optimal one-to-one matching

d_i : The L2 distance for the i-th matched pair(p_i, g_i)

N : The total number of matched pairs.

9. Post Processing Operator

Due to the dense and ambiguous appearance of the instruments in the images, the model frequently produces multiple duplicate detections close to each other (Fig. 6). To mitigate this, we applied a post-processing step to eliminate such points.

First, we sort the detected center points from left to right or top to bottom based on their orientation (Section 7). For each detected point, we examine neighboring points within a distance threshold θ along the given axis. If multiple points are found within this range, we retain only the point with the highest confidence score and discard the others, Alg: 1.

Algorithm 1: Point Selection with Distance Threshold

for $P_i, P_j \in \{left, right\}$ **do**
  if $|P_i - P_j| < d$ **then**
    $P_{selected} \leftarrow \arg\max_{P\in\{P_i,P_j\}} \text{conf}(P)$
    $P_{removed} \leftarrow \arg\min_{P\in\{P_i,P_j\}} \text{conf}(P)$
    Remove $P_{removed}$ from set
  end if
end for

10. Divide and Conquer Inference

CoLSR is designed to handle densely packed instrument clusters, which are the most common setup in real-world surgeries. However, its performance degrades when multiple dense clusters are spatially separated (Fig.7-a). This is due to the visual chain constraint enforced by the neighboring loss, which fails to capture long-range dependencies in such cases. To address this, we follow a two-stage approach, the Divide-and-Conquer strategy (Alg: **??**,**??**). In the first stage, the entire image is processed by the network, and the predicted center points are sorted along the x- or y-axis, depending on the instrument orientation (as described in 7).

We then compute the L2 norm between neighboring center points. If the distance between two neighbors exceeds δ, the points on the left are grouped into one, and those on the right into another. This process is repeated until all center points are assigned to clusters. Each cluster is then cropped from the original image and second-stage inference is performed independently on each dense region. Finally, the predictions of all the clusters are stitched together to produce the final output (Fig.7-b).

	CountGD [1]	REC [2]	DQ-DETR [5]	**CoLSR (Ours)**
Mean L2 distance ↓	12.79	6.89	**5.84**	6.43
Mean of Median L2 distance ↓	12.01	6.33	**5.46**	5.99
Mean of 95th-Percentile L2 distance ↓	21.05	12.66	**10.56**	11.44
Precision ↑	0.41	0.73	0.84	**0.85**
Recall ↑	0.41	0.74	0.81	**0.84**
F1 score ↑	0.41	0.74	0.83	**0.85**

Table 6. Comparison of localization metrics results across different methods.

11. Limitations of Generalization

While our method demonstrates robustness to variations in angle and lighting conditions typical of operating room (OR) environments (Figure. 8, Main Paper - Figure. 7), the scope of this work is limited to surgical instrument counting, as indicated by the paper title. Consequently, generalization to other domains may require further investigation.

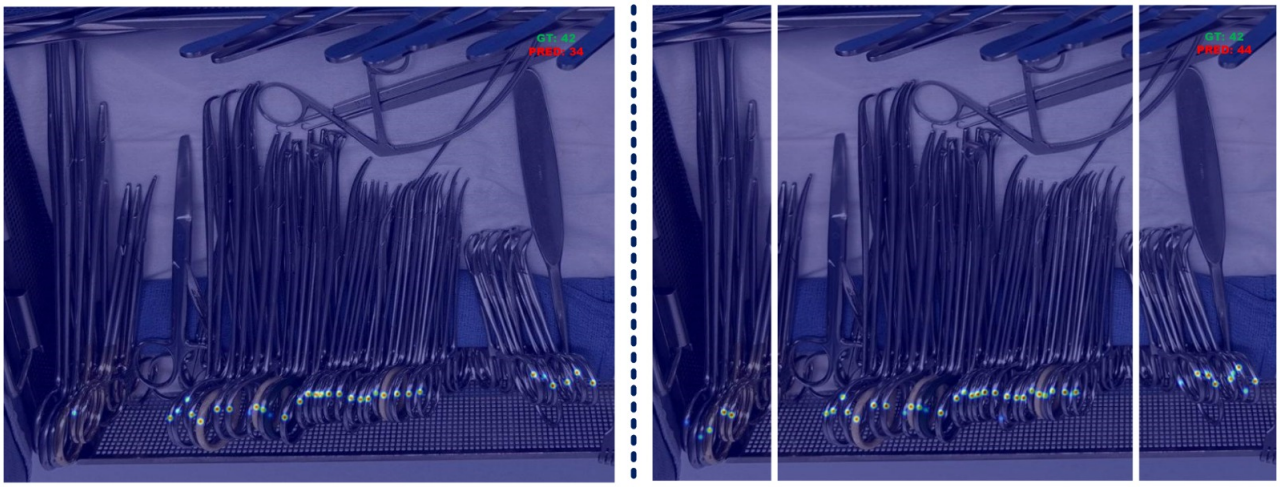

Figure 7. **a)** Prediction with single-pass inference **b)** Prediction with Divide-and-Conquer approach

Algorithm 2: Distance-Based Clustering

$clusters \leftarrow \{\}$
$cluster \leftarrow [0]$ {cluster start}
for $i = 0$ to $|pred\_points| - 2$ **do**
  $p_i \leftarrow pred\_points[i]$
  $p_{i+1} \leftarrow pred\_points[i+1]$
  if $\|p_i - p_{i+1}\|_2 > \delta$ **then**
    $cluster$.append(i) {cluster end}
    $clusters$.append($cluster$)
    $cluster \leftarrow [i+1]$ {next cluster start}
  end if
end for
$slices \leftarrow$ slice_image($clusters$)
return $slices$

Algorithm 3: Two-Pass Counting

$pred\_points \leftarrow$ run_inference($image$) {first pass}
$slices \leftarrow$ create_cluster($pred\_points, \delta$)
$final\_detections \leftarrow \{\}$
for $slice \in slices$ **do**
  $pred\_points \leftarrow$ run_inference($slice$) {second pass}
  $final\_detections$.append($pred\_points$)
end for
return $final\_detections$

Figure 8. **Robust inference samples captured from multiple angles.**

References

[1] Niki Amini-Naieni, Tengda Han, and Andrew Zisserman. Countgd: Multi-modal open-world counting. *Advances in Neural Information Processing Systems*, 37:48810–48837, 2024. 1, 5, 6

[2] Siyang Dai, Jun Liu, and Ngai-Man Cheung. Referring expression counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16985–16995, 2024. 5, 6

[3] Ricardo Guerrero-Gómez-Olmedo, Beatriz Torre-Jiménez, Roberto López-Sastre, Saturnino Maldonado-Bascón, and Daniel Onoro-Rubio. Extremely overlapping vehicle counting. In *Iberian conference on pattern recognition and image analysis*, pages 423–431. Springer, 2015. 5

[4] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 3

[5] Yi-Xin Huang, Hou-I Liu, Hong-Han Shuai, and Wen-Huang Cheng. Dq-detr: Detr with dynamic query for tiny object detection, 2024. 5, 6

[6] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021. 2