Technical Report of Team xmu-mac-automl for the FSOD Challenge 2026

Zhigang Chen, Xiawu Zheng, Rongrong Ji
Key Laboratory of Multimedia Trusted Perception and Efficient Computing,
Ministry of Education of China, Xiamen University, 361005, P.R. China

Abstract

We present an in-context pipeline for few-shot object detection on RF-20VL. The method centers on multimodal large language models (MLLMs) and SAM3 and organizes inference into four stages: offline text prompt selection, support-driven visual prompt box acquisition, joint text-visual inference, and VQA-based re-ranking. For each category, we first search for a compact textual concept that better matches the lexical space of the segmentation model. We then transfer annotated support-side geometry to the query image through support-query concatenation, producing candidate boxes. The selected text prompt and the candidate boxes are passed to SAM3 jointly, and the resulting predictions are calibrated by an MLLM through a binary verification step. This design keeps the pipeline modular and avoids task-specific training.

1. Introduction

We consider the few-shot object detection task defined by the RF-20VL benchmark. Each subset provides a small support set with annotated images and a query set of unlabeled test images. Given a query image and a target category, the goal is to localize all instances of that category in the query image. Our system does not train a separate detector; instead, it uses SAM3 [2] as the core inference engine and adapts it through text and geometric prompts.

2. Preliminaries

MLLM. An MLLM is a generative model that jointly processes visual inputs and natural language, enabling unified perception and language reasoning. In this work, the MLLM is used offline to generate compact category prompts from category names and auxiliary class-level descriptions. Formally, given an image (or a set of images) and a textual instruction prompt, an MLLM produces a textual response as

$$T = \mathcal{M}_{\text{MLLM}}(I, P)\,, \tag{1}$$

where $\mathbf{I}$ denotes the visual input, $\mathbf{P}$ denotes the input prompt, and $\mathbf{T}$ is the generated text output.

SAM3. SAM3 is a foundation model that performs promptable concept segmentation. The target can be specified by a short noun phrase (textual concept), a visual exemplar bounding box, or both. Given a concept prompt, SAM3 predicts instance masks for all instances matching the concept. Formally, we write SAM3 inference as

$$M^{\text{ins}} = \mathcal{M}_{\text{SAM3}}(I,\, T,\, V)\,, \tag{2}$$

where I is the image, T is a textual concept prompt, and V denotes the visual exemplar boxes.

3. Method

Our pipeline consists of four stages, as illustrated in Fig. 1. First, we search for an effective textual concept for each category. Second, we generate query-side candidate boxes as visual exemplars by transferring support annotations through support-query image concatenation. Third, we run SAM3 with both the selected text prompt and the candidate boxes. Fourth, we re-rank the resulting predictions with a binary VQA verification step.

3.1. Text Prompt Extraction

The textual concept provided by a dataset is not always the most effective prompt for SAM3. A category name can be overly broad, domain-specific, or lexically different from the concept names favored by the segmentation model. We therefore perform an offline prompt search for each category and use the selected prompt throughout inference.

Candidate generation. For each category c, we construct an instruction prompt from its category name and a precomputed class-level description from DetPO [3]. The MLLM is asked to generate a small set of short English noun phrases that can refer to the same visual concept:

$$\mathcal{P}_c = \{p_c^k\}_{k=1}^{K}, \tag{3}$$

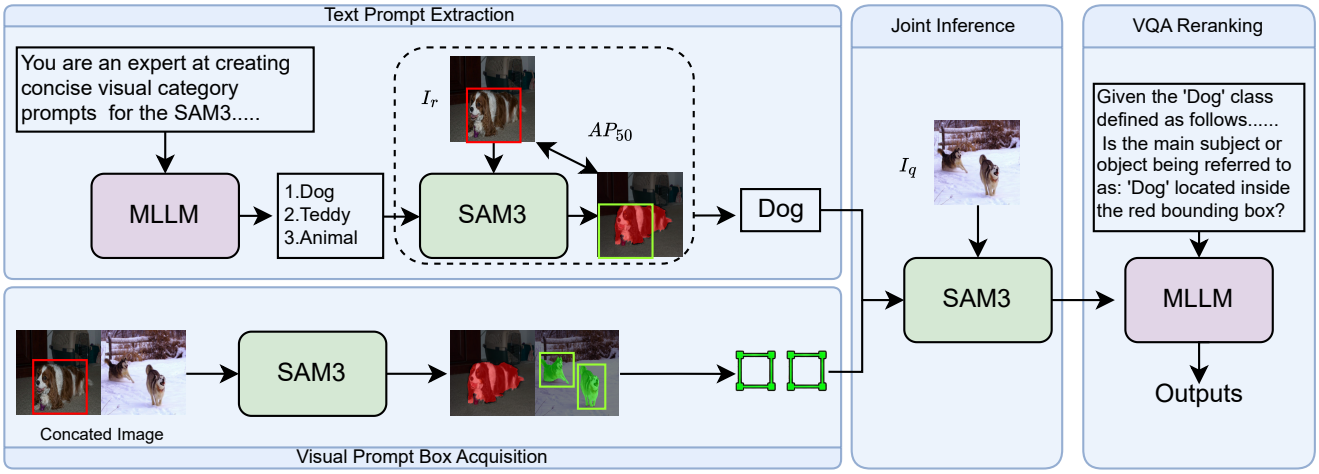

Figure 1. Overall framework of the proposed pipeline.

where $K = 5$ in our implementation and each candidate is constrained to one to three words. This step provides multiple lexical variants for the same target category while keeping the final SAM3 text prompt compact.

Candidate scoring. Each candidate prompt is evaluated on the support images of the corresponding category. Specifically, we run SAM3 in a text-only setting, where the visual exemplar input is omitted:

$$M_{i,k}^{\text{ins}} = \mathcal{M}_{\text{SAM3}}\big(I_i,\, p_c^k,\, \varnothing\big)\,, \qquad I_i \in \mathcal{S}_c, \tag{4}$$

where $\mathcal{S}_c$ denotes the support images for category c. The predictions from all support images are collected and evaluated with COCO API. We use AP@0.5 as the prompt score:

$$s_c(p_c^k) = \text{AP}_{50}\big(\{M_{i,k}^{\text{ins}}\}_{I_i \in \mathcal{S}_c},\, \mathcal{G}_c\big)\,, \tag{5}$$

where $\mathcal{G}_c$ denotes the support annotations for category c.

Prompt selection and lookup. The final text prompt for category c is selected by maximizing the support-set score:

$$p_c^\star = \arg\max_{p \in \mathcal{P}_c} s_c(p). \tag{6}$$

We store the candidate prompts and their evaluation statistics in a per-subset prompt mapping file. During inference, the pipeline loads this mapping and uses $p_c^\star$ as the SAM3 textual concept. If a category is missing from the mapping, we fall back to the original category name.

3.2. Visual Prompt Box Acquisition

To generate candidate boxes for a query image, we reuse the support images of the same category as geometric prompts. For each query sample I_q, we iterate over all support images $I_s \in \mathcal{S}_c$ and form a horizontal concatenation after resizing the support image to match the query height while preserving the aspect ratio:

$$\tilde{I}_{s,q} = \text{Concat}(\text{Resize}(I_s; h_q), I_q)\,, \tag{7}$$

where h_q is the height of the query image. The ground-truth support box is transformed accordingly, converted to normalized $cxcywh$ format, and injected into SAM3 as the visual prompt. Running SAM3 on $\tilde{I}_{s,q}$ yields a set of candidate detections on the concatenated canvas.

We keep only the detections whose centers fall on the query side. After aggregating all boxes over all support images of the same category, we apply NMS with an IoU threshold of 0.5 to obtain the final candidate box set.

This support-query construction lets SAM3 transfer the annotated support location to the query side through visual analogy on the shared canvas.

3.3. Joint Text-Visual Inference

We feed the selected text prompt and the candidate box set into SAM3 jointly for the final detection pass:

$$\mathcal{D}_c^{\text{tv}} = \mathcal{M}_{\text{SAM3}}\big(I_q,\, p_c^\star,\, \mathcal{B}_c^{\text{cand}}\big)\,, \tag{8}$$

where $\mathcal{D}_c^{\text{tv}} = \{(b_j, s_j)\}_{j=1}^{N_c}$ denotes the predicted boxes and their confidence scores for category c. Each candidate box is treated as a positive geometric prompt, and the query image is encoded only once. The resulting predictions are then filtered again with NMS at an IoU threshold of 0.5 to remove duplicate detections.

3.4. VQA Re-ranking

The VQA re-ranking stage uses an MLLM, specifically Qwen3-VL-8B-Instruct [1], and follows the binary verification scheme of DetPO [3]. We also reuse the class-level

category descriptions from DetPO when forming the verification prompt. For each predicted box, we draw a red rectangle on the query image and ask a binary verification question of the form: whether the object referred to by the category is located inside the highlighted region. The model returns token probabilities for affirmative and negative answers, which we normalize into a verification score:

$$r_j = \frac{p_j(\text{Yes})}{p_j(\text{Yes}) + p_j(\text{No})}. \tag{9}$$

4. Conclusion

We described an in-context FSOD pipeline built on MLLM and SAM3. The method combines prompt selection, support-query box transfer, joint text-visual inference, and VQA-based re-ranking into a single modular system. Its main advantage is that both the textual concept and the geometric candidates are derived directly from the support set at inference time, without additional task-specific training.

References

[1] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025. 2

[2] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts, 2025. 1

[3] Gautam Rajendrakumar Gare, Neehar Peri, Matvei Popov, Shruti Jain, John Galeotti, and Deva Ramanan. Detpo: In-context learning with multi-modal llms for few-shot object detection, 2026. 1, 2