

Peer Review

Review of: "Vision-Language In-Context Learning Driven Few-Shot Visual Inspection Model"

Evgenii Arsentev¹

1. Independent researcher

1. What the work claims

The authors want one visual inspection model that works on a product it has never seen, with no retraining and no threshold tuning per product. Their recipe has three parts. First, they fine-tune ViP-LLaVA (CLIP-ViT-L/14 encoder, LLaMA2 decoder, §3.2) on a small web-collected dataset of 941 images in 84 categories (§3.3), in a fixed VQA format: the answer is either "None" or the bounding-box coordinates of the defect. Second, at test time they give the model one example image with explanatory text through in-context learning. Third, they choose that example with a nearest-neighbour rule in ResNet50 feature space (Eq. 1), which they present as an improvement over the cosine-similarity RICES selector.

The headline result, stated in the abstract and in the contributions list, is MCC 0.804 and F1 0.950 on MVTec AD in a one-shot setting (Table 1, row "All category", column ICL (Ours)). On VisA the corresponding figures are F1 0.795 and MCC 0.479 (Table 2, row "All category", ICL (Ours)).

There is a lot here I value. The problem is a real one on factory floors, the output is a box a human can check rather than a bare score, and the authors chose MCC over F1 for a stated and correct reason (§4.1). They also report their failures openly: multiple defects and logical defects such as those in Cable, several objects per image in Candle, Capsules, and Macaroni2, poor localisation. Code is linked. That is more candour than many papers in this area show.

2. Whether the conclusion holds on the data

I start with what holds. Fine-tuning clearly does something useful: the vanilla model gives MCC 0.068 on MVTec AD (Table 1, row All category, Vanilla), and after fine-tuning without any example it reaches 0.519

(same row, w/o ICL). Appendix 7 explains why the vanilla row is mostly N/A (inconsistent answer format, everything called defective), and that explanation is plausible. The argument in §4.1 about F1 inflation is also well made. I checked the arithmetic: predicting all 1,258 positives and 467 negatives as defective gives $F1 = 2 \times 1258 / (2 \times 1258 + 467) = 2516 / 2983 = 0.8434$. The text reports 0.844; this should read 0.843 after rounding. A very small point, but worth fixing because the whole paragraph is about taking metrics seriously.

The main question I have concerns where the in-context example comes from. In Eq. 1, q is "the index of the test image" and i is "the index of the image except for q ". As I read it, the candidate pool is the rest of the test set of the same product. The example is described as a non-defective or defective image with explanatory text, which suggests its label is given to the model. If that is correct, then the nearest neighbour's ground-truth label is passed to the model for every query. MVTec AD and VisA contain many test images of the same defect type that look alike, so the neighbour's label may already predict the query's label well. The model could then score highly by copying the example's verdict, and the gain would come from the benchmark's structure rather than from inspection ability.

The appendix makes me take this seriously. With a randomly chosen example, the best all-category MCC is 0.475 (Table 3, row All category, column 1-neg), and a random defective example gives 0.279 (column 1-pos). Both are below the no-example result of 0.519 (Table 1, w/o ICL). Swapping random for nearest-neighbour selection lifts MCC to 0.804 (Table 1, ICL (Ours)). A jump of $0.804 - 0.475 = 0.329$ from the choice of example alone is large. It fits "better examples help", but it fits label leakage equally well, and the current experiments cannot tell the two apart. I would add three things: the MCC of a plain 1-NN classifier that just copies the neighbour's label (ResNet50 features, same pool); the share of queries whose selected example had the same label as the query; and a run where the pool is restricted to non-defective images, or to a held-out split that never overlaps the test images. If the VLM beats the 1-NN copy baseline clearly, the paper's claim becomes much stronger.

The claim that Eq. 1 beats RICES needs a narrower wording. On MVTec AD, it does: all-category MCC 0.665 for RICES versus 0.804 for Eq. 1 (Table 1). On VisA, it does not: 0.492 versus 0.479 (Table 2, All category, MCC), and RICES is higher on Capsules (0.513 vs 0.389), Fryum (0.741 vs 0.648), Macaroni1 (0.204 vs 0.190), PCB1 (0.792 vs 0.762), and PCB2 (0.493 vs 0.471). The authors acknowledge this in §4.2.2, but the abstract and conclusion do not. I also could not find which encoder RICES used here. If RICES ran on CLIP features and Eq. 1 on ResNet50, the comparison mixes encoder and distance. On L2-normalised features, cosine and Euclidean distance give the same ranking, so the argument in §3.4 about scale and dimension

only applies if features are not normalised. Running both rules on the same ResNet50 features would isolate the effect of the distance itself.

The statement that ICL improves VisA "as well" holds on the aggregate (MCC 0.429 to 0.479, Table 2, All category) but not per product. For Candle, w/o ICL gives MCC 0.539 and ICL (Ours) 0.253. For Macaroni1, it is 0.502 versus 0.190. On MVTec AD, RICES falls below the no-example model on Grid (0.476 vs 0.910) and Capsule (0.384 vs 0.426). These regressions deserve a sentence in the text.

There are four further points on evidence.

- No external baseline appears in any table. WinCLIP and AnomalyGPT are discussed in §2.1, and PatchCore in few-shot mode is standard. The authors argue that WinCLIP needs a threshold. That is fair, but reporting it with a threshold fixed on a separate split, alongside the paper's own numbers, would let readers place 0.804 on a scale.
- Each number comes from a single run. With 1,725 MVTec test images, the aggregate is fairly stable, but per-product MCC on a few dozen images can move a lot. Bootstrap intervals for the All category rows would help.
- Web-scraped training images may overlap the test sets. §4.1 says MVTec AD and VisA "were not used at all during training", but images from these benchmarks are widely re-hosted online. A perceptual-hash check of the 941 images against both test sets would settle this.
- "New products" is partly true. §4.2.1 says Bottle and Wood also appear in the training data, and Appendix 6 mentions carpet renamed to textiles. A split of Table 1 into seen and unseen categories would show how much of the result is true transfer.

Pixel-level AUROC of 0.730 is quoted in §4.2.1, but I could not find how an AUROC was computed from a single box output, nor a table with it. The authors are honest that localisation is weak. I would still report the method and a box-level metric such as IoU.

3. Reproducibility

The training setup is described better than usual: 300 epochs, ZeRO2 with XTuner, 8 NVIDIA 6000Ada-48GB GPUs, per-device batch 4 and global batch 32, AdamW at $1e-4$, warm-up ratio 0.03, cosine decay (§4.1). A GitHub link is given, and the fine-tuning prompt template is quoted in §3.3. I did not run the code; my comments cover only what the paper text states.

What I could not find:

- The size and checkpoint of ViP-LLaVA and LLaMA2 (7B or 13B, which release).
- Decoding settings: temperature, sampling or greedy, maximum tokens. How free text was parsed into a decision, and what happened with unparseable answers. The N/A cells in Tables 1 and 2 partly depend on this.
- The exact ICL prompt at test time, including the "explanatory texts that serve as inspection criteria" promised in the abstract. I did not find a single example of such text in the paper, and I could not tell whether it is written by hand for each product or generated.
- Which ResNet50 weights and layer were used in Eq. 1, whether features were pooled and normalised, and the encoder used for RICES.
- Seeds, and how many random draws stand behind each column of Table 3.
- The GPT-4 prompt for keyword generation, the annotation protocol for boxes (who annotated, how many annotators), and the per-category image counts. Appendix 6 says the images "will be" public, so the dataset is not yet available.
- Input resolution, and whether a validation split was used to choose 300 epochs on 941 images. That is a long schedule for a small set, and some measure of overfitting would help.

4. *What specifically to fix*

1. **Eq. 1 and §4.1.** State explicitly which images form the candidate pool and whether their labels are shown to the model. Then add the 1-NN label-copy baseline and the label-agreement rate described in my section 2. This is the most important addition.
2. **Eq. 1 notation.** The expression $\operatorname{argmin}(\min \|f(x_i) - f(x_q)\|_2)$ has a nested min that does nothing. Writing $i^* = \operatorname{argmin}_{i \neq q} \|f(x_i) - f(x_q)\|_2$ would be clearer.
3. **§4.1, F1 example.** Correct 0.844 to 0.843 (2516/2983).
4. **Abstract and §5.** Say that Eq. 1 improves on RICES on MVTec AD but not on VisA, citing Table 2, All category, MCC 0.492 vs 0.479.
5. **§3.4.** Rerun RICES on the same ResNet50 features, and say whether the features are normalised. Without this, the cosine-versus-Euclidean argument is not tested.
6. **Tables 1 and 2.** Add at least one external few-shot baseline (WinCLIP or PatchCore) with a threshold fixed on data outside the test set. A column of per-product test counts would also let readers see why some cells are N/A.

7. **Table 1, caption.** It says "Bold means the highest performance", but as rendered on Qeios I could not see which values are bold. Please mark best values in a way that survives the HTML conversion.
8. **§4.2.2.** Mention the per-product regressions with ICL: Candle (MCC 0.539 to 0.253) and Macaroni1 (0.502 to 0.190).
9. **§3.3.** The fine-tuning question asks for "the anomaly mode and the bounding box coordinate", while the answer is described as coordinates only. Please say whether the defect type is part of the target text.
10. **§4.2.1, pixel-level AUROC 0.730.** Explain how this number is computed from a box and put it in a table, ideally with IoU.
11. **Table 1 vs training set.** Split results into categories present and absent in the 84 training categories, and report a near-duplicate check against the test sets.
12. **Table 3.** Say how random examples were drawn and over how many repeats, and add an Eq. 1 column so the comparison with random is inside one table.
13. **§3 or appendix.** Show one complete ICL prompt with its explanatory text.

5. Recommendation

Major revisions. The idea is useful, and the paper is honest about what fails. Fine-tuning brings a clear and well-documented gain, and the code is available. My hesitation is narrow, but it touches the headline: if the in-context example is drawn from the labelled test set, part of the 0.804 may come from the neighbour's label. A 1-NN label-copy baseline and a clean candidate pool would answer this in a few runs. With that, the VisA wording, and the missing decoding and prompt details, I expect this to become a solid contribution.

Declarations

Potential competing interests: No potential competing interests to declare.