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

Rishikesh Bhyri   Brian R Quaranto   Junsong Yuan   Peter C W Kim [†]   Nan Xi [†]

State University of New York at Buffalo

{rbhyri, brianqua, jsyuan, pckim, nanxi}@buffalo.edu
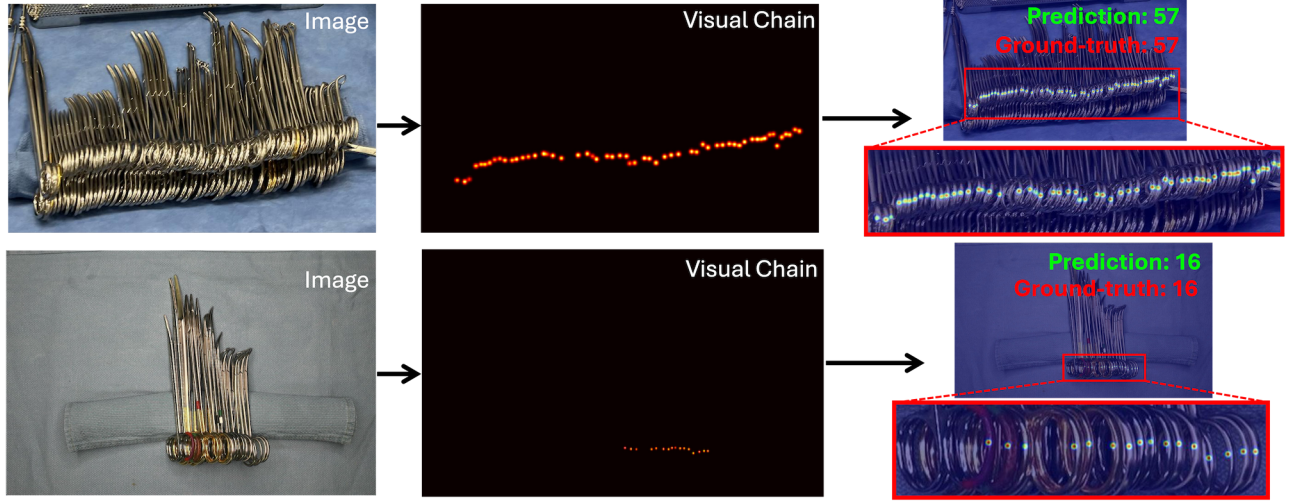
[†] Equal Advising

Figure 1. **High-density surgical instrument counting.** Counting surgical instruments reliably in high density scenarios is challenging due to severe visual clutter and tight spatial packing of objects. To improve robustness, we propose Chain-of-Look Spatial Reasoning to introduce *visual chains* into the counting process, explicitly modeling the sequential characteristic of human visual counting. In the above figure, the first column indicates original high-density surgical instrument images, the second column presents visual chains and the third column shows the predicted counting results, where detected surgical instrument handles are highlighted with laser points.

Abstract

Accurate counting of surgical instruments in Operating Rooms (OR) is a critical prerequisite for ensuring patient safety during surgery. Despite recent progress of large visual-language models and agentic AI, accurately counting such instruments remains highly challenging, particularly in dense scenarios where instruments are tightly clustered. To address this problem, we introduce Chain-of-Look, a novel visual reasoning framework that mimics the sequential human counting process by enforcing a structured visual chain, rather than relying on classic object detection which is unordered. This visual chain guides the model to count along a coherent spatial trajectory, improving accuracy in complex scenes. To further enforce the physical plausibility of the visual chain, we introduce the neighboring loss function, which explicitly models the spatial constraints inherent to densely packed surgical instruments. We also present SurgCount-HD, a new dataset comprising 1,464 high-density surgical instrument images. Extensive experiments demonstrate that our method outperforms state-of-the-art approaches for counting (e.g., CountGD, REC) as well as Multimodality Large Language Models (e.g., Qwen, ChatGPT) in the challenging task of dense surgical instrument counting. The code and dataset is available at https://github.com/rishi1134/CoLSR.git

1. Introduction

Counting surgical objects including instruments before and after a surgical procedure is a critical safety protocol in Operating Rooms (OR), aimed at preventing retained surgical items. Despite its importance, this process is still predominantly performed manually by surgical staff, making

it time-consuming, labor-intensive, and prone to human error, particularly in high-density settings where instruments are closely clustered or visually occluded. These challenges are further exacerbated under time pressure or in emergency procedures. Given that the average operating room costs approximately $100 per minute, delays caused by manual counting can have significant financial implications in addition to clinical risks. Thus, automating the surgical instrument counting process holds great potential to reduce the workload on surgical teams, minimize human errors, and enhance workflow efficiency and patient safety. However, accurate automated counting remains a challenging task due to visual complexity and high similarity among instruments in real-world OR environments.

Most existing approaches to object counting fall into two main categories: density map-based and detection-based methods. Density map-based methods estimate object counts by summing the predicted density values across an image, while detection-based methods count the number of predicted bounding boxes. Although these methods have achieved strong performance in various scenarios such as crowd counting and open-set counting, they fundamentally treat counting as a set-based problem, ignoring the sequential nature of how humans count, particularly in complex, high-density environments. In practice, humans typically follow a consistent visual path when counting objects to avoid omissions or duplications, and to verify the counting. For instance, as shown in Figure 1, technicians and nurses counting the surgical instruments tend to follow a structured visual sequence, such as scanning from left to right or right to left. This sequential reasoning process is crucial in ensuring accurate counts under cluttered and visually challenging conditions, yet it is largely overlooked in existing automated counting frameworks.

Motivated by the importance of sequential visual reasoning in human counting behavior, we introduce the **Chain-of-Look Spatial Reasoning** (CoLSR) framework that explicitly models the counting sequence in dense object scenes by locating each object as a counting point. CoLSR *explicitly models the sequential nature of human counting*, which is particularly critical in *complex, high-density environments*. Introducing a direction for counting is especially important in high-stakes surgical scenarios, where medical staffs always follow a **strict counting direction** to ensure accuracy rather than counting instruments in a random order. Unlike traditional methods that treat object instances independently, our CoLSR not only predicts the locations of target objects (e.g., the handles of surgical instruments in our task) but also captures the *spatial dependencies* and *structural relationships* among them. To achieve this, we first generate visual chains using a transformer-based counting model, CountGD [2]. These visual chains provide guidance for our model, enabling it to reason spatially and im-

prove prediction accuracy. To further align the predicted visual chains with the underlying spatial structure of the scene, we introduce a novel neighboring loss that encourages the predicted object order to match the ground-truth sequence. The neighboring loss encourages the model to consider the proximity and ordering of adjacent objects, and enforces consistency with realistic spatial arrangements by penalizing implausible gaps or overlaps. Therefore, the neighboring loss guides the model to learn a coherent spatial chain that mirrors the sequential patterns humans naturally follow during counting, leading to more robust and accurate performance in high-density scenarios.

We evaluate our CoLSR framework through extensive experiments on a high-density surgical instrument dataset that we construct. Empirical results demonstrate that CoLSR consistently outperforms state-of-the-art (SOTA) object counting methods and multimodality large language models in the context of densely packed surgical instruments, highlighting its effectiveness in real-world, high-complexity scenarios.

In summary, our contributions include:
- We introduce the novel and challenging task of dense surgical instrument counting, a problem with significant clinical implications.
- We introduce the Chain-of-Look Spatial Reasoning framework that incorporates *visual chains* into the counting process, explicitly modeling the sequential nature of human visual counting. We design the neighboring loss to equip the model with spatial reasoning capabilities by enforcing inter-object relationships and realistic spatial constraints.
- We construct a comprehensive dataset comprising 1,464 high-density surgical instrument images collected from diverse real-world clinical settings. Extensive experiments show that CoLSR delivers significant improvements over existing methods for high-density surgical instrument counting, achieving both high accuracy and fast inference.

2. Related Work

2.1. Object Counting

Human Counting. The spatial strategies and typical scanning patterns humans employ when counting stationary objects are deeply rooted in the psychology of enumeration. Gelman et al. [10] highlight the principle of one-to-one correspondence (each item gets one and only one tag) which strongly implies a systematic way of going through a set to ensure accuracy. Logan et al. [24] further demonstrate that the number of eye movement fixations increases linearly with the number of objects during counting, indicating a sequential, item-by-item processing strategy when dealing with larger quantities.

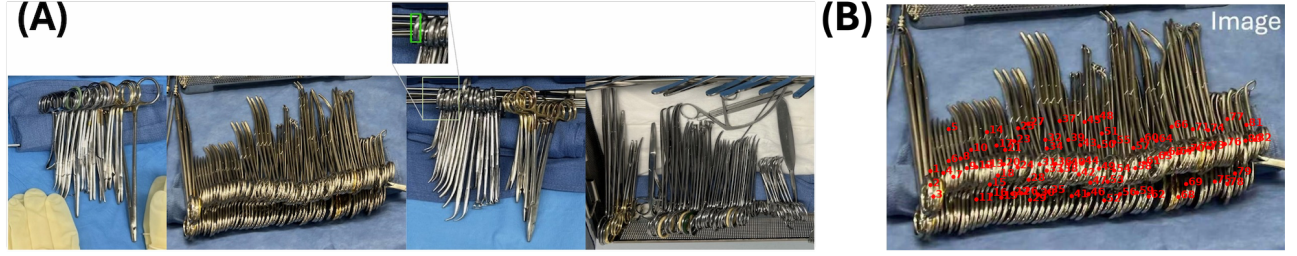**Machine Counting.** Object counting with machines has

Figure 2. (A) **Representative images from the SurgCount-HD dataset.** Sample images from the dataset, showing typical variations and an example annotation. (B) **Test result from GPT5.** We evaluate GPT-5 on an example from our SurgCount-HD dataset, where detected surgical instruments are highlighted with red dots. GPT-5 predicts a count of 84, whereas the ground truth is 57.

traditionally been a detection-based or density-based estimation problem. The density-based approach is particularly prevalent in crowd-counting applications [6, 12, 18, 19, 25, 27, 34] mainly due to its dense and cluttered scenes. Prior work in density-based counting has shown superior counting accuracy [1, 3, 4, 9, 22] compared to detection-based methods, which rely on bounding box prediction, a process that becomes challenging in the presence of overlapping and occluded object boundaries. Recent detection-based approaches [2, 7, 21] have begun to achieve improved accuracy with the advantage of the inherent object localization capability. Recent works like DAVE [26], DQ-DETR [15], and CAD-GD [30] leverage both detection- and density-based approaches to achieve better counting and localization accuracy.

Although CountGD demonstrates strong performance in open-set object counting, its category-agnostic design limits its semantic specificity and dense object localization. However, its open-world setup enables generation of 'good-enough' priors even in few-shot settings, making it a useful guiding mechanism for class-specific models with reduced data and computational overhead.

2.2. Chain-of-Look Prompting

Chain-of-thought prompting [31–33] has been shown to enhance the reasoning capabilities of Large Language Models (LLMs) and reduce hallucinations by breaking down the task into smaller steps in LLMs. Drawing a parallel to vision domain, we adopt the CoL prompting strategy to support spatial reasoning and model the sequential nature of counting by prompting the detected visual cues in a chained fashion, thereby enabling more structured visual attention across densely packed instruments.

2.3. Prompt Tuning

Parameter-Efficient Fine-Tuning (PEFT) approaches, such as prompt tuning [16], have proven effective in adapting to newer data distributions while reducing both data and computational requirements. Beyond efficiency, Yao et al. [35] demonstrate that fusing frozen tokens with learnable prompts further boosts the generalization and discrimina-

tive capability of prompt tuning. Moreover, Kang et al. [17] introduce semantic-conditioned prompts that guide the image encoder toward extracting target-semantic-highlighted visual features.

With these existing contributions in mind, we explore integrating such PEFT approaches with the existing CountGD framework to address its limitations in handling out-of-distribution classes, particularly in highly dense scenarios.

3. Surgical Instrument Counting Dataset

We introduce SurgCount-HD, a novel dataset consisting of **H**igh-**D**ensity arrangements of **Surg**ical instruments collected prior to surgical procedures. Each image contains various types of surgical instruments compactly organized on the back table (a common surgical preparation surface). The dataset focuses on instrument layouts where handles are oriented toward the camera, and bounding-box annotations are provided for these handles, as shown in Figure 2 (A). Translational and rotational augmentations were applied and the final the dataset comprises 1,236 training images and 228 test images. All images were resized such that the shorter edge is scaled to 800 pixels while preserving the original aspect ratio. All annotations represent a single class, namely "*circular instrument handle*".

We used Roboflow [29] platform to manually label instrument handles across densely packed scenes. The annotation process required substantial manual effort due to the high density and visual similarity among instruments. The data collection and annotation process spanned several months and involved multiple domain experts to ensure accuracy and consistency.

The SurgCount-HD dataset presents significant challenges due to the tightly clustered and visually occluded surgical instruments. Even for human annotators, counting in such high-density scenarios is time-consuming and labor-intensive. To assess the difficulty of this dataset, we evaluated GPT5 [13] on selected examples from SurgCount-HD. As shown in Figure 2 (B), GPT5 performs poorly in these dense settings (detected 84 instruments, where the ground-truth is 57), highlighting the challenge of this SurgCount-

HD dataset.

4. Chain-of-Look Spatial Reasoning

4.1. Problem Formulation

In this section, we introduce the Chain-of-Look Spatial Reasoning (CoLSR) framework for high density surgical instrument counting. Given an image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ of high density surgical instruments, our goal is to train a model $\mathcal{F}_\theta$ to localize surgical instrument handles and count the number y of surgical instruments in the image based on the localized instrument handles: $y = \mathcal{F}_\theta(I)$, where θ denotes the parameters of our model, H and W represents the height and weight of the given image. Two major components construct the CoLSR framework: (1) *Visual Chain Generator* for producing the visual chain to guide the counting process; (2) *Neighboring Loss Function* to introduce physical constraints along the visual chain.

4.2. Visual Chain Generator

As shown in Figure 3, the first part of CoLSR is to generate visual chain of the given image. The visual chain serves as a *structured visual sequence* to guide the model's counting process under cluttered and visually challenging conditions.

The visual chain generator is constructed based on the CountGD model [2]. Different from CountGD, we also take class-specific text tokens as input to enhance the quality of generated visual chain. Figure 4 (a) depicts the detailed architecture of Visual Chain Generator.

Image Encoder. We first encode the input image $\mathbf{I}$ with a Swin-B version of Swin Transformer [23] based Image Encoder f_I into spatial feature maps at three different scales, followed by 1×1 convolution to produce image tokens $\mathbf{z_I}$ of 256 dimensions. The visual exemplar tokens $\mathbf{z_B}$ are obtained from the image tokens using aligned region-of-interest pooling (RoIAlign) with the pixel coordinates specified by the visual exemplars $\mathbf{B}$. The generated visual exemplar tokens also have 256 dimension, which is the same with image tokens and text tokens.

Text Encoder. For text input, a BERT-based text transformer [8] encoder f_T is employed to encode the text description $\mathbf{T_S}$ into a sequence of tokens $\mathbf{z_T}$ with at most 256 dimensions. Then the n image tokens, p visual exemplar tokens and q text tokens are applied with the feature enhancer f_ϕ.

Feature Enhancer. The generated visual exemplar tokens $\mathbf{z_B}$ is fused with the text tokens $\mathbf{z_T}$ through the feature enhancer f_ϕ with 6 blocks self-attention modules. The generated fused feature $\mathbf{z_{B,T}}$ are further fused with the image tokens $\mathbf{z_I}$ through the feature enhancer f_ϕ with 6 blocks cross-attention modules. To enhance the grounding ability of our model, we take the prompt tuning approach to introduce class-specific text tokens $\mathbf{T_C}$ as additional inputs for the feature enhancer (see supplementary for details).

These class-specific text tokens serve as learnable parameters to further improve the results of generated visual chains. Therefore, the outputs from the feature enhancer are computed as

$$\mathbf{z_{B,T}}, \mathbf{z_I} = f_\phi((f_\theta(\mathbf{X}), \\ \mathrm{RoIAlign}(f_\theta(\mathbf{X}), \mathbf{B}), f_T(\mathbf{T_S}), f_T(\mathbf{T_C})). \tag{1}$$

Query Selection. k image patch tokens are selected which achieve the highest similarity with the fused visual exemplar $\mathbf{B}$ and text description $\mathbf{T_S}$. Following CountGD, we set k to 900, serving as cross-modality queries input to the cross-modality decoder f_ψ.

Cross-modality Decoder. The cross-modality decoder f_ψ contains 6 blocks of self-attention and cross-attention to enhance the cross-modality queries. The final output of confidence score $\hat{\mathbf{Y}}$ is computed as

$$\hat{\mathbf{Y}} = \mathrm{Sigmoid}(f_\psi(\mathbf{z_I}, \mathbf{z_{B,T}}, f_S(\mathbf{z_I}, \mathbf{z_{B,T}}^T, k))\mathbf{z_{B,T}}^T), \tag{2}$$

where f_S denotes the above Query Selection module.

4.3. Neighboring Loss

The generated visual chain provides a coarse estimate of the instrument count and serves as a structural guidance to guide our model toward precise surgical instrument counting. To incorporate directional consistency along the visual chain, we introduce a neighboring loss term into the training objective. As illustrated in Figure 4 (b), given the predicted visual chain $\mathbf{C}$, we first use the Hungarian matching algorithm to associate each predicted bounding box with its corresponding ground-truth bounding box. Specifically, for the predicted $\{b_i\}_{i=1}^{N_P}$ and ground-truth surgical instrument handle $\{b_j\}_{j=1}^{N_G}$, the value function $\mathbf{v}$ for Hungarian matching algorithm is defined as:

$$\mathbf{v}_{i,j} = d_{i,j} + \mathcal{L}_{i,j}^{cls}, \tag{3}$$

where $\mathbf{v}_{i,j}$ is the value function for the pair (i,j) from predictions and ground-truth, $d_{i,j}$ indicates the l_1 norm of the center points and $\mathcal{L}_{cls}$ denotes the classification cost (see supplementary for further details).

Given the matched bounding boxes, we examine the local regions of detected surgical instrument handles in a fixed direction (either left-to-right or right-to-left). As illustrated in Figure 4 (b), we introduce a neighboring loss that encourages the distances between adjacent center points of bounding boxes in the predictions to closely match those in the ground truth:

$$\mathcal{L}_{neigh} = \sum_{i=1}^{N} ||d_P^i - d_G^i||_2, \tag{4}$$

where d_P^i denotes the distance between two neighboring center points of predicted bounding boxes, d_G^i indicates
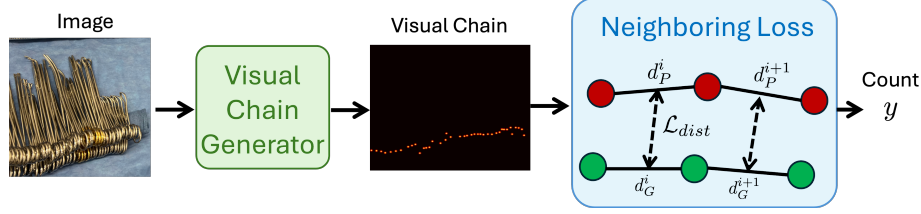
Figure 3. **Architecture of Chain-of-Look Spatial Reasoning framework.** High density surgical instrument images are first fed into visual chain generator to produce visual chains. Neighboring loss is further applied to guide the counting process following the visual chain.
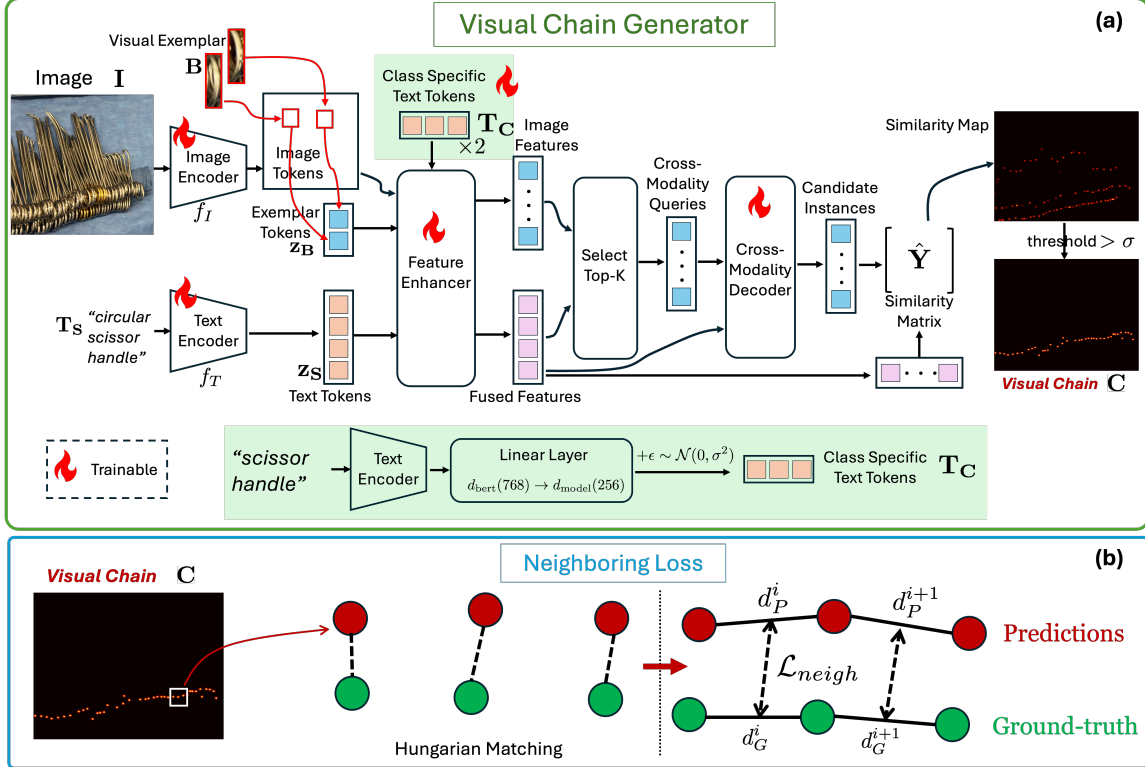

Figure 4. **Visual Chain Generator and Neighboring loss function.** (a) Detailed architecture of Visual Chain Generator; (b) Neighboring loss and Distance loss. Detailed illustrations on the architecture can be found in Section 4.

the distance between two counterpart neighboring center points of ground-truth bounding boxes. This neighboring loss function promotes spatial consistency in the ordering of instruments and enforces a visual chain structure in the model's reasoning process, enabling the Chain-of-Look mechanism. We further discuss this effect in Section 1 of the Supplement Materials.

4.4. Training

As shown in Figure 4 (a), we train the image encoder f_I, text encoder f_T, feature enhancer f_ϕ, cross-modality decoder f_ψ and the learnable class specific text tokens $\mathbf{T_C}$. The optimization objective of the whole model includes CountGD [2]'s original bounding box localization loss, classification loss, and our proposed neighboring loss:

$$\mathcal{L} = \lambda_{loc}\mathcal{L}_{loc} + \lambda_{neigh}\mathcal{L}_{neigh} + \lambda_{cls}\mathcal{L}_{cls}$$
$$= \lambda_{loc}\sum_{i=1}^{N_G}|\hat{c}_i - c_i| + \lambda_{neigh}\sum_{i=1}^{N_G}||d_P^i - d_G^i||_2 \quad (5)$$
$$+ \lambda_{cls}FocalLoss(\hat{\mathbf{Y}}, T),$$

where λ_{loc}, λ_{cls} and λ_{neigh} are weights to control each loss term, $\hat{\mathbf{Y}}$ is the similarity matrix from Equation 2 and $T \in \{0,1\}^{N_P \times (N_G+1)}$ denotes the optimal Hungarian matching between the N_P predicted queries and the N_G ground truth handle instances, including an additional label for "no object" similar to CountGD.

During training, the model receives a high density surgical instrument image $\mathbf{I}$ along with visual exemplars $\mathbf{B}$ as inputs. These inputs are processed through the image encoder

Method	MAE ↓	RMSE ↓
CountGD [2]	7.84	10.71
DQ-DETR [15]	4.24	6.81
CrowdDiff [28]	18.63	22.93
REC [7]	2.82	4.50
Qwen2.5-VL-7B-Instruct [5]	17.06	21.72
CoLSR (Ours)	**0.88**	**1.27**

Table 1. Comparison with state-of-the-art methods, including: (1) counting and detection methods spanning detection-based (DQ-DETR), density-based (CountGD, REC), and diffusion-based (CrowdDiff) approaches; (2) multimodality large vision-language model (Qwen-2.5-VL).

Method	MAE ↓	RMSE ↓
$\Delta\mathcal{L}_{neigh}$	1.81	2.73
ΔCSL	2.05	3.30
ΔVisual Exemplars	1.5	2.21
ΔPost	0.996	1.48
CoLSR (Full)	**0.88**	**1.27**

Table 2. **Ablation study results.** $\Delta\mathcal{L}_{neigh}$: without Neighboring Loss; ΔCSL: without class-specific learnable prompts; Δ Visual Exemplars: without visual exemplars; ΔPost: without post processing.

and cross-modality modules to generate query representations, which are then optimized using the loss functions.

4.5. Inference

During inference, we only pass a high density surgical instrument image $\mathbf{I}$ as input to our model. The outputs are predicted surgical instrument handle bounding boxes. We further execute a post processing operator $\mathcal{P}$ to remove the redundant predicted bounding boxes that share the horizontal regions more than a predetermined threshold τ. Detailed descriptions of post processing operator can be found in the supplement.

5. Experiments

5.1. Evaluation Metrics

We employ Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) as evaluation metrics. The detailed formulation of these two counting metrics and other localization metrics can be found in the supplements.

5.2. Quantitative Results

In Table 1, we compare the performance of CoLSR with state-of-the-art (SOTA) methods on the task of high-density surgical instrument counting. For a fair comparison, all SOTA counting baselines are finetuned on our SurgCount-HD dataset, except Qwen. CoLSR outperforms all competing methods in both MAE and RMSE metrics. The major reason lies in the primary limitation of existing counting methods, where they treat object instances as independent entities, lacking the spatial reasoning necessary to capture the dependencies and structural relationships among densely packed instruments. In contrast, CoLSR explicitly models physical constraints, enabling it to reason over spatial arrangements and structural coherence more effectively. In addition, multimodality large vision-language models (MLVL) such as GPT5 [13] and Qwen-2.5-VL [5] also perform much worse compared with CoLSR, where the MAE of MLVLs are more than 10 times higher than CoLSR.

5.3. Qualitative Results

Figure 5 presents qualitative results of high-density surgical instrument counting using CoLSR. The visualizations high-light the robustness of our approach across various challenging scenarios, including variations in camera angles (Figure 5 (f), (h), (i)), instrument orientations (Figure 5 (b), (c), (e)), and dense packing patterns (Figure 5 (a), (d)).

Figure 7 provides a visual comparison between our approach and existing SOTA methods for high-density instrument counting. As shown, SOTA methods often fail to detect all instrument handles, particularly in cluttered regions, resulting in under-counting. In contrast, CoLSR accurately localizes the instrument handles, as indicated by the cropped bounding boxes, demonstrating its effectiveness in handling densely packed scenes. To evaluate the generalization ability of our method, we test our model on in the wild images from operating rooms. Results in Figure 6 indicate our method continually achieves robust results, demonstrating the generalization ability to real world operating room scenarios. We show more results in supplement material to evaluate the generalization ability of our method.

5.4. Inference Speed

Our model is lightweight and achieves fast inference, running over **_100× faster_** than manual human counting. Our mobile application achieves a peak end-to-end (E2E) latency (including pre-processing, inference, and post-processing) of only 0.32s, compared to 44s required for manual counting. Average latency is 0.28 ± 0.02s for our mobile application versus 25.12 ± 11.63s for human counting. A comparison figure and demo video are included in the supplementary material for clearer visualization.

5.5. Ablation Study

We conduct the following ablation studies to verify the effectiveness of each proposed component, including the neighboring loss, the class-specific learnable prompts and visual exemplars.

Effectiveness of neighboring loss function ($\Delta\mathcal{L}_{neigh}$). Removing the neighboring loss diminishes the model's ability to construct a visual chain for spatial reasoning, as shown in Fig.8 (a). This leads to a performance drop of approximately 105% in terms of MAE, as shown in Table 2.
Role of Learnable CSL Tokens (Δ CSL). Eliminating the CSL prompts significantly impairs the model's ability to
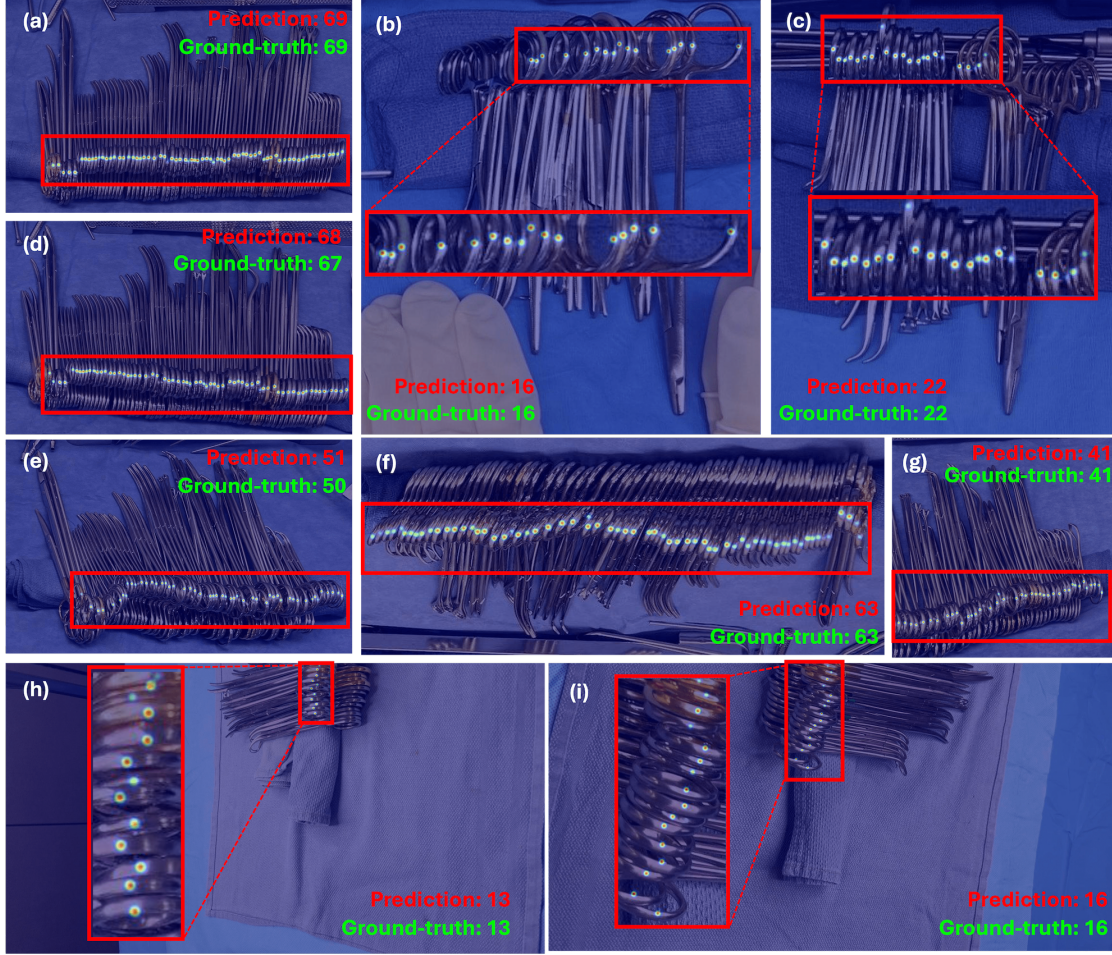
Figure 5. **Qualitative results.** We present qualitative results from our CoLSR. Predicted surgical instruments number and ground-truth number are listed on each image. The detected surgical instrument handles are highlighted with laser points, which are also highlighted with red bounding boxes.
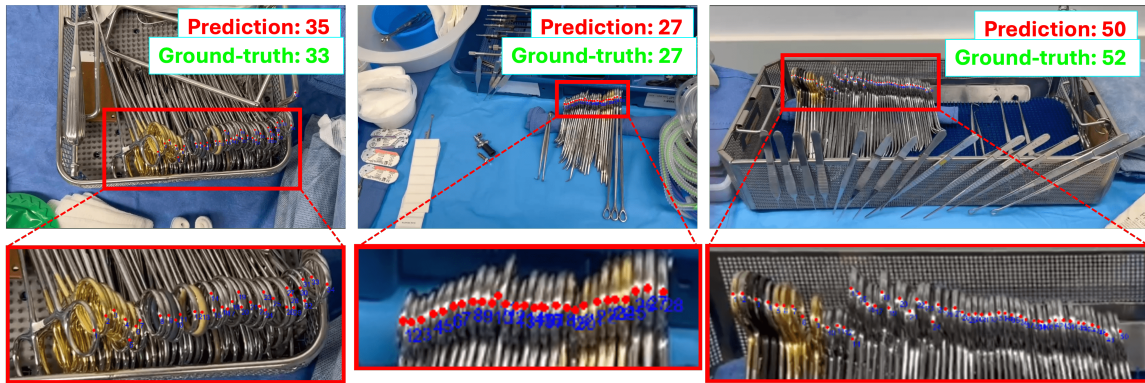

Figure 6. **Generalization ability analysis.** We evaluate our model's generalization ability via in the wild images in operating rooms. The detected surgical instrument handles are highlighted with laser points, which are also highlighted with red bounding boxes.

detect fine-grained handle boundaries (further discussed in the supplementary material). This is further exacerbated when instruments are densely packed, causing the handle boundaries to appear merged as highlighted in Figure. 8 (b). Consequently, the model's performance degrades by approximately 133% in terms of MAE, as shown in Table 2.
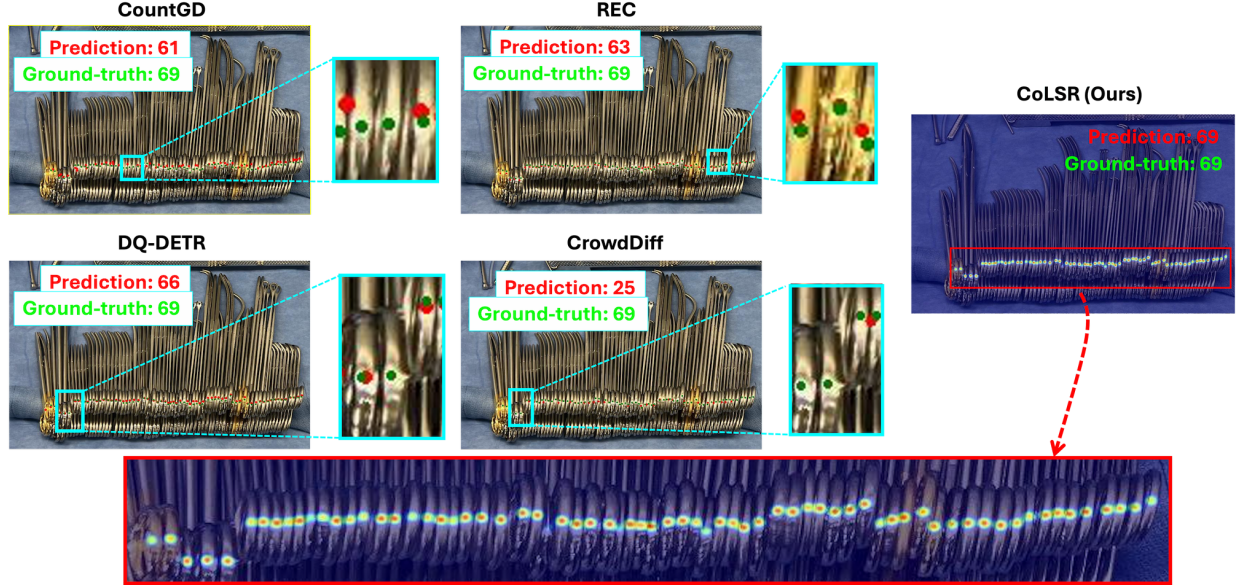
Figure 7. **Comparison with SOTA methods.** Our CoLSR approach is compared with four existing SOTA methods for counting: CountGD, DQ-DETR, CrowdDiff and REC. For the four figures on the left side, green dots represent ground-truth, red dots represent predictions from different models.
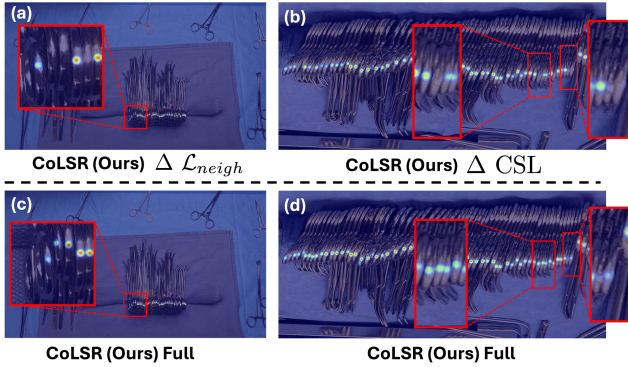

Figure 8. **Ablation Studies.** (a) The highlighted region shows where the model failed to make correct predictions, indicating the model's limited ability to form coherent visual chains. (b) Missed handles are mostly in areas with unclear boundary separation, making them harder to detect without class-specific learnable prompts. (c, d) Compared with the ablated results in (a) and (b), CoLSR effectively generates accurate predictions for the location of tightly packed surgical instrument handles.

Pure Zero-shot training and inference (Δ Visual Exemplars). As shown in Table 2, training and evaluation in a purely zero-shot setting without visual exemplars leads to a performance drop of approximately 70% in terms of MAE.

Role of postprocessing (Δ Post). We remove postprocessing during inference and found slight drop of both MAE and RMSE in Table 2.

More ablation studies. In supplementary, we show additional ablation studies including multi-loss weight selection, CSL token placement, among other variations.

5.6. Failure Analysis

In the supplementary material, we present representative failure cases. Most errors arise from the dense and visually ambiguous appearance of instruments in the images. Empirical results indicate that performance degrades in scenarios where gaps or occlusions disrupt the continuity of surgical instruments, making spatial reasoning more challenging. Potential solutions include leveraging multi-view inputs (e.g., short video sequences capturing multiple viewpoints) or incorporating depth information to better handle severe occlusions. As future work, we plan to integrate such diverse visual inputs into the counting pipeline to further enhance the robustness and reliability of our approach.

6. Conclusion

We introduce Chain-of-Look spatial reasoning framework that is inspired by human sequential counting behavior, designed to improve accuracy in densely packed surgical instrument scenes. By enforcing a structured visual chain and introducing a neighboring loss to model spatial constraints, our method outperforms existing SOTA counting models as well as multimodality large language models. This framework offers a generalizable approach that can be extended to broader applications requiring spatial reasoning in dense visual environments. Additionally, we introduce SurgCount-HD, a high-density surgical instrument dataset to facilitate benchmarking and drive future research in this domain.

7. Acknowledgment

References

[1] Niki Amini-Naieni, Kiana Amini-Naieni, Tengda Han, and Andrew Zisserman. Open-world text-specified object counting, 2023. 3

[2] Niki Amini-Naieni, Tengda Han, and Andrew Zisserman. Countgd: Multi-modal open-world counting. *Advances in Neural Information Processing Systems*, 37:48810–48837, 2024. 2, 3, 4, 5, 6, 1

[3] Carlos Arteta, Victor Lempitsky, J. Alison Noble, and Andrew Zisserman. Interactive object counting. In *Computer Vision – ECCV 2014*, pages 504–518, Cham, 2014. Springer International Publishing. 3

[4] Carlos Arteta, Victor Lempitsky, and Andrew Zisserman. Counting in the wild. In *Computer Vision – ECCV 2016*, pages 483–498, Cham, 2016. Springer International Publishing. 3

[5] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 6

[6] Siu-Yeung Cho, T.W.S. Chow, and Chi-Tat Leung. A neural-based crowd estimation by hybrid global learning algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29(4):535–541, 1999. 3

[7] 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. 3, 6, 5

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019. 4

[9] Nikola Djukic, Alan Lukezic, Vitjan Zavrtanik, and Matej Kristan. A low-shot object counting network with iterative prototype adaptation, 2023. 3

[10] Rochel Gelman, Charles R Gallistel, and Rochel Gelman. *The child's understanding of number*. Harvard University Press, 2009. 2

[11] 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. 4

[12] Mingyue Guo, Li Yuan, Zhaoyi Yan, Binghui Chen, Yaowei Wang, and Qixiang Ye. Regressor-segmenter mutual prompt learning for crowd counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28380–28389, 2024. 3

[13] https://openai.com/gpt5/. Gpt5, 2025. 3, 6

[14] 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

[15] Yi-Xin Huang, Hou-I Liu, Hong-Han Shuai, and Wen-Huang Cheng. Dq-detr: Detr with dynamic query for tiny object detection, 2024. 3, 6, 5

[16] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning, 2022. 3

[17] Seunggu Kang, WonJun Moon, Euiyeon Kim, and Jae-Pil Heo. Vlcounter: Text-aware visual representation for zero-shot object counting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2714–2722, 2024. 3

[18] D. Kong, D. Gray, and Hai Tao. A viewpoint invariant approach for crowd counting. In *18th International Conference on Pattern Recognition (ICPR'06)*, pages 1187–1190, 2006. 3

[19] Victor Lempitsky and Andrew Zisserman. Learning to count objects in images. In *Proceedings of the 24th International Conference on Neural Information Processing Systems - Volume 1*, page 1324–1332, Red Hook, NY, USA, 2010. Curran Associates Inc. 3

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

[21] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2018. 3

[22] Chang Liu, Yujie Zhong, Andrew Zisserman, and Weidi Xie. Countr: Transformer-based generalised visual counting, 2023. 3

[23] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 4

[24] Gordon Logan, Jane Zbrodoff, and Xingshan Li. Do the eyes count? the role of eye movements in visual enumeration. *Journal of Vision*, 8(6):115–115, 2008. 2

[25] A.N. Marana, S.A. Velastin, L.F. Costa, and R.A. Lotufo. Estimation of crowd density using image processing. In *IEE Colloquium on Image Processing for Security Applications (Digest No: 1997/074)*, pages 11/1–11/8, 1997. 3

[26] Jer Pelhan, Alan Lukežič, Vitjan Zavrtanik, and Matej Kristan. Dave – a detect-and-verify paradigm for low-shot counting, 2024. 3

[27] Zhuoxuan Peng and S. H. Gary Chan. Single domain generalization for crowd counting, 2024. 3

[28] Yasiru Ranasinghe, Nithin Gopalakrishnan Nair, Wele Gedara Chaminda Bandara, and Vishal M Patel. Crowd-diff: Multi-hypothesis crowd density estimation using diffusion models. In *Proceedings of the IEEE/CVF Conference*

on Computer Vision and Pattern Recognition, pages 12809–12819, 2024. 6

[29] Roboflow Team. Roboflow: Computer vision platform. https://roboflow.com, 2025. 3

[30] Zhicheng Wang, Zhiyu Pan, Zhan Peng, Jian Cheng, Liwen Xiao, Wei Jiang, and Zhiguo Cao. Exploring contextual attribute density in referring expression counting. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 19587–19596, 2025. 3

[31] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022. 3

[32] Nan Xi, Jingjing Meng, and Junsong Yuan. Chain-of-look prompting for verb-centric surgical triplet recognition in endoscopic videos. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 5007–5016, 2023.

[33] Nan Xi, Jingjing Meng, and Junsong Yuan. Open set video hoi detection from action-centric chain-of-look prompting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3079–3089, 2023. 3

[34] Weidi Xie, J. Alison Noble, and Andrew Zisserman. Microscopy cell counting and detection with fully convolutional regression networks. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, 6(3): 283–292, 2018. 3

[35] Hantao Yao, Rui Zhang, Lu Yu, Yongdong Zhang, and Changsheng Xu. Sep: Self-enhanced prompt tuning for visual-language model, 2024. 3

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

Supplementary Material

8. 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 9 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.

9. 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 10th 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 [2] configuration.

10. 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. 10). 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:

$$\mathbf{F}_{encoder} = [\mathbf{T}_{\text{CSL}} \,;\, \mathbf{T}_{\text{text}} \,;\, \mathbf{T}_{\text{vis}}] \in \mathbb{R}^{(l+2h) \times d}, \quad (6)$$

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.

$$\mathbf{F}_{decoder} = [\mathbf{T}_{\text{text}} \,;\, \mathbf{T}_{\text{vis}}] \in \mathbb{R}^{(l+h) \times d} \quad (7)$$

10.1. 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 $\alpha = 0.25, \gamma = 2$:

$$v(i,k) = \underbrace{\|\mathbf{b}_i - \mathbf{b}_k\|_1}_{\text{bbox cost}} + \underbrace{\sum_{j=1}^{C} \tilde{y}_{kj} \left[\mathcal{L}_{\text{pos}}(p_{ij}) - \mathcal{L}_{\text{neg}}(p_{ij})\right]}_{\text{cls cost}},$$

$$\text{s.t.} \quad \mathcal{L}_{\text{pos}}(p) = -\alpha(1-p)^{\gamma} \log(p + \varepsilon),$$
$$\mathcal{L}_{\text{neg}}(p) = -(1-\alpha)p^{\gamma} \log(1 - p + \varepsilon),$$

$$(8)$$

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.

11. 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 11 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.
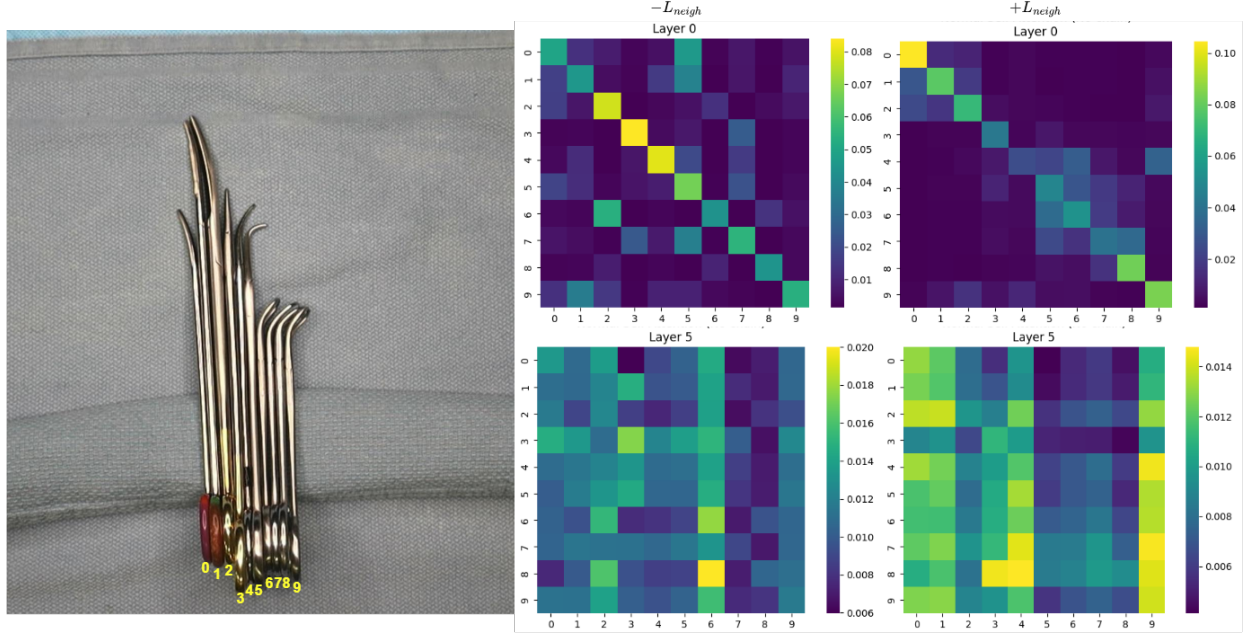
Figure 9. **Analysis on Chain-of-Look Visual Reasoning via Spatial Neighboring Loss.** Left: original surgical image. Right: attention maps from different decoder layers. "$-L_{\text{neigh}}$" denotes models trained without the Neighboring Loss, whereas "$+L_{\text{neigh}}$" indicates models trained 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.
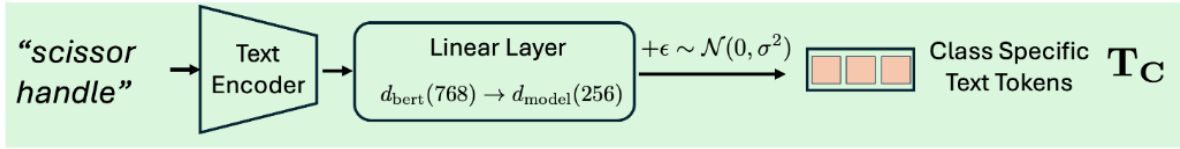

Figure 10. CSL Prompts Initialization with BERT Text Encoder

12. Extended Ablation Results

12.1. CSL Prompts Placement: Appending vs. Prepending

Previous studies [20] have highlighted how prompt placement affect transformer models. Our analysis (Table. 3) 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.

12.1.1. 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-

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 3. **Prompt Placement** Performance comparison across CSL prompt placements. See Section 12.1.1 for metric definitions.

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 av-
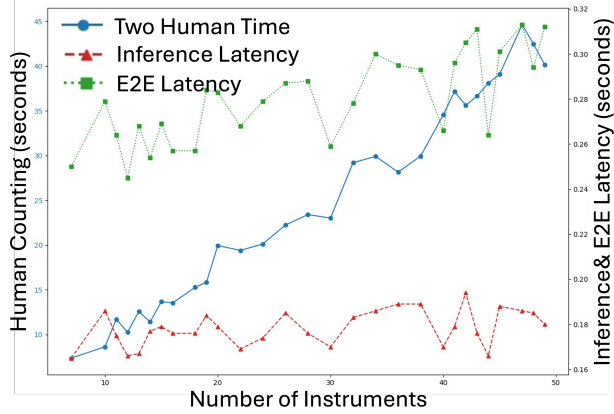
Figure 11. Time comparison between human counting and our model

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

Table 4. **Prompt Initialization Strategy** Prepending task-specific initialized CSL prompts yields better performance compared to random initialization.

erages 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 4 shows that using task-specific initialization leads to notable performance improvements.

12.2. 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 [14] 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.

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

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

$(\lambda_{cls}, \lambda_{loc}, \lambda_{neigh})$	$\mathcal{L}_{cls}$ **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 6. **Gradient Magnitude Analysis** Multi-Loss scaling factor selection.

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. 12). 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 13) 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.

12.3. 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. 6 there exists a high imbalance in gradient magnitudes between the cross-entropy (CE) loss 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, en-
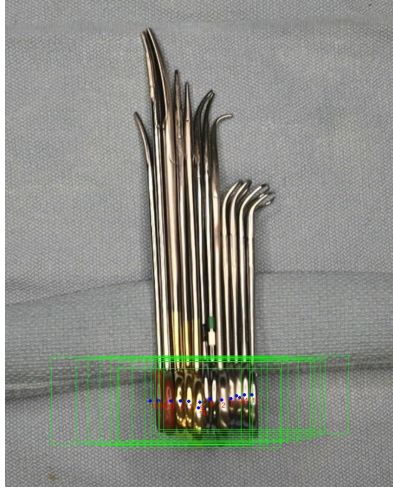
Figure 12. Predicted bounding boxes using the LoRA method. Boxes are noticeably oversized and misaligned.

abling more expressive multi-objective optimization.

13. 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. 13(b–c). In contrast, CSL tokens learn contrastive features, as shown in Fig. 13(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.

14. 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.

$$1 - \text{int}\left(\left(P_x^{\max} - P_x^{\min}\right) > \left(P_y^{\max} - P_y^{\min}\right)\right)$$
$$= \begin{cases} 0 & \text{for } x\text{-axis} \\ 1 & \text{for } y\text{-axis}, \end{cases} \quad (9)$$

where:

$$P_{\text{pred}} = \{(x_i, y_i) \mid \text{point } i \text{ is predicted}\},$$
$$P_x = \{x_i \mid (x_i, y_i) \in P_{\text{predicted}}\} \quad \text{with } P_x \subseteq [0, W],$$
$$P_y = \{y_i \mid (x_i, y_i) \in P_{\text{predicted}}\} \quad \text{with } P_y \subseteq [0, H],$$
$$(10)$$

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

15. Evaluation Metrics

15.1. Counting Metrics

15.1.1. MAE, RMSE

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

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} |N_P - N_G|,$$
$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (N_P - N_G)^2}, \quad (11)$$

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.

15.1.2. Grid Average Mean Absolute Error

We also measure the Grid Average Mean Absolute Error (GAME) [11] 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. 7). 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.
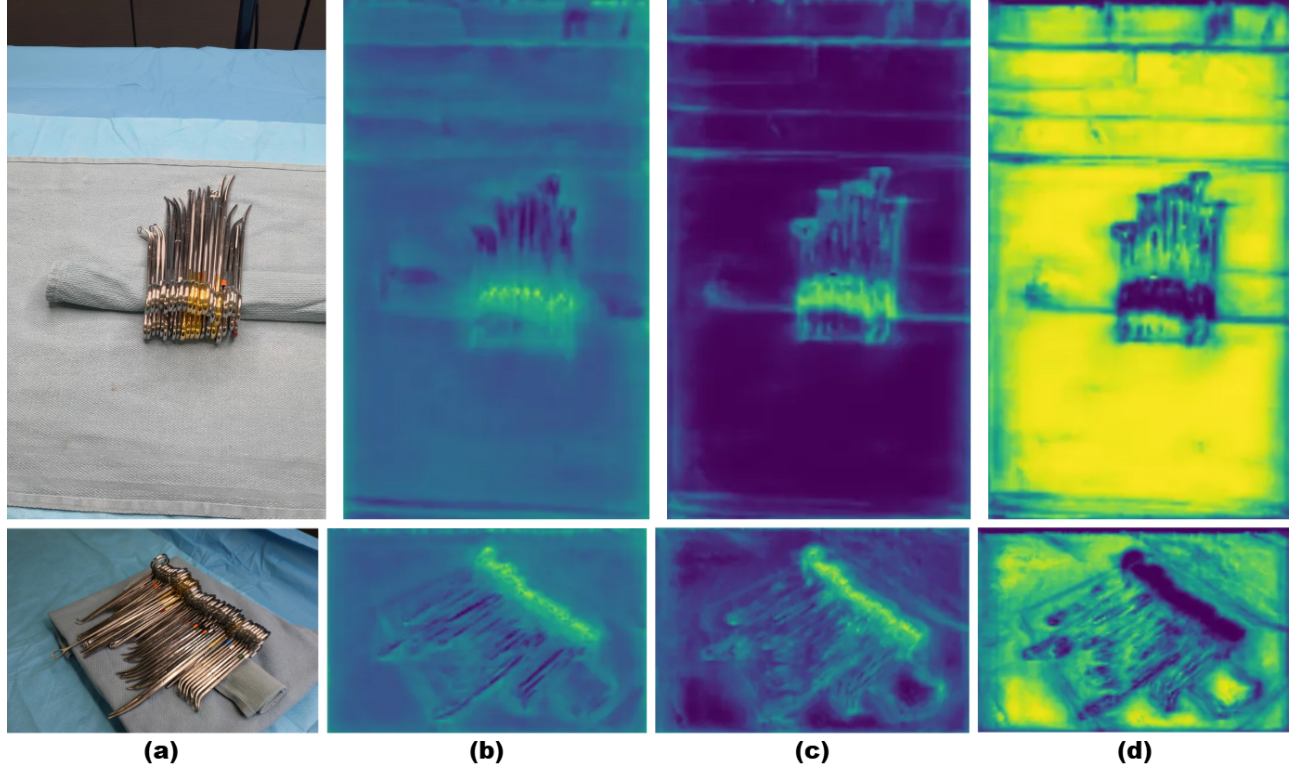
Figure 13. **a)** Original Input Image **b)** Image-Text Attention Map extracted from the Feature Fusion Block - Without CSL Prompts **c)** Image-Text Attention Map when trained with CSL Prompts **d)** Image-CSL Token Attention Map

Moreover, we also use detection-related counting metrics such as precision, recall and F1-score as defined in Equation. 12.

Method	GAME-L1 ↓	GAME-L2 ↓	GAME-L3 ↓
CountGD [2]	1.01	0.41	0.14
REC [7]	0.60	0.25	0.08
DQ-DETR [15]	0.68	0.25	0.07
CoLSR (Ours)	**0.54**	**0.23**	**0.07**

Table 7. GAME scores (L1, L2, L3) for different methods.

15.2. 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. 8). A similar procedure is applied for computing the Mean IoU reported in the Table. 5.

Steps for a single input :

$$P_{\text{filtered}} = p \in P_{\text{pred}} \mid \exists b \in B_{GT} \text{ such that } p \in b$$

$$M^* = \underset{M}{\arg\min} \sum_{(p,g) \in M} |p - g|_2$$

$$d_i = |p_i - g_i|_2$$

$$\bar{d} = \frac{1}{N} \sum_{i=1}^{N} d_i \qquad (12)$$

$$\text{Median Error} = \text{median}(d_1, d_2, \ldots, d_N)$$

$$\text{95th Percentile Error} = P_{95}(d_1, d_2, \ldots, d_N)$$

$$\text{True Positive (TP)} = N$$

$$\text{False Positive} = len(P_{\text{pred}}) - TP$$
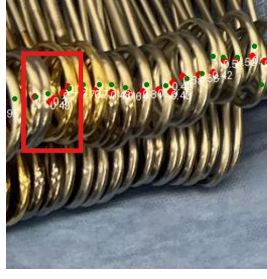
$$\text{False Negative} = len(G_{GT}) - TP$$

Figure 14. Example of duplicate points highlighted

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.

16. 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. 14). 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 14). 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

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

---

17. 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.15-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: 2,3). 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 14).

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.15-b).

	CountGD [2]	REC [7]	DQ-DETR [15]	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 8. Comparison of localization metrics results across different methods.

18. Limitations of Generalization

While our method demonstrates robustness to variations in angle and lighting conditions typical of operating room (OR) environments (Figure. 16, 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.
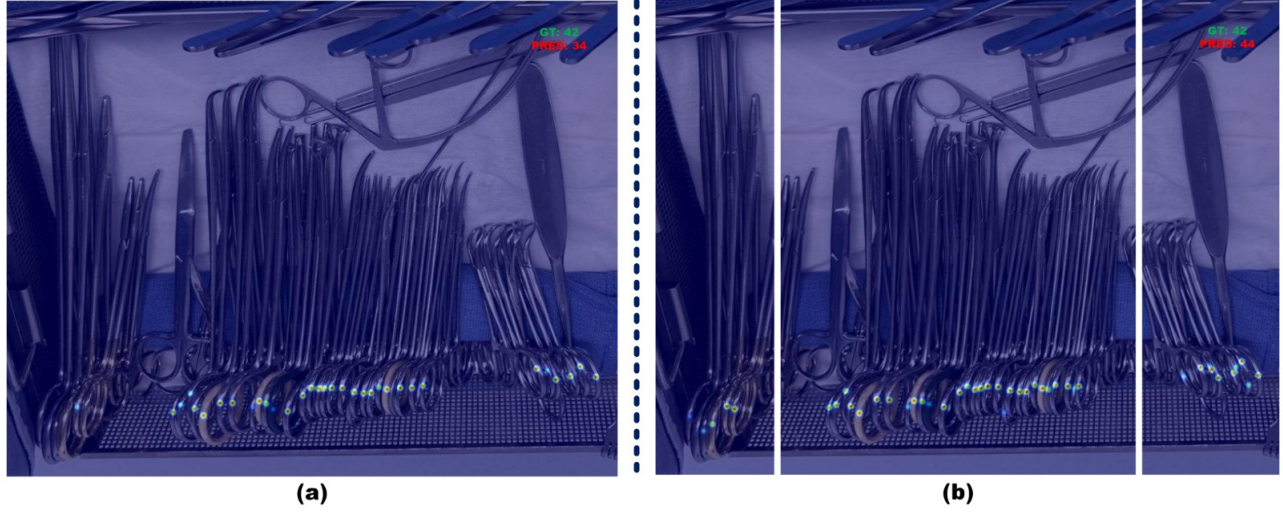
Figure 15. **a)** Prediction with single-pass inference **b)** Prediction with Divide-and-Conquer approach

---

Algorithm 2: Distance-Based Clustering

1: $clusters \leftarrow \{\}$
2: $cluster \leftarrow [0]$ {cluster start}
3: **for** $i = 0$ to $|pred\_points| - 2$ **do**
4:    $p_i \leftarrow pred\_points[i]$
5:    $p_{i+1} \leftarrow pred\_points[i+1]$
6:    **if** $\|p_i - p_{i+1}\|_2 > \delta$ **then**
7:      $cluster$.append(i) {cluster end}
8:      $clusters$.append($cluster$)
9:      $cluster \leftarrow [i+1]$ {next cluster start}
10:    **end if**
11: **end for**
12: $slices \leftarrow$ slice_image($clusters$)
13: **return** $slices$

Algorithm 3: Two-Pass Counting

1: $pred\_points \leftarrow$ run_inference($image$) {first pass}
2: $slices \leftarrow$ create_cluster($pred\_points, \delta$)
3: $final\_detections \leftarrow \{\}$
4: **for** $slice \in slices$ **do**
5:    $pred\_points \leftarrow$ run_inference($slice$) {second pass}
6:    $final\_detections$.append($pred\_points$)
7: **end for**
8: **return** $final\_detections$

Figure 16. **Robust inference samples captured from multiple angles.**