PHANERO: PARTIAL-OCCLUSION HIDDEN-STATE ALIGNMENT WITH NEURAL-DIMENSION EVIDENCE FOR RECOVERING OBJECTS

Anonymous authors
Paper under double-blind review

ABSTRACT

A system that already runs a vision-language model (VLM) for captioning or planning can ask it for a bounding box instead of serving a separate detector. Occlusion is what breaks that arrangement: boxes emitted as text become unreliable exactly when the object is partly hidden, and the standard repairs — prompting, supplied geometry, fine-tuning — all act before deployment. PHANERO asks what can be repaired during the decode itself. Occlusion displaces the decoder's final hidden state, and part of that displacement is linearly recoverable. From clean and occluded decodes of the *same* image we fit one map back to the clean state and apply it through a forward hook where coordinates become tokens, keeping the unrepaired boxes alongside. Nothing is trained and no gradient is taken; the map is fit once per model on COCO under synthetic occlusion, then used unchanged on KITTI and HOOT under natural occlusion, where it raises recall at IoU 0.5 (R_{50}) in all nine model×domain cells and outscores published pixel-, attention- and visual-token-space repairs in nine of eighteen paired tests, losing one. Keeping the unrepaired boxes means any second decode collects part of that gain, so we pre-registered the test that separates them: the same union built from a norm-matched random write and from a resampled decode. The learned correction clears the stronger null on all three models ($+0.016$ to $+0.079$; 6 of 9 cells, missing KITTI on two models and Qwen3-VL HOOT), on the weakest model partly as extra boxes. Rescaled to its edit size, maps without the pairing or the cross-dimensional terms fall short of it in 10 of 11 localization tests. What this buys is narrow and cheap. A stock open-vocabulary detector with a tuned threshold still beats every cell we repair; the claim is only that a VLM already in service can localize better under occlusion for one extra decode — $1.05\times$ a plain query ($1.30\times$ by total time) with both decodes batched.

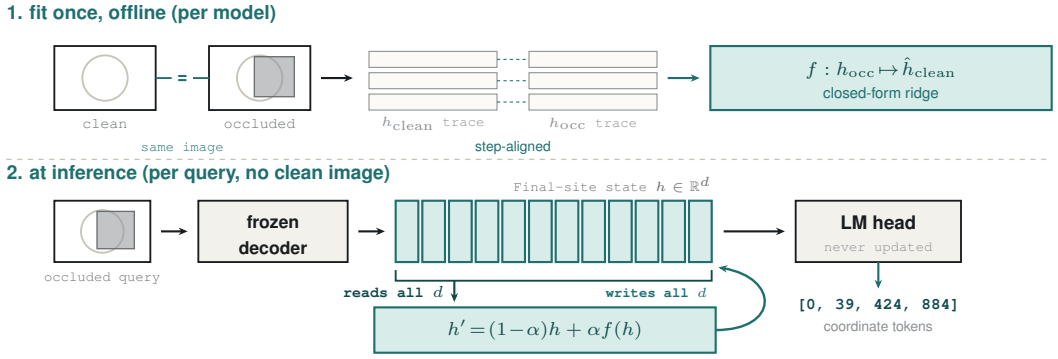

Figure 1: **PHANERO reconstructs a frozen decoder's clean final hidden state from its occluded one.** *Offline*, clean and synthetically occluded versions of the *same* image are decoded and their Final-site traces paired step-by-step; a closed-form ridge fit gives f. Section 4.4 shows the repair needs both this pairing and the full cross-dimensional map: a full-rank map fit on the unpaired populations, or a paired per-dimension one, recovers only a small fraction of the effect. *At inference* no clean image is available: f reads the entire d-dimensional state and its output replaces the whole state (shaded); restricting the write to a subset is an ablation. The LM head is never updated.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

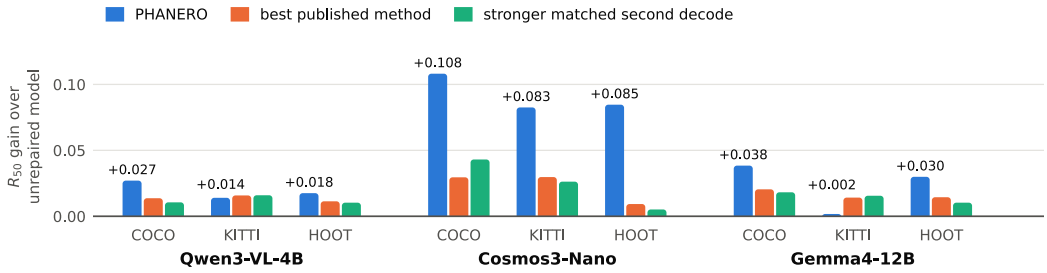

Figure 2: **At a glance.** Gain in recall at IoU 0.5 over each unrepaired model, on the full evaluation sets (751/800/969 questions). PHANERO (blue) against the best of three published methods in each cell (orange) and the stronger of two second decodes that learned nothing (green). It has the largest gain in seven of nine cells; on KITTI with Qwen3-VL and Gemma4 it does not exceed them. Exact values: tables 1, 6 and 30.

1 Introduction

A robot that already runs a vision-language model (VLM) to describe what it sees can also ask that model where things are: *give the box for the cup*. For any system that carries a VLM for captioning, question answering or planning, this is the cheapest detector there is — nothing extra to train per category, serve, or keep in sync. It fails where it matters most. When the object is partly hidden, the coordinates the model writes out go wrong: the strongest VLMs trail humans by 30–40 points on occlusion questions (Wasi et al., 2026), and their errors grow with the number of occluded objects rather than with scene complexity (Pothiraj et al., 2025).

The usual remedies either act before deployment or need information a deployed system does not have. More reasoning in language does not reliably help and can cost over 20 points (Wasi et al., 2026; Vo et al., 2026); explicit spatial input helps but is built from annotations, maps or LiDAR (Vo et al., 2026; Pothiraj et al., 2025; Yang & Di, 2024); inpainting the hidden part gives inconsistent gains (Pothiraj et al., 2025); and fine-tuning changes the model every other capability in the system depends on.

PHANERO repairs the model's internal state instead, while it writes the box. Show a VLM the same image twice, once clean and once with part of the object covered, and its final hidden state — the vector each coordinate digit is read from — shifts. Part of that shift is predictable. We record it on about three thousand COCO queries, fit one linear map that undoes it, and at test time apply the map at every decoding step through a standard forward hook (fig. 1). Nothing is trained by gradient descent, no weight changes, and no clean image, box or region is needed at deployment. The unrepaired answer's boxes are kept alongside the repaired ones, so the repair adds candidates rather than overwriting an answer that was already right.

What it delivers (fig. 2). On three current VLMs (Qwen3-VL-4B, Cosmos3-Nano, Gemma-4-12B) and three datasets — COCO, where the map is fit, and KITTI driving and HOOT tracking footage, where it is used unchanged under natural occlusion:

- recall at IoU 0.5 rises in **all nine** model×domain cells, by up to +0.108 (Cosmos3-Nano on COCO, 0.533 → 0.641);
- against three published methods — attention read-out (Kang et al., 2025), amodal completion (Ao et al., 2025) and per-query visual-token optimization (Wu et al., 2024a) — it wins **17 of 27** paired comparisons and loses one; against the two stronger ones alone, 9 wins, 8 ties and 1 loss;
- it costs one extra decode, and because both decodes share one prefill they run as a single batch: 1.05× a plain query on Qwen3-VL as a per-query median, 1.30× by total time, with accuracy unchanged, where the closest competitor cannot batch and costs 2.2× by either measure.

Why these numbers hold up. Results like these invite three objections, and we tested each with the analysis fixed before the runs. *Any second answer would help* — keeping the baseline's boxes means a second decode of any kind can raise recall, so we compare against the same union built from a random edit of equal size and from simply resampling the model; the learned repair beats the stronger of the two on all three models, in 6 of the 9 cells (section 4.3). *It was tuned on the test set* — it writes every dimension, so there is no size to search, and a search on held-out validation data would have done worse out of domain (section C.1). *A simpler steering method would match it* — four cheaper corrections fit on the same data, including the standard ones from the steering literature

and each rescaled to PHANERO's edit size, fall short of it in 10 of 11 localization comparisons (section 4.4). Table 7 collects these and five further checks in one place.

Contributions.

- A deployment-time repair for VLM localization under occlusion that trains nothing, touches no weight, and adds one precomputed matrix per model (26–67 MB), fit in seconds from cached activations.
- Evidence that it transfers: fit on synthetic occlusion in COCO, it helps under natural occlusion in driving and tracking footage, and outscores published pixel-, attention- and token-space methods.
- A test that any method keeping its baseline's answer should pass — the matched second-decode null — pre-registered, with code, fitted maps, per-query outputs and the scripts behind every table released anonymously at `https://anonymous.4open.science/r/phanero-4D6E`.

2 Related Work

Diagnosing the failure. Recent benchmarks establish that occlusion-induced mislocalization is real and severe, attributing it to depth ordering, crowding and the construction of a coherent scene (Wasi et al., 2026; Vo et al., 2026; Pothiraj et al., 2025). They diagnose; we take their diagnosis as the starting point.

Fixes outside the model. Prompting frozen VLM detectors improves *what* is recognized, not whether a partly hidden object can still be placed (Madan et al., 2024). Amodal completion reconstructs the hidden appearance in pixels (Ao et al., 2025), and generic inpainting helps some models and hurts others (Pothiraj et al., 2025). Oracle coordinates and annotation-derived maps help but are, by their authors' own account, diagnostic priors built from ground truth (Pothiraj et al., 2025; Vo et al., 2026). Closest in spirit, FeatComp++ completes occluded pedestrian features inside a specialist detector (Zhang et al., 2024); we repair the native state through which a general VLM writes its own coordinates.

Fixes inside the model. Frozen VLMs keep usable spatial structure: a few attention heads suffice for grounding (Kang et al., 2025; Lin et al., 2024), and heads, subspaces or activations can be edited at inference to steer behavior (Yin et al., 2024; Wu et al., 2024b; Rodriguez et al., 2025; Wang et al., 2025; Sivakumar et al., 2025; Feng et al., 2026). These read localization out or steer semantics; none restores a displaced spatial state. ControlMLLM optimizes the visual tokens of each query afresh (Wu et al., 2024a), which needs a region to steer toward and a backward pass per query. Table 8 places each method by the internal object it acts on.

What is new here is not the unit of intervention — editing individual dimensions has precedent (Sivakumar et al., 2025; Feng et al., 2026) — but what is computed and from what evidence: a reconstruction of the clean final state from the occluded state of the *same* image, written back where that state becomes coordinates. And it writes every dimension, where the steering literature finds that editing fewer works better (Feng et al., 2026); for this task we find the opposite (section 3.3).

3 Method

PHANERO has three steps. The first two happen once per model, offline; the third is a hook attached for the length of a query.

3.1 Record and Fit, Once per Model

For each training query we decode the image twice: as it is, and with a flat patch covering 40% of the target object's box at a random position, the rest of the object left visible. At every decoding step we record the model's *final hidden state* — the vector at the output of the last normalization layer, which the output layer turns into the next token — and pair the t-th step of the occluded decode with the t-th step of the clean one. A ridge regression then fits one linear map from occluded states to clean ones. It is a closed-form least-squares solve that takes seconds on the cached vectors, involves no gradient through the VLM, and needs only the box annotations the training set already has.

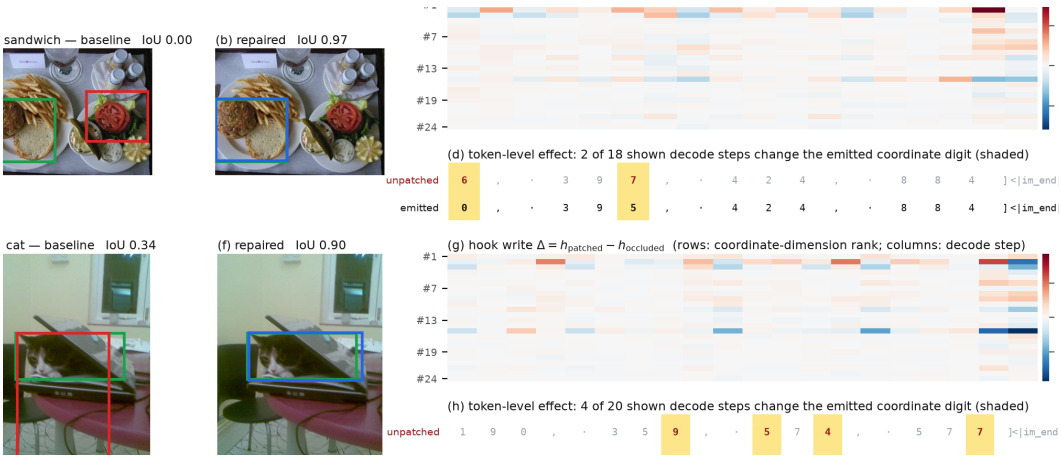

Figure 3: The correction traced end to end on two Qwen3-VL queries (COCO, frozen weights, no LoRA, $k{=}d$), showing its two characteristic failure modes. **Top, *wrong object*:** the baseline (red) locks onto the tomato rather than the 40%-occluded sandwich (green), IoU 0.00 (a); repaired (blue) it moves onto the sandwich, IoU 0.97 (b). **Bottom, *wrong extent*:** the baseline over-extends the cat's box through the table (IoU 0.34) (e); the repair tightens it to the visible animal (IoU 0.89) (f). In both, the hook's write at the Final site (c, g) is small and spread across dimensions, not a single-dimension edit, while its effect in token space is sparse (d, h): 3 of 18 and 4 of 20 decode steps change the emitted digit (shaded). Relocating or resizing the box is the arithmetic consequence of a handful of token substitutions — the level at which the repair must act for the unchanged LM head to read it out.

The map is *full*: every output dimension depends on every input dimension. That matters, because occlusion does not simply dim each dimension; it moves evidence between them, and section 4.4 shows that maps unable to capture this fall short of it on every model, even rescaled to its edit size.

3.2 Apply, per Query

At test time the model receives only the occluded image and the question — no clean image, no box, no region. A forward hook on the final normalization layer replaces the state h_t at every generated token with

$$h'_t \;=\; (1-\alpha)\,h_t \;+\; \alpha\,(h_t W + b), \qquad \alpha = 1, \tag{1}$$

where W ($d{\times}d$) and b are the fitted map; a partial blend $\alpha{<}1$ is an ablation (section B.2). The image and prompt are processed exactly as in an unrepaired query, and when the query ends the hook is removed: no weight is ever written, and the checkpoint is byte-identical before and after.

Figure 3 shows what this does on two real queries. The write is small and spread across many dimensions, yet only three or four emitted digits change — enough to move a box onto the right object, or to pull an over-extended one back to the visible animal.

3.3 Which Dimensions: All of Them

Steering methods usually edit a few dimensions or heads, and the recent finding is that editing fewer works better (Feng et al., 2026). We expected the same and found the opposite. Writing only the top-ranked 2–20% of dimensions recovers at most two fifths of the gain on any model, and the gain peaks at 80–100% (sections B.3 and B.4). Writing all of them also removes the one setting that would otherwise need tuning.

3.4 Keeping the Original Answer

What PHANERO returns is the *repair*: the patched decode's boxes, plus every box from the unrepaired decode that overlaps none of them at IoU $\geq$0.5. A correct original box is never lost. The price is that a second decode of *any* kind can add recall this way, which is why section 4.3 compares against second decodes that learned nothing. Triggering the repair only when the state looks anomalous did not beat applying it to every query (section D.11).

Table 1: Published comparators against each model's frozen base weights on the full held-out COCO set (n=751). Cells are the gain over each model's own Base under R_{50} (subscript = baseline failures recovered / n_{fail}) and average precision over IoU 0.50–0.95 with a size-penalizing proxy. LH = Localization Heads, Amod. = Amodal, CMLLM = ControlMLLM; bold = best per row. PHANERO writes every dimension (k=d, α=1), so nothing about it was searched here. Format-drift correction, Amodal's 570 real composites, cross-domain results and significance: sections C.7 and E.1

	ΔR_{50}					ΔAP				
Model	Base	LH	Amod.	PHANERO	CMLLM	Base	LH	Amod.	PHANERO	CMLLM
Qwen3-VL	0.572	$-0.002_{1/249}$	$+0.014_{23/249}$	$\mathbf{+0.027}_{27/249}$	$+0.005_{6/249}$	0.310	−0.0067	−0.0061	**+0.0131**	+0.0069
Cosmos3-Nano	0.533	$+0.000_{1/280}$	$+0.023_{37/280}$	$\mathbf{+0.108}_{89/280}$	$+0.030_{26/280}$	0.266	−0.0045	+0.0003	**+0.0388**	+0.0290
Gemma4	0.587	$-0.002_{0/242}$	$+0.009_{14/242}$	$\mathbf{+0.038}_{35/242}$	$+0.020_{22/242}$	0.336	−0.0026	−0.0107	+0.0129	**+0.0140**

Table 2: **Every cell, every method.** Gain in R_{50} over each model's Base (given), full evaluation sets; *: $p < 0.05$ against Base (paired bootstrap); bold: largest gain in the row. The last column is PHANERO's head-to-head record against the three published methods in that row (wins–ties–losses, paired, on the queries each can act on). Average precision and every test: table 30.

Model	Dom.	Base	LH	Amod.	CMLLM	PHANERO	W–T–L
Qwen3-VL	COCO	0.572	−0.002	+0.014*	+0.005	**+0.027***	2–1–0
	KITTI	0.290	+0.000	+0.010*	**+0.016***	+0.014*	1–2–0
	HOOT	0.328	+0.002	+0.011*	+0.002	**+0.018***	2–1–0
Cosmos3-Nano	COCO	0.533	+0.000	+0.023*	+0.030*	**+0.108***	3–0–0
	KITTI	0.402	+0.001	+0.012*	+0.030*	**+0.083***	3–0–0
	HOOT	0.358	+0.000	+0.009	−0.003	**+0.085***	3–0–0
Gemma4	COCO	0.587	−0.002*	+0.009	+0.020*	**+0.038***	2–1–0
	KITTI	0.279	−0.002	+0.001	**+0.014**	+0.002	0–2–1
	HOOT	0.269	+0.000	+0.014*	+0.012	**+0.030***	1–2–0

4 EXPERIMENTS

Setup. Three VLMs — Qwen3-VL-4B-Instruct (Qwen Team, 2025), Cosmos3-Nano (NVIDIA, 2026) and Gemma-4-12B-it (Gemma Team, 2026) — on three datasets: a held-out COCO split (Lin et al., 2014) (751 queries), KITTI driving scenes (Geiger et al., 2012) (800) and HOOT tracking footage (Sahin & Itti, 2023) (969), the last two never seen when the map is fit. The metric is recall at IoU 0.5 (R_{50}): the share of annotated objects matched by a predicted box that overlaps it by at least half, one box per object (Juanico et al., 2025). Every comparison is paired — same questions, same baseline boxes — and tested with a paired bootstrap. Query construction is in section A.1.

4.1 AGAINST PUBLISHED METHODS

PHANERO gives the largest recall gain on every model (table 1), against three published techniques ported to the same frozen weights: Localization Heads (Kang et al., 2025), which reads a box out of attention; Amodal Appearance Completion (Ao et al., 2025), which paints the hidden part back in; and ControlMLLM (Wu et al., 2024a), which optimizes the visual tokens of each query.

The lead is largest where the baseline is weakest: on Cosmos3-Nano recall reaches 0.641 against ControlMLLM's 0.563, and the lead holds on KITTI and HOOT (+0.054 and +0.088 head to head, section E.1). Under average precision, which also rewards tight boxes, PHANERO leads on Qwen3-VL and Cosmos3-Nano; on Gemma4 ControlMLLM keeps a narrow edge (+0.0140 against +0.0129) while PHANERO leads on recall (+0.038 against +0.020). Table 2 extends the comparison to all nine cells: PHANERO has the largest gain in seven, and counting only the queries each competitor can act on it wins 17 of the 27 head-to-head tests and loses one, to ControlMLLM on Gemma4 KITTI.

It rescues complete failures, most clearly on the weakest model. A question the unrepaired model gets completely wrong is rescued if a method's output finds at least one of its objects. Compared on the failures each method can act on (table 3), PHANERO rescues 2.4× as many as ControlMLLM, 2.0× as many as Amodal, and twice as many as resampling. Most of that margin is Cosmos3-Nano's, whose patched decode also emits extra boxes: without it, PHANERO is level with ControlMLLM (5.0% against 4.7%) and Amodal (6.5% against 5.8%), and still rescues 1.6× what resampling does. Figures 4 and 5 show rescues on real tracking and driving footage.

Table 3: **Complete failures rescued**, on comparable rows. A failure is a question on which the unrepaired model finds none of the annotated objects; a method rescues it if its output finds at least one. Each pair is PHANERO / the method, in %, on the failures that method can act on (ControlMLLM needs a baseline box to steer toward, Amodal a composite; the matched second decodes act everywhere). Cosmos3-Nano's patched decode emits extra boxes, and a rescue needs only one hit, so it is also shown without it. Per cell: table 21

	vs CMLLM	vs Amodal	vs Null-S	vs Null-R
All three models	11.5 / 4.9	13.3 / 6.5	10.8 / 5.4	10.8 / 3.2
failures compared	3,367	2,204	4,201	4,201
Without Cosmos3-Nano	5.0 / 4.7	6.5 / 5.8	5.6 / 3.5	5.6 / 1.5
failures compared	2,079	1,489	2,876	2,876



Figure 4: Recovery on real HOOT frames, frozen weights, $k{=}d$. Within each pair, left is the baseline (red) and right is patched (blue); green is ground truth. Queries where the baseline misses a lone target (IoU <0.3) and the patch hits it (≥ 0.75): 2, 23 and 6 of 969, so these illustrate the mechanism, not the typical query (section C.5).

4.1.1 What It Takes to Deploy

Table 4 summarizes what each method needs. **It acts on every query.** ControlMLLM needs a region to steer toward and Amodal a completed image; without one they return the baseline — on 12% and 41% of queries, mostly where the baseline already fails (section E.1). **It fits a standard serving stack**: a forward hook and one matrix, where ControlMLLM needs eager attention and a backward pass per query. **It is cheaper once batched.** Run one after the other, PHANERO's two decodes cost $2.0\times$ a plain query and ControlMLLM's $2.2\times$ (per-query medians; within 4% in absolute seconds). But only PHANERO's two decodes share a prefill — its write lands after the last layer — so only they can run as one batch: $1.05\times$ a plain query as a per-query median and $1.30\times$ by total time, against ControlMLLM's $2.2\times$ by either, with recall unchanged over the 751 COCO questions ($+0.003$, $p{=}0.34$). That saving assumes decoding is memory-bound, as it is for a single request or a lightly loaded server; a server already batching many requests pays for the second row in throughput. ControlMLLM's steered decode cannot start until its first decode has finished, and as images get larger its three prefills and backward pass widen the gap: projected to an edge accelerator, it rises to $3\times$ a plain decode at $\sim$2,300 visual tokens while batched PHANERO stays within 2% of a plain decode as a per-query median, and 6–14% above it by total time (fig. 6 and table 27) — optimistic figures, since on the A100 the model projects $1.12\times$ by total time where we measured $1.30\times$.

4.2 On Top of Fine-Tuning

PHANERO adds recall on top of LoRA fine-tuning in every cell (table 5), whether the adapter helped, hurt or did nothing. LoRA (Hu et al., 2022) is the weight-space alternative: a rank-16 adapter trained on COCO, at a budget chosen on validation data. Its effect is model-specific — significant only out of domain for Qwen3-VL, only in domain for Cosmos3-Nano, and harmful on Gemma4 COCO — while PHANERO alone is the larger single effect in seven of nine cells. Stacked, PHANERO adds a significant gain over LoRA in eight of nine. Part of that is what a second decode of the adapted model collects on its own: against that null the stacked gain holds in two cells and reverses in one (section D.2).



Figure 5: Recovery on real KITTI driving frames, frozen weights, $k{=}d$. Within each pair, left is the baseline (red) and right is the patched decode (blue); green is ground truth, which in this benchmark marks only the occluded instances asked about. Same rule as fig. 4 — a lone target the baseline misses (IoU <0.3) and the patch finds (≥ 0.75), the highest-IoU such query per model. It holds for 2 of 800 KITTI questions on Qwen3-VL, 16 on Cosmos3-Nano and none on Gemma4, which is therefore absent. Cosmos3-Nano's patched decode also boxes two vans the benchmark does not score: the unoccluded one in the foreground, and a distant one inside a region KITTI labels `DontCare`.

Table 4: What each method needs at inference, Qwen3-VL on COCO, 30 queries. Cost is against a plain decode, as the median of per-query ratios and as total time over the queries; "acts" is the share of queries a method changes at all. Only PHANERO's two decodes can share one batch.

	cost				
	median	total	grad.	eager	acts
Plain decode	1.00×	1.00×	—	—	—
PHANERO, batched	**1.05×**	**1.30×**	no	no	100%
PHANERO, sequential	2.00×	2.23×	no	no	100%
ControlMLLM	2.17×	2.22×	yes	yes	88%
Localization Heads	0.20×	0.15×	no	yes	100%
Amodal Completion	offline		—	—	59%

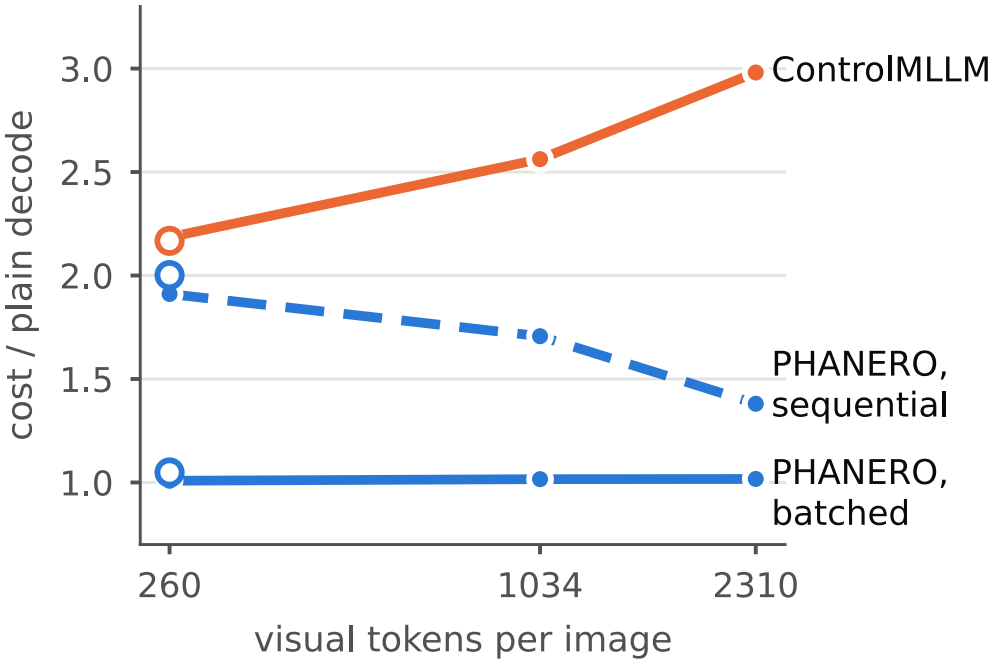

Figure 6: **Cost as images get larger**, relative to a plain decode, projected to a Jetson AGX Orin with a roofline model (section D.10); hollow markers are the values measured on an A100 at native resolution. Per-query medians; by total time the batched pair sits 6–14% above a plain decode, and every value is tabulated in table 27. Batched, PHANERO stays near the cost of a plain decode; run in sequence, its overhead shrinks as prefill comes to dominate; ControlMLLM, which prefills three times and runs a backward pass, grows to 3×.

Table 5: Cross-model, cross-domain effect of each condition. Subscript = the LoRA budget N (k = thousand): one per model, chosen on the image-disjoint COCO validation split and applied unchanged to all three domains. Bold and *: $p < 0.05$; $^{**}p < 0.01$, $^{***}p < 0.001$ (paired bootstrap + Wilcoxon). Both PHANERO columns report *repair*, the default. Populations, budgets, additivity and multiplicity: section E.2. The LoRA+PHANERO column clears a sampled decode of the same adapted model in two of nine cells and loses one (section D.2).

	LoRA-only			PHANERO-only			LoRA+PHANERO		
Model	COCO	KITTI	HOOT	COCO	KITTI	HOOT	COCO	KITTI	HOOT
Qwen3-VL	-0.005_{2k}	$\mathbf{+0.033}^{*}_{2k}$	$\mathbf{+0.044}^{***}_{2k}$	$\mathbf{+0.027}^{***}$	$\mathbf{+0.014}^{*}$	$\mathbf{+0.018}^{**}$	$\mathbf{+0.041}^{***}_{2k}$	$\mathbf{+0.036}^{***}_{2k}$	$\mathbf{+0.034}^{***}_{2k}$
Cosmos3-Nano	$\mathbf{+0.074}^{***}_{1k}$	-0.006_{1k}	$+0.015_{1k}$	$\mathbf{+0.108}^{***}$	$\mathbf{+0.083}^{***}$	$\mathbf{+0.085}^{***}$	$+0.020_{1k}$	$\mathbf{+0.050}^{**}_{1k}$	$\mathbf{+0.050}^{**}_{1k}$
Gemma4	$\mathbf{-0.031}^{**}_{250}$	-0.013_{250}	$+0.005_{250}$	$\mathbf{+0.038}^{***}$	$+0.002$	$\mathbf{+0.030}^{***}$	$\mathbf{+0.070}^{***}_{250}$	$\mathbf{+0.042}^{***}_{250}$	$\mathbf{+0.050}^{**}_{250}$

Table 6: PHANERO against matched second decodes, frozen weights, full n (751/800/969). The first four columns are gains in R_{50} over Base: *repair*, the patched decode alone, and the same union built from Null-R or Null-S. "−null" is repair minus the stronger null's union (subscript R/S); FP-AP is that difference under an AP counting every surplus box as a false positive. Bold and *: $p < 0.05$; $^{**}p < 0.01$, $^{***}p < 0.001$.

Model	Dom.	repair	patched	Null-R	Null-S	−null	FP-AP
Qwen3-VL	COCO	$\mathbf{+0.027}^{***}$	$+0.004$	$+0.007$	$\mathbf{+0.011}^{*}$	$\mathbf{+0.016}^{*}_{S}$	-0.0012
	KITTI	$\mathbf{+0.014}^{*}$	-0.001	$\mathbf{+0.009}^{***}$	$\mathbf{+0.016}^{**}$	-0.002_{S}	$+0.0038$
	HOOT	$\mathbf{+0.018}^{**}$	$\mathbf{+0.014}^{*}$	$+0.003$	$\mathbf{+0.010}^{**}$	$+0.007_{S}$	-0.0015
Cosmos3-Nano	COCO	$\mathbf{+0.108}^{***}$	$\mathbf{+0.031}^{*}$	$\mathbf{+0.028}^{***}$	$\mathbf{+0.043}^{***}$	$\mathbf{+0.065}^{***}_{S}$	-0.0351
	KITTI	$\mathbf{+0.083}^{***}$	-0.008	$\mathbf{+0.026}^{***}$	$\mathbf{+0.021}^{*}$	$\mathbf{+0.056}^{***}_{R}$	-0.0205
	HOOT	$\mathbf{+0.085}^{***}$	$\mathbf{+0.039}^{**}$	$+0.005$	-0.004	$\mathbf{+0.079}^{***}_{R}$	-0.0120
Gemma4	COCO	$\mathbf{+0.038}^{***}$	$+0.004$	$+0.005$	$\mathbf{+0.018}^{***}$	$\mathbf{+0.020}^{*}_{S}$	-0.0022
	KITTI	$+0.002$	$\mathbf{-0.014}^{*}$	$+0.006$	$\mathbf{+0.016}^{*}$	-0.014_{S}	-0.0060
	HOOT	$\mathbf{+0.030}^{***}$	$+0.007$	$+0.003$	$\mathbf{+0.010}^{*}$	$\mathbf{+0.020}^{**}_{S}$	-0.0030

Table 7: Objections a reader may raise, what we did about each, and where the evidence is. Every test was run on the full evaluation sets; the first was fixed in writing before any of its runs.

Objection	What we did	Outcome	Where
"Any second answer would do this."	Same union built from a random edit of equal size, and from resampling the model	Beats the stronger null on all three models, $+0.016$ to $+0.079$, in 6 of 9 cells: not on KITTI for Qwen3-VL and Gemma4, nor on Qwen3-VL HOOT	section 4.3
"It was tuned on the test set."	Writes every dimension, so nothing is searched; a validation search run as a check	Where that search picks differently (Cosmos3-Nano), it does worse out of domain: -0.031, -0.073	section C.1
"Simpler steering would match it."	Mean shift, Linear AcT, a paired per-dimension map, an unpaired full optimal-transport map; each also at PHANERO's edit size	At their own size none gains more than resampling; PHANERO clears it by $+0.016$ to $+0.065$. Size-matched, the OT map clears it on Cosmos3-Nano; PHANERO still wins 10 of 11	section 4.4
"It just adds boxes."	Box counts; an AP charging every extra box; an AP demoting large boxes	Level on Qwen3-VL and Gemma4; Cosmos3-Nano emits 1.6–1.8× the nulls' boxes	section 4.3
"It breaks unoccluded queries."	The same repair on unoccluded control sets	Patched decode changes no cell significantly; the repair lowers none and raises five	section C.6
"Test images leaked into the fit."	Rescored without the 176 COCO test questions whose images the fit saw	Gains unchanged on the other 575	section A.1
"It holds for one model only."	Three models, then the whole recipe repeated on a fourth, smaller one	Holds on all three; partial on the 2B model	section D.3
"It is too slow to serve."	End-to-end timing; the two decodes batched and rescored on 751 questions	1.05× a plain query (1.30× by total time), recall unchanged; ControlM-LLM 2.2×	section 4.1.1

4.3 Is It Just a Second Guess?

Keeping the original boxes means any second answer can add recall. To measure what the learned correction adds beyond that, we built two second decodes that learned nothing, fixed the analysis before running them (section D.1), and scored them with the same union. *Null-R* applies, at every step, a random edit of exactly the size PHANERO's would be; *Null-S* writes nothing and resamples the model at its shipped temperature.

Both nulls collect something — resampling alone adds $+0.010$ to $+0.018$ on Qwen3-VL and Gemma4 — **and the learned correction beats the stronger of them on all three models**: by $+0.065$, $+0.056$ and $+0.079$ on Cosmos3-Nano, $+0.020$ on Gemma4 COCO and HOOT, and $+0.016$ on Qwen3-VL COCO. On KITTI, Qwen3-VL and Gemma4 gain nothing beyond resampling, and Qwen3-VL HOOT's $+0.007$ is not significant. Against the random edit, the test fixed in advance, PHANERO is significant in 7 of 9 cells after correction for multiple comparisons, against a threshold of five set beforehand. On Qwen3-VL and Gemma4 the repaired decode emits about as

many boxes as the nulls, so the gain is not bought with boxes; on Cosmos3-Nano it emits 1.6–1.8× as many, and part of its margin comes with them.

4.4 What Makes It Work

Both of PHANERO's ingredients matter: the map is fit on *paired* states of the same image, and it is *full*, letting each dimension draw on all the others. We tested four cheaper corrections on the same data and hook (section C.2): a constant shift toward the clean average; Linear AcT (Rodriguez et al., 2025), the standard transport-based steering method, which matches the two populations dimension by dimension without pairing; the same per-dimension map fit on the pairs; and a full optimal-transport map between the unpaired populations, as expressive as PHANERO's but blind to which clean state belongs to which occluded one. At their own edit size **none gains more than resampling alone**, on any model, while PHANERO clears that floor by $+0.016$ to $+0.065$. But they also edit less, so we reran each at PHANERO's per-step edit size, the way Null-R is built (section C.3). That closes much of the gap: the unpaired full-rank map then clears the floor on Cosmos3-Nano ($+0.035$), so part of PHANERO's margin there is edit size, not direction. PHANERO still beats the size-matched controls in 10 of the 11 comparisons that are localization results (the twelfth, a paired diagonal on Cosmos3-Nano, knocks the model off its output format), that map on Cosmos3-Nano included ($+0.030$, p=0.004); the exception is the same map on Qwen3-VL, which comes within $+0.014$ (p=0.056), and on Qwen3-VL every margin is small ($+0.014$ to $+0.020$).

The gain survives stricter metrics, up to a point. It holds in 8 of 9 cells under average recall and corrected average precision, both of which reward tighter boxes (table 25 and section D.7), and under a variant of AP that demotes large boxes it is *larger* in 8 of 9 cells, the opposite of what box inflation would produce. It fades as the overlap criterion tightens and is gone by IoU 0.90 (section D.6): the repair puts the box on the right object; it does not sharpen its edges.

4.5 Limits

Scope. Decoders that write coordinates as text, from 2 to 12 billion parameters; other coordinate formats are untested. **Where the gain is.** Largest on the weakest baseline, Cosmos3-Nano, where it partly comes with extra boxes; on KITTI it does not exceed resampling for Qwen3-VL and Gemma4; and on the 2B model the margin over resampling holds in one of three cells, because a weaker model's resampled answer is itself worth more (section D.3). The margin is also concentrated where the object is only partly hidden: with more than 60% of it occluded, Qwen3-VL and Gemma4 gain no more than resampling (section D.5). **Precision.** It relocates boxes rather than refining them. **Absolute level.** A dedicated open-vocabulary detector with a tuned threshold still scores higher (section D.12); the case for PHANERO is the VLM a system already runs. **Comparators.** ControlMLLM is steered toward the baseline's own boxes rather than a user-supplied region, which may understate it. **Safety.** The patched decode alone can hurt — it makes up to 17.8% of a cell's questions worse — but keeping the original boxes caps that at 3.5% (table 14), and on unoccluded queries the repair lowers recall in no cell (sections C.5 and C.6).

AI USE STATEMENT

Generative AI tools were used to improve the clarity of prose based on author-created outlines, assist in anonymizing the code repository for double-blind review, support the graphical presentation of the architecture using TikZ, and iteratively review manuscript coherence under page-length constraints. Generative AI was not used to formulate the research questions or hypotheses, design the study or experiments, generate or alter experimental results, or develop the interpretation of the results based on the related literature. All AI-assisted outputs were reviewed and verified by the authors, who take full responsibility for the final content of the work.

ETHICS STATEMENT

This work uses previously released benchmark datasets and does not collect new human-subject data or use identity or sensitive-attribute labels. The datasets are used for research purposes subject to their applicable terms and licenses. Because the proposed intervention modifies localization outputs, inappropriate deployment in safety-critical or surveillance settings without task-specific validation could cause harm. We therefore report failure cases, output-format instability, cross-domain limitations, and conditions under which the intervention can degrade localization. The research was conducted in accordance with the ICLR Code of Ethics, and we are not aware of additional ethical concerns specific to the experiments reported here.

REPRODUCIBILITY STATEMENT

To support reproducibility and independent verification, we provide an anonymized code repository containing the implementation, reproducibility notes, provenance information, and audit scripts used to verify the numerical results reported in the main paper and supplementary material. The repository is available at `https://anonymous.4open.science/r/phanero-4D6E` and is structured to enable tracing and independent checking of the published results. Dataset construction and implementation details are provided in Section A, with hyperparameter analyses, controls, metric analyses, and full result breakdowns in Sections B to E.

REFERENCES

Jiayang Ao, Yanbei Jiang, Qiuhong Ke, and Krista A. Ehinger. Open-world amodal appearance completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6490–6499, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Ao_Open-World_Amodal_Appearance_Completion_CVPR_2025_paper.html.

Zijian Feng, Tianjiao Li, Zixiao Zhu, Hanzhang Zhou, Junlang Qian, Li Zhang, Jia Jim Deryl Chua, Lee Onn Mak, Gee Wah Ng, and Kezhi Mao. Fine-grained activation steering: Steering less, achieving more. In *International Conference on Learning Representations*, pp. 39421–39443, 2026. URL https://openreview.net/forum?id=guSVafqhrB.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3354–3361, 2012. URL https://doi.org/10.1109/CVPR.2012.6248074.

Gemma Team. Gemma 4 technical report, 2026. URL https://arxiv.org/abs/2607.02770. arXiv:2607.02770; checkpoint `google/gemma-4-12B-it`.

Jan Hosang, Rodrigo Benenson, Piotr Dollár, and Bernt Schiele. What makes for effective detection proposals? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(4):814–830, 2016. URL https://doi.org/10.1109/TPAMI.2015.2465908.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Drandreb Earl O. Juanico, Rowel O. Atienza, and Jeffrey Kenneth Go. Interpretable open-vocabulary referring object detection with reverse contrast attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 734–743, 2025. URL https://arxiv.org/abs/2507.19891.

Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. Your large vision-language model only needs a few attention heads for visual grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9339–9350, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Kang_Your_Large_Vision-Language_Model_Only_Needs_A_Few_Attention_Heads_CVPR_2025_paper.html.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Computer Vision – ECCV 2014*, pp. 740–755, 2014. URL https://link.springer.com/chapter/10.1007/978-3-319-10602-1_48.

Zhiwei Lin, Yongtao Wang, and Zhi Tang. Training-free open-ended object detection and segmentation via attention as prompts. In *Advances in Neural Information Processing Systems*, volume 37, pp. 69588–69606, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/80f48ffa8022773973a4a5cec7cce19c-Abstract-Conference.html.

Anish Madan, Neehar Peri, Shu Kong, and Deva Ramanan. Revisiting few-shot object detection with vision-language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 19547–19560, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/22b2067b8f6808126240320258 64c5a1-Abstract-Datasets_and_Benchmarks_Track.html.

NVIDIA. Cosmos 3: Omnimodal world models for physical AI, 2026. URL https://arxiv.org/abs/2606.02800. arXiv:2606.02800; checkpoint nvidia/Cosmos3-Nano.

Atin Pothiraj, Elias Stengel-Eskin, Jaemin Cho, and Mohit Bansal. CAPTURE: Evaluating spatial reasoning in vision language models via occluded object counting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8001–8010, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Pothiraj_CAPTURE_Evaluating_Spatial_Reasoning_in_Vision_Language_Models_via_Occluded_ICCV_2025_paper.html.

Qwen Team. Qwen3-VL technical report, 2025. URL https://arxiv.org/abs/2511.21631. arXiv:2511.21631; checkpoint Qwen3-VL-4B-Instruct.

Pau Rodriguez, Arno Blaas, Michal Klein, Luca Zappella, Nicholas Apostoloff, Marco Cuturi, and Xavier Suau. Controlling language and diffusion models by transporting activations. In *International Conference on Learning Representations*, pp. 89812–89855, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/df4f6e43446b1ee29c5a33d32c279f83-Abstract-Conference.html.

Gozde Sahin and Laurent Itti. HOOT: Heavy occlusions in object tracking benchmark. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 4830–4839, 2023. URL https://openaccess.thecvf.com/content/WACV2023/html/Sahin_HOOT_Heavy_Occlusions_in_Object_Tracking_Benchmark_WACV_2023_paper.html.

Anushka Sivakumar, Andrew Zhang, Zaber Ibn Abdul Hakim, and Chris Thomas. SteerVLM: Robust model control through lightweight activation steering for vision language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 23640–23665, 2025. URL https://aclanthology.org/2025.findings-emnlp.1285/.

Hao Vo, Khoa Vo, Phu Loc Nguyen, Sieu Tran, Duc Minh Nguyen, Ngo Xuan Cuong, Gladys Gawugah, Sreevenkata Anjani Tishita Godavarthi, Chase Rainwater, Nghi D. Q. Bui, Anh

Nguyen, Duy Minh Ho Nguyen, and Ngan Le. DriveSpatial: A benchmark for spatiotemporal intelligence in VLMs for autonomous driving. *arXiv preprint arXiv:2605.23176*, 2026. URL https://arxiv.org/abs/2605.23176.

Weixuan Wang, Jingyuan Yang, and Wei Peng. Semantics-adaptive activation intervention for LLMs via dynamic steering vectors. In *International Conference on Learning Representations*, pp. 79334–79351, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/c4d26a95fd83f8e590f81c54ae670b5d-Abstract-Conference.html.

Azmine Toushik Wasi, Wahid Faisal, Abdur Rahman, Mahfuz Ahmed Anik, Munem Shahriar, Mohsin Mahmud Topu, Sadia Tasnim Meem, Rahatun Nesa Priti, Sabrina Afroz Mitu, Md. Iqramul Hoque, Shahriyar Zaman Ridoy, Mohammed Eunus Ali, Majd Hawasly, Mohammad Raza, and Md Rizwan Parvez. SpatiaLab: Can vision–language models perform spatial reasoning in the wild? In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=fWWUPOb0CT.

Mingrui Wu, Xinyue Cai, Jiayi Ji, Jiale Li, Oucheng Huang, Gen Luo, Hao Fei, Guannan Jiang, Xiaoshuai Sun, and Rongrong Ji. ControlMLLM: Training-free visual prompt learning for multimodal large language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 45206–45234, 2024a. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/4fd96b997454b5b02698595df70fccaf-Abstract-Conference.html.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. ReFT: Representation finetuning for language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 63908–63962, 2024b. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/75008a0fba53bf13b0bb3b7bff986e0e-Abstract-Conference.html.

Shuxin Yang and Xinhan Di. OCC-MLLM-Alpha: Empowering multi-modal large language model for the understanding of occluded objects with self-supervised test-time learning. In *ECCV 2024 Workshop on Observing and Understanding Hands in Action (HANDS)*, 2024. URL https://arxiv.org/abs/2410.01861.

Fangcong Yin, Xi Ye, and Greg Durrett. LoFiT: Localized fine-tuning on LLM representations. In *Advances in Neural Information Processing Systems*, volume 37, pp. 9474–9506, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/122ea6470232ee5e79a2649243348005-Abstract-Conference.html.

Shanshan Zhang, Mingqian Ji, Yang Li, and Jian Yang. Imagine the unseen: Occluded pedestrian detection via adversarial feature completion. *arXiv preprint arXiv:2405.01311*, 2024. URL https://arxiv.org/abs/2405.01311.