Supplementary Material for EGVLR: Evidence-Grounded Vision–Language Reinforcement for Anomaly Reasoning

Shih-Chih Lin[1], Ying-Heng Lu[1], Dong You Ye[1], and Shang-Hong Lai[1]

National Tsing Hua University, Hsinchu, Taiwan

Supplementary Overview

This document provides implementation details for EGVLR. The main paper focuses on MMAD industrial anomaly reasoning, while the supplementary material describes the full prompt protocol, visual pre-alignment data construction, programmatic rationale generation, data mixture, reward implementation, and box-guided mask rendering. All trainable stages use the same EDDP schema:

```
<think>
<evidence>...</evidence>
<logic>...</logic>
</think>
<location>...</location>
<answer>...</answer>
```

The location field is either a list of EDDP normalized boxes in the 0–1000 coordinate system or an empty list `[]`.

1 Implementation Details

Backbone and input order. Unless otherwise specified, EGVLR is instantiated with Qwen3-VL-8B-Instruct. All paired-image samples follow the query-first and normal-reference-second order. Images are resized by the backbone processor, while all generated and parsed boxes are represented in EDDP normalized 0–1000 coordinates.

Training stages. PVE-FT and KG-IT are optimized with autoregressive supervised fine-tuning. GS-DPO is initialized from the KG-IT checkpoint and uses group-sampled completions with the reward defined in the main paper. We keep the reward parser, answer parser, and location parser fixed across all ablations.

Training data and leakage control. Following recent industrial anomaly reasoning practice [1], any MMAD images used for auxiliary training are excluded from MMAD evaluation to avoid data leakage. We also include a small amount of Real-IAD visual data in Stage-I visual pre-alignment to expose the model to real industrial defect appearances [1, 2]. These Real-IAD samples are used only for visual-evidential grounding and are not part of the MMAD evaluation set.

Table S1: Default implementation settings used in the reported EGVLR runs.

Item	Setting
Backbone	Qwen3-VL-8B-Instruct
Input order	query/test image first, normal reference second
Output schema	EDDP: `<evidence>`, `<logic>`, `<location>`, `<answer>`
Parser	deterministic parser shared by training, ablation, and evaluation
Box format	EDDP normalized coordinates, 0–1000
Stage-I data	synthetic visual-evidential QA plus small Real-IAD visual pre-alignment samples
Stage-II mixture	Domain QA / one-normal visual QA / comparative QA
Stage-III objective	GRPO-style group preference optimization with fixed parser rewards
Semantic scorer	sentence-embedding-based semantic similarity over reference sentences
Spatial matching	IoU-based Hungarian matching with count penalty
Segmentation rendering	external segmentation backend, visualization only

Reward and rendering settings. Reward coefficients are selected as a fixed default configuration for the reported runs and are not tuned separately for each ablation or test setting. BGSR uses an external segmentation backend only for visualization; it does not modify model answers or QA accuracy.

Parser handling. Malformed outputs are not manually corrected. Missing tags, invalid answer tokens, invalid coordinates, or non-positive-area boxes are handled by the same deterministic parser and reward rules used in all ablations. This avoids introducing post-hoc repair rules that could differ between methods.

2 Stage-I PVE-FT Details

2.1 Grid Convention

For all Stage-I visual grounding questions, the grid is defined as follows:

> **Grid convention:**
> Divide only the first image, i.e., the query/test image, into a 3×3 grid over the full displayed image canvas.
>
> **Grid layout:**
> 1 = top-left, 2 = top-center, 3 = top-right
> 4 = middle-left, 5 = center, 6 = middle-right
> 7 = bottom-left, 8 = bottom-center, 9 = bottom-right
>
> **Coordinate system:**
> Use the viewer's image coordinate system: top means the upper part of the displayed image, bottom means the lower part, left means the left side, and right means the right side.
>
> **Bounding boxes:**
> EDDP boxes use normalized coordinates from 0 to 1000: x increases from left to right, and y increases from top to bottom.

The second image, when provided, is only a normal reference and is not divided into grids.

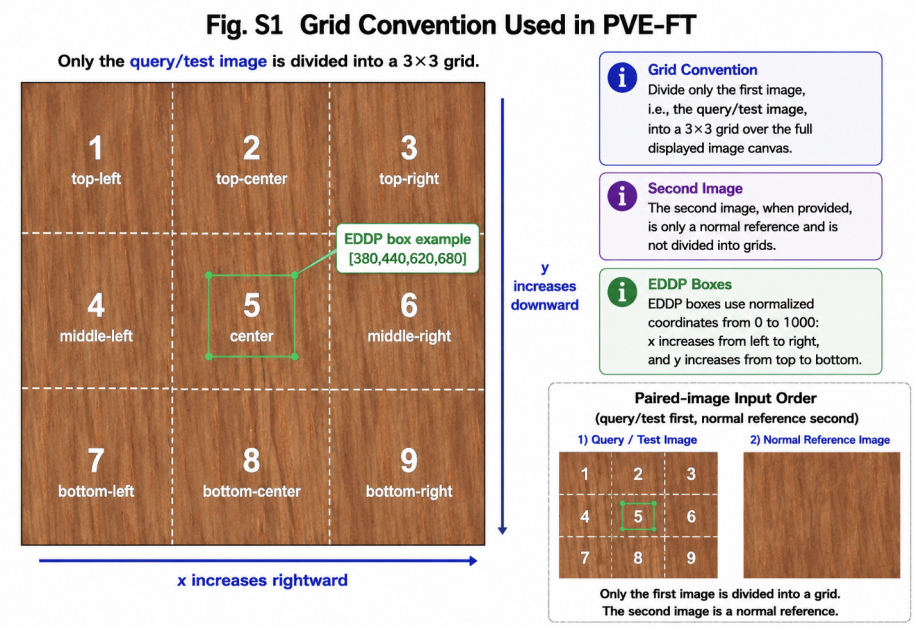

Fig. S1: Grid convention used in PVE-FT. Only the query/test image is divided into a 3×3 grid. Grid IDs are used only in prompts and rationales; the `<location>` field stores EDDP normalized boxes in the range 0–1000 or an empty list.

2.2 Synthetic Anomaly Generation

Stage-I PVE-FT uses normal industrial images to create query images with known spatial metadata. For each normal image, we randomly sample a same-category normal reference. A localized anomaly is then generated using either CutPaste-style patch replacement or DTD texture insertion. The generator records the anomaly mask, pixel-space box, EDDP normalized 0–1000 box, grid ID, and optional subregion. This metadata is used to build the question, the answer option, the location target, and the programmatic rationale. In addition, a small amount of Real-IAD visual data is used for Stage-I visual pre-alignment when available, following the same EDDP schema and excluding any MMAD training images from MMAD evaluation.

2.3 Stage-I Visual QA Families

PVE-FT contains five families:

1. **F1 Synthetic anomaly detection**: decide whether the query image contains a visible localized anomaly compared with the reference.
2. **F2 3×3 grid localization**: identify which grid contains the main abnormal region.

Algorithm 1: Stage-I Visual Evidence Data Construction

Input: Normal image pool $\mathcal{I}$; DTD texture pool $\mathcal{T}$; output root.
Output: Stage-I EDDP-format QA samples.

foreach *object category* c **do**
  collect normal images $\mathcal{I}_c$;
  foreach *sampled base image* $I \in \mathcal{I}_c$ **do**
    sample reference $I_{ref} \in \mathcal{I}_c$, $I_{ref} \neq I$;
    sample anomaly method $m \in \{\text{CutPaste}, \text{DTD}\}$;
    synthesize query image $\tilde{I}$ and mask M;
    compute normalized box $B^\star$ and grid ID $g^\star$;
    create F1 anomaly detection sample;
    create F2 grid localization sample;
    optionally create F3 subregion localization sample;
    create F4 positive local verification sample;
    create F4 decoy local verification sample with empty location;
    create F5 normal-reference null-calibration sample;

3. **F3 Fine subregion localization**: identify the subregion inside the selected grid cell.
4. **F4 Local verification with decoys**: focus on a specified grid and decide whether it truly contains a defect.
5. **F5 Null-hypothesis calibration**: compare two normal images and output an empty location.

2.4 Programmatic Visual Rationales

Because Stage-I anomalies are generated programmatically, their rationales are also generated from controlled metadata rather than from external LLMs. The rationale only verbalizes visible spatial evidence. It should not mention the synthetic generation method in the target response, because real defects at evaluation time are not synthetic. For example, the target should say "a localized abnormal patch" or "a localized texture inconsistency," not "a CutPaste anomaly."

Abnormal example.

```
<think>
<evidence>A localized abnormal patch is visible in Grid 5 (center) of
the query image.</evidence>
<logic>The patch lies in the center cell of the 3x3 grid, so the answer
should select Grid 5 and the bbox should localize that patch.</logic>
</think>
<location>[[430,410,575,545]]</location>
<answer>B</answer>
```

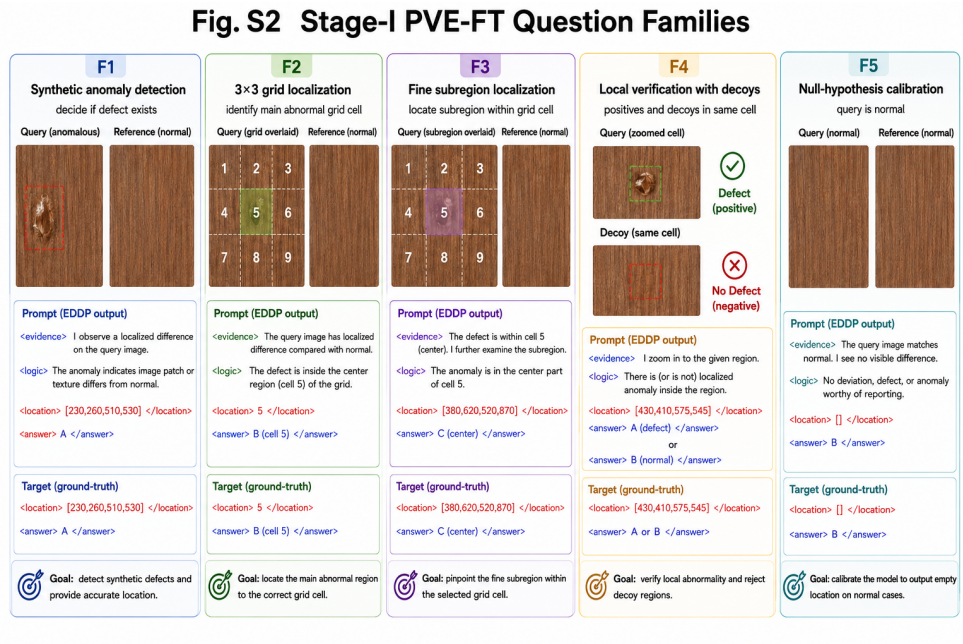

Fig. S2: Stage-I PVE-FT question families. The five families cover synthetic anomaly detection, 3×3 grid localization, fine subregion localization, local verification with positive/decoy regions, and normal-reference null-hypothesis calibration under the same EDDP schema.

Normal example.

```
<think>
<evidence>The query image is visually consistent with the normal reference
and shows no localized abnormal cue in any 3x3 grid region.</evidence>
<logic>Since no valid spatial defect evidence is visible, no bbox should
be reported and the null hypothesis should be retained.</logic>
</think>
<location>[]</location>
<answer>B</answer>
```

Local verification decoy example.

```
<think>
<evidence>The focused Grid 3 (top-right) shows normal appearance and
contains no localized abnormal cue.</evidence>
<logic>A defect should only be reported when localized evidence is present
inside the queried region.</logic>
</think>
<location>[]</location>
<answer>B</answer>
```

Table S2: Stage-II KG-IT data sources.

Source	Role
Domain QA	object and defect-domain knowledge
MMAD one-normal visual QA	query-reference anomaly QA behavior
Comparative inspection QA	comparative reasoning and hard-negative calibration

3 Stage-II KG-IT Data Mixture

KG-IT uses a mixture of domain-oriented and visual-comparative instruction data. In our MMAD instantiation, the mixture contains three complementary sources: domain QA, MMAD one-normal visual QA, and comparative inspection QA. These sources are combined to balance object-level domain knowledge, query-reference anomaly reasoning, and hard-negative comparative calibration.

For samples without reliable box supervision, we keep the `<location>` field empty, i.e., `[]`, but set `require_location=false` in metadata. Therefore, such samples preserve the output format but do not contribute to the box reward in GS-DPO.

4 Stage-III GS-DPO Details

4.1 Reward Components

The GS-DPO reward consists of five components: format, answer, box, sentence-embedding-based semantic rationale, and coupling reward. In the reported experiments, we use a fixed default weighting scheme for these reward components across all comparisons. The weights are kept unchanged for all ablations and test settings, and are not tuned separately for each variant.

Threshold setting. We use fixed thresholds for semantic-direction filtering and localization validity: $\tau_{\text{sem}} = 0.03$ and $\tau_{\text{box}} = 0.10$. The same thresholds are used for all models and ablations to avoid per-setting tuning.

4.2 Semantic Reference Banks

We use sentence embeddings from `BAAI/bge-small-en-v1.5` [3] as a lightweight semantic scorer. The generated rationale is formed by concatenating the generated `<evidence>` and `<logic>` fields. We compare its embedding with three semantic reference banks corresponding to normal-supporting, abnormal-supporting, and domain-oriented reasoning. The reference banks are fixed textual references used only for reward computation, and are not trainable memory modules.

Table S3: Example semantic references for sentence-embedding-based rationale scoring.

Type	Example reference sentence
Normal	The query image is normal and shows no visible localized defect.
Abnormal	The query image contains a visible localized defect.
Domain	The rationale describes object category, components, or industrial domain knowledge.

4.3 Box Matching and Normal-Case Rule

For abnormal samples, predicted boxes and ground-truth boxes are matched using IoU-based Hungarian matching. Unmatched extra boxes receive a count penalty. For normal or negative-region samples, the only valid location is an empty list. Any non-empty predicted box is treated as invalid localization.

Multi-box prediction. The `<location>` field may contain multiple boxes for multi-instance defects. During reward computation, predicted boxes are matched to ground-truth boxes with IoU-based Hungarian matching. To avoid reward hacking through excessive proposals, unmatched predicted boxes are penalized by the count penalty in R_{box}. In implementation, we cap the parsed box list to a small fixed maximum number of valid boxes and discard boxes with invalid coordinates or non-positive area. Unless otherwise specified, the same parsing and matching rule is used for all ablations.

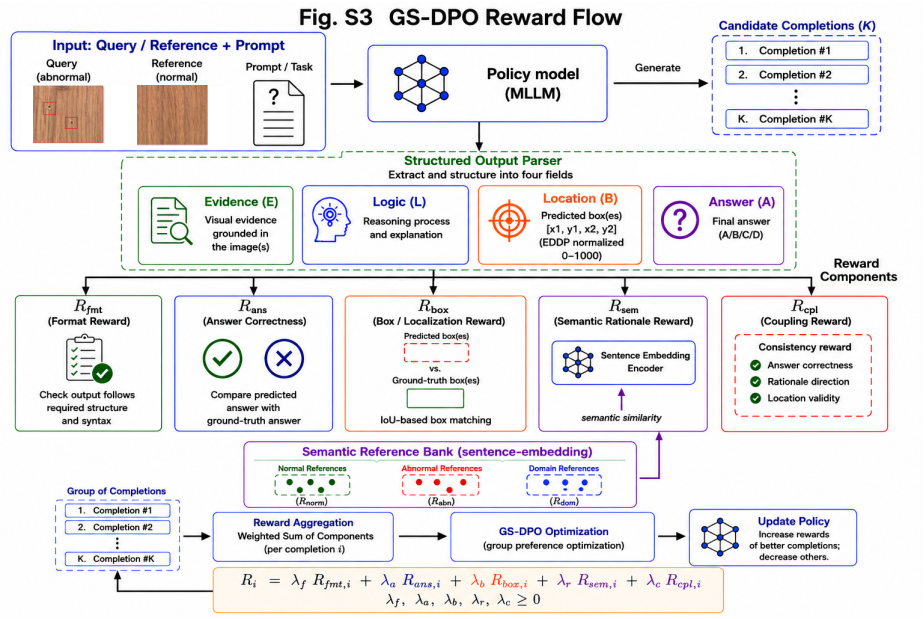

Fig. S3: GS-DPO reward computation. The reward is computed from format validity, answer correctness, box matching, sentence-embedding-based semantic similarity, and answer–location–rationale consistency.

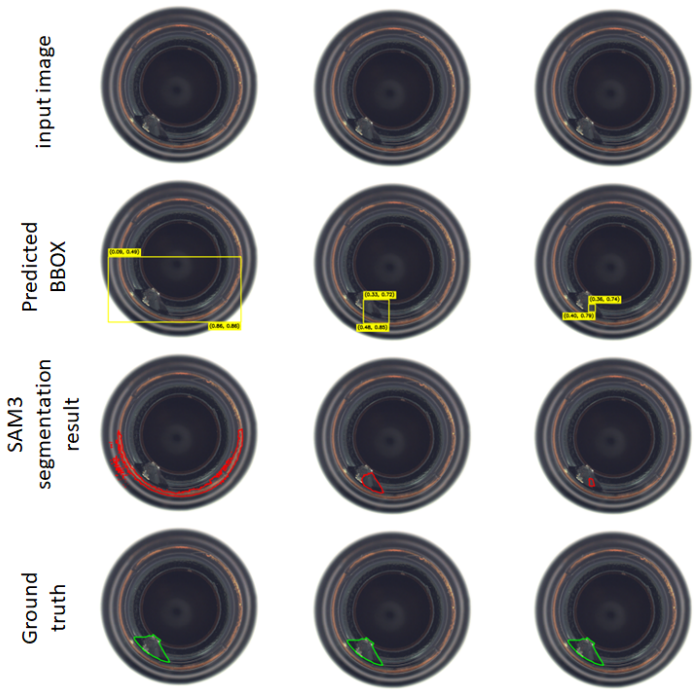

Fig. S4: Box-Guided Segmentation Rendering example. Predicted boxes from `<location>` are converted into segmentation prompts and rendered by an external segmentation backend. BGSR does not change the MLLM answer; it only renders masks from predicted boxes.

5 Stage-IV BGSR Details

BGSR is a visualization module. It parses the predicted `<location>` field. If the location is empty, no mask is generated. If a non-empty box is predicted, the normalized coordinates are converted back to image pixels and sent to an off-the-shelf segmentation backend.

$$x_1^{pix} = \frac{x_1}{1000}W, \quad x_2^{pix} = \frac{x_2}{1000}W, \quad y_1^{pix} = \frac{y_1}{1000}H, \quad y_2^{pix} = \frac{y_2}{1000}H. \tag{1}$$

6 Additional Qualitative Examples

We provide an additional MMAD qualitative example beyond the main paper. The example includes the query image, the normal reference, the generated EDDP trace, and the predicted location, illustrating how EGVLR grounds its answer in localized structural evidence.

Model	Diagnostic reasoning trace	Prediction
Qwen3-VL-Instruct (8B)	The response identifies the object as a cigarette pack, but attributes the anomaly to surface scratches and scuffs. It overfocuses on superficial texture differences around the printed region and misses the primary package-structure cue: layer lifting and partial separation near the package edge.	× Scratches
EGVLR (Ours) (8B)	`<think>` `<evidence>` Compared with the normal reference, branding and package-style differences alone are not sufficient evidence of anomaly. Localized inspection shows that the outer package layer is lifted and partially separated near the edge, forming a visible opening / peeling structure rather than a mere surface scuff. `</evidence>` `<logic>` The abnormal cue is material separation and opened package geometry. Therefore, the defect corresponds to peeling rather than scratches; surface marks may coexist but do not explain the primary structural change. `</logic>` `</think>` `<location>[[190,40,620,190]]</location>` `<answer>A</answer>`	✓ **Peeling**

Fig. S5: Qualitative comparison on a GoodsAD *cigarette_box* case [4]. The ground-truth defect type is peeling. Qwen3-VL-Instruct overfocuses on superficial scratches, whereas EGVLR follows the unified EDDP schema, focuses on damage-related structural evidence, localizes the damage-related region with an EDDP normalized 0–1000 box, and predicts the correct answer.

7 Limitations

EGVLR relies on box-level evidence during reasoning and uses an external segmentation backend only for mask rendering. The sentence-embedding-based semantic reward is a lightweight text-based signal over semantic reference banks and does not by itself verify whether a rationale is visually true. In addition, Stage-I visual pre-alignment uses synthetic localized anomalies, which may not cover all real defect morphologies, cluttered scenes, or extremely small defects. Our current evaluation focuses on MMAD-style industrial anomaly reasoning; broader cross-dataset evaluation and detector- or segmenter-assisted MLLM comparisons remain important future work. Future work will explore stronger multi-instance localization rewards, richer visual-semantic verification, and uncertainty-aware normal-case decisions.

8 Reproducibility Checklist

- All paired-image data use query-first and normal-reference-second order.
- All trainable stages use the EDDP schema.
- All boxes use EDDP normalized 0–1000 coordinates.
- Normal, negative-region, and non-spatial QA samples use `<location>[]</location>`.
- Stage-II samples without box supervision are marked as `require_location=false`.
- BGSR is used only for visualization and does not change answer accuracy.

References

1. Kang, H., Lee, W., Kim, J., Park, H.: JUDO: A juxtaposed domain-oriented multimodal reasoner for industrial anomaly QA. In: The Fourteenth International Conference on Learning Representations (ICLR) (2026)
2. Wang, C., Zhu, W., Gao, B.B., Gan, Z., Zhang, J., Gu, Z., Qian, S., Chen, M., Ma, L.: Real-IAD: A real-world multi-view dataset for benchmarking versatile industrial anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024)
3. Xiao, S., Liu, Z., Zhang, P., Muennighoff, N.: C-Pack: Packaged resources to advance general Chinese embedding. arXiv preprint arXiv:2309.07597 (2023)
4. Zhang, J., Ding, R., Ban, M., Dai, L.: PKU-GoodsAD: A supermarket goods dataset for unsupervised anomaly detection and segmentation. IEEE Robotics and Automation Letters **9**(3), 2008–2015 (2024)