

Peer Review

Review of: "When Large Vision-Language Models Meet Person Re-Identification"

Evgenii Arsentev¹

1. Independent researcher

1. What the work claims

The paper asks a simple question that I have not seen answered cleanly before: can a generative vision-language model be put inside an ordinary person re-identification pipeline, where the output has to be a feature vector for cosine retrieval rather than a sentence? The proposed LVLM-ReID does this with Qwen2-VL-2B (§4.1.3). A pedestrian image is encoded into visual tokens, the frozen LLM receives the fixed instruction "Summarize the person image into one word, focusing on age, gender, clothing, and biometric features" (§3.2), and the representation of the single generated token, called <REID>, is concatenated with the visual tokens and passed through a 4-layer Transformer, the Semantic-Guided Interaction (SGI) module (§3.3). The refined <REID> token is trained with label-smoothed identity loss and triplet loss (§3.3, Eqs. 5–7, combined in Eq. 8 of §3.4) and is used as the retrieval descriptor at test time (§3.4). A learnable per-camera embedding is added either to the generated token or to the patch embeddings (§3.2, "Camera semantic supplementation").

The headline results are in Table 1: 82.8 mAP / 92.2 Rank-1 on DukeMTMC-reID, 89.2 / 95.6 on Market-1501, and 82.3 / 84.6 on CUHK03 (row LVLM-ReID). Ablations (Tables 2–5) are offered as evidence that the generated token, the bidirectional SGI design, the input-side camera embedding, and end-to-end gradient flow each contribute.

2. Whether the conclusion holds on the data

What I value most here is the framing. Instead of asking the LVLM to compare image pairs, which would be hopeless at gallery scale, the authors keep the standard extract-once, match-by-cosine paradigm and use the language model only to shape one token per image. That is a practical idea, and the method needs no extra text annotation. The ablation set is also well chosen: Table 2 contains a learnable-token control

("Ours w/o PSTG"), which is exactly the comparison a sceptical reader asks for first. The CUHK03 gain is the most convincing number in the paper: 82.3 mAP against 80.3 for CLIP-ReID (Table 1, CUHK03 columns), a difference of 2.0 points, with Rank-1 84.6 against 81.6.

Some sentences go further than the tables, though. In §4.2, the text says the method "achieves consistently better results across the three datasets." On Market-1501, that is not what Table 1 shows. LVLM-ReID has 89.2 mAP, while PFD and CLIP-ReID both report 89.6. Its Rank-1 of 95.6 is below SAN (96.1), RGA-SC (96.1), MGN (95.7), and Pyramid (95.7). The abstract's word "competitive" is the accurate one, and I would align §4.2 with it. On DukeMTMC-reID, the mAP margin over CLIP-ReID is $82.8 - 82.5 = 0.3$ points. With a single run and no reported variance, I cannot tell whether that is a real difference.

Table 2 needs a second look. The row "Ours w/o SGI" (79.0 / 90.0 / 87.3 / 94.5) is equal to or slightly below the Baseline row (79.0 / 90.2 / 87.3 / 94.7) on every metric. So the generated token by itself adds nothing over averaging the visual tokens. The text describes this variant as achieving "reasonably good performance, suggesting that our PSTG mechanism effectively captures essential pedestrian semantic information," which the numbers do not support. The honest reading, I think, is more interesting: the semantic token helps only when it can interact with the visual tokens. From the same table, SGI with a learnable token lifts DukeMTMC-reID mAP from 79.0 to 80.9 (+1.9), and swapping in the generated token lifts it to 82.8 (another +1.9). That decomposition is worth stating in the paper.

The bigger open question is what the generated token actually carries. In "Ours w/o PSTG" the LLM is, as I read §4.3, removed from the path entirely. The full model therefore differs from that control in two ways at once: it runs the visual tokens through roughly two billion frozen LLM parameters, and it uses an appearance-focused instruction. The gain could come from either. I could not find an experiment that separates them, for example, the same pipeline with an empty or unrelated instruction, or the LLM hidden state at the last visual token with no instruction at all. Section 3.2 says "Through empirical evaluation, we find that simple, clear instructions work effectively," but no instruction ablation is reported. Without it, the claim that the token encodes "age, gender, clothing" semantics rests on the attention maps in Fig. 3, which are suggestive but qualitative.

Table 5 has a result I could not interpret. "Stop Gradient" reaches 73.1 mAP on DukeMTMC-reID and 84.7 on Market-1501, well below the Baseline in Table 2 (79.0 and 87.3), which uses no LLM at all. If the SGI module still receives the visual tokens directly, the visual encoder should still get gradients through that path, and I would expect this variant to land somewhere near the baseline, not 5.9 points under it (79.0 -

73.1). A sentence explaining which gradients are blocked, and why the result falls below the baseline, would make this ablation much more persuasive.

Table 3 is also mixed rather than monotone. CSS-*reid* improves DukeMTMC-reID mAP over "w/o CSS" (82.3 vs 81.6) but lowers Market-1501 mAP (88.4 vs 89.1). The text's statement that the result "shows that camera semantics can improve the representation ability of the generated tokens" holds for one dataset only. Note too that the full model's advantage over "w/o CSS" on Market-1501 is 0.1 mAP (89.2 vs 89.1).

Finally, cost. The method runs a 2B-parameter LVLM forward pass on every query and gallery image. The Limitations paragraph acknowledges overhead in general terms, but Table 1 compares against CNN and ViT methods without parameter counts, FLOPs, or throughput. Readers will want to know what the 0.3 to 2.0 mAP gains cost.

3. Reproducibility

Several details are given clearly. The exact instruction text is printed in §3.2. Input size, patch size, and token count are stated in §4.1.3, and I checked the arithmetic: with 280×140 input and 14-pixel patches, $N = (280 \times 140) / 14^2 = 39,200 / 196 = 200$ patches, and the 2×2 merge gives $n = 200 / 4 = 50$, matching the stated 50 visual tokens. Batch composition (16 identities $\times$ 4 images), the Adam optimizer with weight decay 3×10^{-4} , a warmup from 3×10^{-5} to 3×10^{-4} over the first 10 epochs, schedule (60 epochs, decay 0.1 at epoch 30), triplet margin 0.3, and the loss weights 0.25 and 1 are also reported.

What is missing is the rest of a reproduction kit. I found no link to code or trained weights. The model is named only as "Qwen2-VL-2B"; I could not tell whether the base or Instruct checkpoint was used, or which release. There is no statement of random seeds, number of runs, GPU type, training time, or memory.

The generation step itself is underspecified. §3.2 says the LVLM "generate[s] a semantic token", and the camera paragraph refers to "the encoding of the <REID> token." Is this the final-layer hidden state at the next-token position, the embedding of the decoded word, or something else? If a word is actually decoded (greedy? sampled?), how does gradient flow back to the visual encoder, which Table 5 says it does? §3.4 also says the LLM parameters are frozen "though we retain its gradients"; one sentence on what that means in the implementation would remove the ambiguity.

4. What specifically to fix

1. §4.2, sentence "Our method achieves consistently better results across the three datasets": please revise. Table 1, Market-1501 columns, shows lower mAP than PFD and CLIP-ReID (89.2 vs 89.6) and lower Rank-1 than SAN, RGA-SC, MGN, and Pyramid.
2. Report mean and standard deviation over at least three seeds for the LVLM-ReID row of Table 1 and for the rows of Table 2. The DukeMTMC-reID mAP margin over CLIP-ReID is 0.3 points.
3. Add an instruction ablation to §4.3. Even three rows would do: the current instruction, an empty or neutral instruction, and an unrelated one (for example, asking about the background). This is the cleanest way to show that the appearance-focused prompt, and not the extra LLM depth, drives the gain from 80.9 to 82.8 mAP (Table 2, DukeMTMC-reID).
4. Rewrite the "Effectiveness of the SGI module" paragraph. Table 2 shows "Ours w/o SGI" at or below the Baseline on all four metrics, so it does not show that PSTG alone captures useful semantics.
5. Table 5: explain why "Stop Gradient" (73.1 mAP, DukeMTMC-reID) falls 5.9 points below the no-LLM Baseline of Table 2. Which parameters still train in that variant?
6. In §3.2, define precisely what vector the "encoding of the <REID> token" is (hidden state at which layer and position, or embedding of a decoded word), and how the token is decoded.
7. A cost table, please. Parameters, inference FLOPs, or images per second for LVLM-ReID next to TransReID and CLIP-ReID would let readers weigh the accuracy gains in Table 1.
8. Table 1 should mark which compared methods also use camera IDs at test time, since LVLM-ReID relies on the camera embedding (Table 3, the last row, CSS-x_p, where the embedding is added to the patch embeddings) and not all baselines do.
9. §4.1.1, Market-1501: the test set is given as 19,281 images. The commonly used split has 19,732 gallery images plus 3,368 queries; I would double-check this figure and state query and gallery counts separately for all three datasets.
10. Consider adding MSMT17, which is larger and harder than the three benchmarks used. A brief note on the status of DukeMTMC-reID, which its creators withdrew, would also be appropriate.
11. Small things: the duplicated citation "[12][12][14]" in §2.2 (presumably [12][13][14]); "w/o PSTG" in §4.4; and the claim in §5 that the experiments "certify" the integration, which is stronger than single-run results allow.

5. Recommendation

Minor revisions. The idea of using one instruction-guided generated token as the ReID descriptor is new, practical, and well motivated, and the CUHK03 result is a clear gain. The fixes are mostly about matching the text to Tables 1–3, adding seeds, one instruction ablation, and a cost table, and explaining the Stop Gradient result. None of them changes the core contribution. I would be glad to see the revised version.

Declarations

Potential competing interests: No potential competing interests to declare.