Pointing-VLA: Typed Spatial Grounding Interfaces for Vision-Language-Action Manipulation

Xiwen Chen[1,2,*], Zelin Li[1,*,†], Zhiruo Zhou[1,3], Huiming Chen[4] Chenwei Wang[5], Xiaojun Zhu[1,†]

[1]SIGS, Tsinghua University, China; [2]Shanghai Jiao Tong University, China; [3]Wuhan University of Technology, China; [4]City University of Hong Kong, Hong Kong (China SAR); [5]AiDlab, Hong Kong (China SAR);
[*]Equal Contributions; [†]Correspondence: chiellini.lee@gmail.com; zhu.xiaojun@sz.tsinghua.edu.cn

Abstract

Vision-language-action (VLA) models often expose spatial grounding through autoregressive text coordinates or opaque action tokens, creating brittle interfaces between multimodal reasoning and robot execution. We present Pointing-VLA, a typed hidden-state spatial readout built on Embodied-R1. Geometry-specific heads predict normalized points, object-functional grounding (OFG) heatmaps, and visual trajectories without serializing geometry as text. For the evaluated Bridge/WidowX and physical pick-place deployments, an explicit execution contract assigns PICK to source-conditioned OFG and PLACE to Pointing, providing direct stage-aligned spatial targets. Pointing-VLA achieves SOTA performance on Bridge/WidowX, averaging 72.9% across the evaluated four-task set without Bridge-specific finetuning under collision-enabled CuRobo execution. Pointing and OFG show complementary strengths across native and cross-dataset evaluations. The OFG/contact readout transfers to NORA-1.5, preserving or improving success while reducing recorded controller time by more than 20×; typed heads are also 6.68–6.90× faster than Embodied-R1 text decoding on a shared external suite. When integrated as spatial guidance for a $\pi_{0.5}$ action policy, Pointing-VLA raises autonomous real-robot success from 52.7% to 80.7% across three visual contexts. These results establish typed spatial readouts as an efficient, inspectable interface between embodied reasoning and robot execution.

Introduction

Large vision-language-action (VLA) models can connect observations, language instructions, and robot behavior, but their output interface remains a bottleneck. Many manipulation subtasks first require an explicit spatial commitment: the point to touch, the functional part to use, or the short visual trace an executor should follow. Autoregressive text coordinates are convenient for prompting, yet they make geometry an afterthought: the model must spend tokens spelling out numbers, the evaluator must parse strings back into pixels, and dense part-level evidence is discarded before execution. Direct action tokens avoid parsing, but they also hide the spatial rationale that a robot stack or human auditor may need to inspect.

Figure 1 separates two interface failures observed in representative simulator cases: text generation can fail before producing valid geometry, and a syntactically valid coordinate can still be spatially wrong. Parsing is therefore an additional execution dependency rather than a guarantee of correct grounding.

The central claim of this paper is that embodied grounding should be treated as typed interface prediction rather than text-coordinate generation. Referring points, functional regions, and visual trajectories are related but not identical targets: a point is sparse, an affordance is often a dense part-level region, and a trajectory is temporally structured. Collapsing them into one output distribution can create negative interference even when they share a visual-language backbone, while serializing all of them as text makes downstream execution slower and less stable.

This formulation is grounded in the vision-for-action and population-readout view of computation: downstream readouts decode spatial variables from distributed internal activity instead of serializing them through the system that represents them (Goodale and Milner 1992; Andersen and Buneo 2002; Cisek 2007). Pointing-VLA realizes this principle with separate point, affordance-heatmap, and trajectory heads that emit each target in the geometry consumed by robot-side execution.

We propose Pointing-VLA, a lightweight typed grounding framework built on an embodied vision-language backbone. Rather than asking the backbone to say coordinates, Pointing-VLA reads spatial intent from multimodal hidden states. A compact adapter and learned queries feed point, object-functional grounding (OFG), and visual trajectory generation (VTG) decoders. Deployment contracts bind each structured execution slot to the geometry it requires; the primary pick-place scaffold uses source-conditioned OFG for PICK and Pointing for PLACE. The resulting spatial targets connect embodied reasoning to downstream wrappers, planners, and executors through deterministic, stage-aligned contracts. Experiments evaluate the

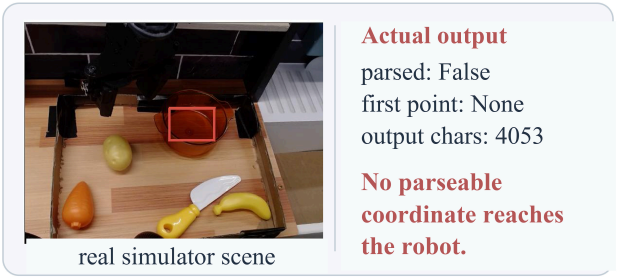

Figure 1: Representative text-coordinate failures. **Left:** generation yields no parseable coordinate. **Right:** `[222, 172]` is parseable but lies outside the target region.

interface from native spatial grounding through head specialization, cross-backbone transfer, and robot deployment. On Bridge/WidowX, the fixed OFG PICK and Pointing PLACE composition achieves SOTA performance, averaging 72.9% across the evaluated four-task set without Bridge-specific finetuning under collision-enabled CuRobo execution. Cross-dataset results reveal complementary Pointing and OFG geometries, while NORA-1.5 transfer preserves or improves success with more than 20× shorter recorded controller time. Real-robot deployment completes the evidence chain from hidden-state readout to physical execution.

Our contributions are:

- We formulate embodied grounding as typed interface prediction, replacing serialized text coordinates with geometry-specific robot-facing readouts.
- We introduce Pointing, OFG, and VTG hidden-state readouts with explicit execution contracts, including a fixed source-conditioned OFG-PICK/Pointing-PLACE deployment that aligns each stage with its required geometry.
- We validate the interface through native and cross-dataset grounding, runtime and cross-backbone transfer, Bridge/WidowX evaluation, and autonomous real-robot deployment.

Related Work

Generalist robot policies and VLA systems adapt large pretrained models to robot action spaces, from embodied multimodal models and Robotics Transformers to cross-embodiment datasets and open robot policies (Driess et al. 2023; Brohan et al. 2023; Zitkovich et al. 2023; Open X-Embodiment Collaboration et al. 2024; Octo Model Team et al. 2024; Kim et al. 2025; Black et al. 2025). This line of work has improved policy learning, action parameterization, and inference throughput, but it mainly asks how perception and language should become actions. Pointing-VLA asks a complementary interface question: what spatial representation should be exposed before final execution?

Spatial structure has long been useful for manipulation. Neuroscience distinguishes object recognition from action-facing visuomotor representations, including posterior-parietal intention maps and competing affordance-like action candidates (Goodale and Milner 1992; Andersen and Buneo 2002; Cisek 2007). Robot learning makes a compatible engineering move: Transporter Networks preserve spatial feature maps for pick-and-place displacements, CLIPort combines semantic "what" and spatial "where" pathways, and PerAct discretizes 3D voxel actions for language-conditioned manipulation (Zeng et al. 2021; Shridhar, Manuelli, and Fox 2022, 2023). Recent grounding-centric VLA work extends this idea to embodied reasoning: RoboPoint predicts spatial affordance keypoints from language instructions (Yuan et al. 2025a), FSD studies seeing-to-doing affordance grounding (Yuan et al. 2026), and Embodied-R1 formalizes embodied pointing across REG, relational-region grounding, OFG, and VTG (Yuan et al. 2025b). Affordance datasets such as AGD20K further emphasize that actionable regions are often object parts rather than whole-object centers (Luo et al. 2022). These systems show that spatial outputs can bridge VLM reasoning and robot behavior, but they leave open which interface an executor should consume. Pointing-VLA studies this interface choice directly: sparse referring points, dense functional heatmaps, and visual waypoint traces are decoded in their native geometries rather than serialized as text coordinates.

This interface complements action-sequence policies such as Diffusion Policy and ACT (Chi et al. 2023; Zhao et al. 2023) by exposing typed spatial targets before low-level action generation. In the evaluated pick-place systems, the execution phase itself supplies a stable contract: functional contact is decoded for PICK and a compact target point is decoded for PLACE.

Method

Hidden-State Spatial Readout

Pointing-VLA operationalizes typed interface prediction by decoding geometry-specific robot targets directly from shared multimodal hidden states. Figure 2 summarizes this hidden-state readout and its execution-facing spatial outputs.

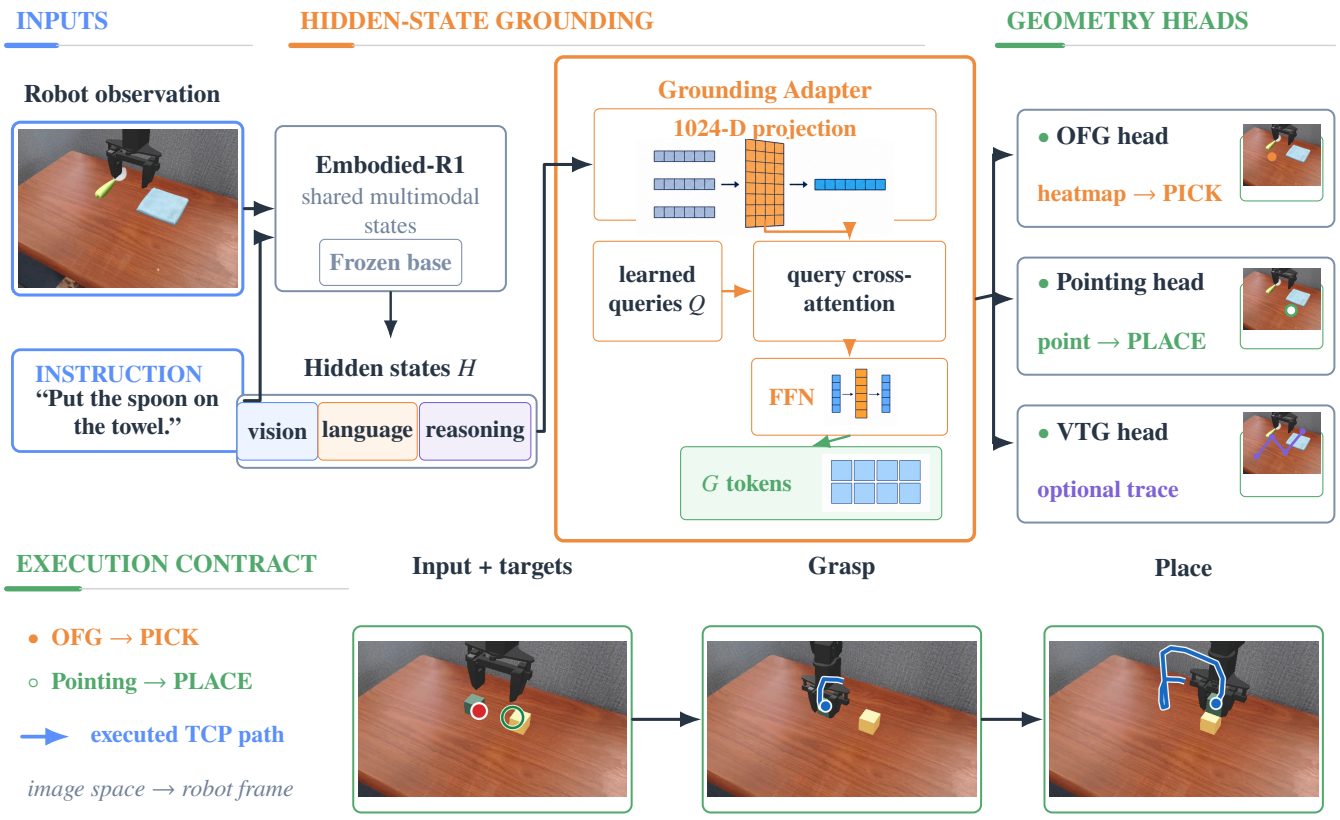

Figure 2: Pointing-VLA architecture and typed interfaces. Embodied-R1 hidden states feed point, OFG/contact-heatmap, and VTG-waypoint heads; fixed PICK/PLACE slots use source-conditioned OFG/Pointing, and a wrapper maps image geometry to robot coordinates.

Let $e \in \{\mathrm{pt}, \mathrm{ofg}, \mathrm{vtg}\}$ index a spatial expert and let $H^e_{\mathrm{mm}} = \{h^e_i\}_{i=1}^N$ denote its Embodied-R1 multimodal states. Pointing-VLA reuses one backbone architecture, base weights, and Grounding Adapter body, while each specialist retains its own LoRA, learned queries, and decoder state. For the point and OFG experts, the shared adapter body projects the sequence into a grounding space and reads it with expert-specific task-conditioned queries:

$$X_e = W^e_h H^e_{\mathrm{mm}} + P_e,$$
$$A_e = \mathrm{softmax}\left(\frac{(Q_e W^e_Q)(X_e W^e_K)^\top}{\sqrt{d}}\right),$$
$$G_e = \mathrm{FFN}_e(A_e X_e W^e_V), \qquad g_e = \mathrm{Pool}(G_e),$$

where P_e supplies sequence position, G_e preserves a fixed set of grounded tokens, and g_e is its global summary. The point and OFG experts use eight grounding queries. The strongest VTG expert instead lets learned temporal queries attend directly to the full multimodal sequence, avoiding a fixed grounding-token bottleneck.

Spatial Interfaces and Deployment Contracts

Learned Spatial Heads. The Pointing decoder uses a learned pointer query q_{pt} to select the grounded evidence needed for one image location:

$$A_{\mathrm{pt}} = \mathrm{softmax}\left(\frac{(q_{\mathrm{pt}} W_Q)(G_{\mathrm{pt}} W_K)^\top}{\sqrt{d}}\right),$$
$$z_{\mathrm{pt}} = A_{\mathrm{pt}}(G_{\mathrm{pt}} W_V), \qquad (\hat{x}, \hat{y}) = \sigma(W_p\, \mathrm{LN}(z_{\mathrm{pt}}) + b_p).$$

Thus $(\hat{x}, \hat{y}) \in [0,1]^2$ is emitted directly by the learned head, without generating or parsing coordinate text. For an image of width W and height H, the corresponding pixel is

$$(\hat{u}, \hat{v}) = \big((W-1)\hat{x}, (H-1)\hat{y}\big).$$

The prediction is evaluated with point-in-bbox (PIB) or point-in-mask (PIM).

OFG preserves spatial extent instead of collapsing the source to one token. Let F_{vis} be the visual tower's 2D feature map. The global grounded summary modulates

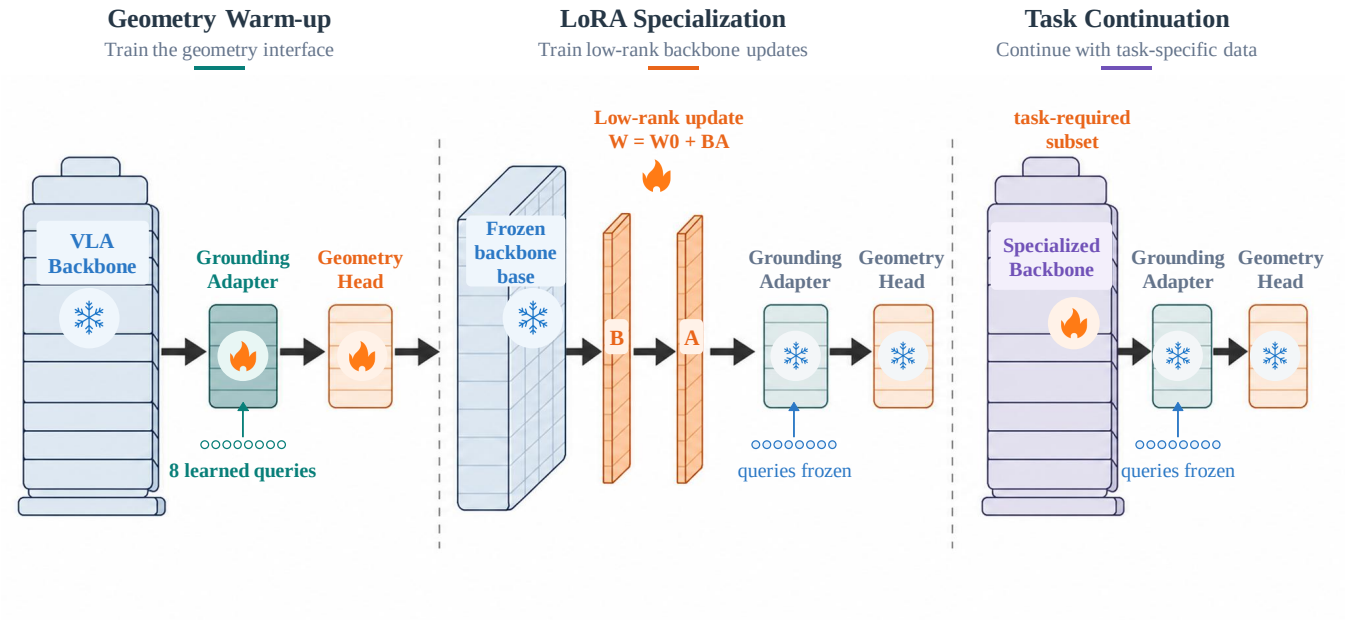

Figure 3: Stage-wise training. Geometry warm-up trains the adapter, eight queries, and geometry head; LoRA specialization updates only low-rank backbone parameters; task continuation updates only the required subset. Snowflakes and flames mark frozen and trainable modules.

this map through FiLM before a convolutional heatmap decoder (Luo et al. 2022):

$$
\begin{aligned}
F'_{\text{ofg}} &= \gamma(g_{\text{ofg}}) \odot F_{\text{vis}} + \beta(g_{\text{ofg}}),\\
\hat{H}_{\text{ofg}} &= f_{\text{hm}}(F'_{\text{ofg}}),\\
(i^*, j^*) &= \arg\max_{i,j} \hat{H}_{\text{ofg}}[i,j],\\
\hat{p}_{\text{ofg}} &= \left(\frac{j^*}{W_o - 1}, \frac{i^*}{H_o - 1}\right).
\end{aligned}
$$

Here $\hat{H}_{\text{ofg}}$ contains heatmap logits, and its normalized peak is the execution-facing functional contact.

For VTG, learned waypoint queries q_t^τ with temporal encodings attend to the multimodal sequence in order:

$$
\begin{aligned}
z_t^\tau &= \text{Attn}(q_t^\tau, H_{\text{mm}}^{\text{vtg}}),\\
\hat{\tau}_t &= \sigma(W_\tau z_t^\tau + b_\tau), \qquad t = 1, \ldots, T.
\end{aligned}
$$

We use $T = 8$ normalized image-space waypoints and report RMSE, average displacement error (ADE), and final displacement error (FDE). These waypoints describe a visual trace rather than low-level robot actions.

Structured PICK/PLACE Contracts. Training follows target geometry: point, region, and visual-trace samples supervise the corresponding learned heads. For pick-place deployment, the scaffold constructs slot-specific queries from instruction ℓ, source description c_{src}, and goal description c_{goal}:

$$
\begin{aligned}
q_{\text{pick}} &= \mathcal{T}_{\text{pick}}(\ell, c_{\text{src}}),\\
q_{\text{place}} &= \mathcal{T}_{\text{place}}(\ell, c_{\text{goal}}),\\
\hat{y}_{\text{pick}} &= F_{\text{ofg}}(H_{\text{mm}}, G, q_{\text{pick}}),\\
\hat{y}_{\text{place}} &= F_{\text{pt}}(H_{\text{mm}}, G, q_{\text{place}}).
\end{aligned}
$$

This assignment is deterministic: source-conditioned OFG emits the functional PICK contact, while Pointing emits the PLACE target. VTG serves annotated waypoint requests; the reported pick-place deployments use the fixed OFG-PICK/Pointing-PLACE contract. An external wrapper converts image-space outputs to robot-frame targets. With calibrated depth D, camera intrinsics K, and camera-to-robot transform $T_{r \leftarrow c}$, this boundary can be written as

$$
\begin{aligned}
\tilde{p} &= [\hat{u}, \hat{v}, 1]^\top,\\
X_c &= D(\hat{u}, \hat{v}) K^{-1} \tilde{p},\\
\bar{X}_r &= T_{r \leftarrow c} \bar{X}_c,
\end{aligned}
$$

where $\bar{X}$ denotes homogeneous coordinates. The existing executor then consumes X_r; the learned heads remain image-space predictors.

Training Objective

The experts use a weighted spatial objective:

$$
\mathcal{L} = \lambda_{\text{pt}}\mathcal{L}_{\text{pt}} + \lambda_{\text{box}}\mathcal{L}_{\text{box}} + \lambda_{\text{ofg}}\mathcal{L}_{\text{ofg}} + \lambda_{\text{vtg}}\mathcal{L}_{\text{vtg}} + \lambda_{\text{aux}}\mathcal{L}_{\text{aux}}.
$$

Point and trajectory coordinates use SmoothL1, while OFG mask or keypoint supervision is converted to a dense target H^*:

$$
\begin{aligned}
\mathcal{L}_{\text{pt}} &= \tfrac{1}{K}\textstyle\sum_k \text{SmoothL1}(\hat{p}_k, p_k),\\
\mathcal{L}_{\text{ofg}} &= \tfrac{1}{|\Omega|}\textstyle\sum_{u \in \Omega} \ell_{\text{hm}}(\hat{H}_{\text{ofg}}(u), H^*(u)),\\
\mathcal{L}_{\text{vtg}} &= \tfrac{1}{T}\textstyle\sum_t \text{SmoothL1}(\hat{\tau}_t, \tau_t).
\end{aligned}
$$

Here ℓ_{hm} is mean-squared error for Gaussian keypoint targets and class-balanced binary cross-entropy with

logits for dense affordance masks. Language-modeling and legacy reconstruction terms are inactive; optimization focuses on adapters, heads, and LoRA specialization (Hu et al. 2021).

For source-conditioned OFG specialization, we keep the Embodied-R1 base, Pointing/VTG query and decoder modules, and executor frozen while jointly updating the OFG LoRA, shared Grounding Adapter body, OFG learned query, and heatmap decoder. Dense source-mask reconstruction is combined with source-over-distractor ranking and distractor suppression. This adaptation teaches OFG to preserve functional contact while resolving the source instance directly in the PICK readout.

The final pointing model additionally applies GRPO-style group-normalized refinement. With policy center $\mu = \hat{p}$ and learned $\sigma = \exp(s)$, samples are

$$a_i = \mathrm{clip}(\mu + \sigma\epsilon_i, 0, 1), \qquad \epsilon_i \sim \mathcal{N}(0, I).$$

For target box B^*, reward and group-normalized advantage are

$$\begin{aligned} r_i &= \begin{cases} 1, & a_i \in B^*, \\ 0.5\max(0, 1 - d_i/\rho), & a_i \notin B^*, \end{cases} \\ A_i &= \frac{r_i - \mathrm{mean}_j(r_j)}{\mathrm{std}_j(r_j) + \epsilon}. \end{aligned}$$

With $\pi_i = \pi(a_i \mid \mu, \sigma)$ and Gaussian reference KL D_{ref}, the objective is

$$\begin{aligned} \mathcal{L}_{\mathrm{GRPO}} &= -\tfrac{1}{M}\textstyle\sum_{i=1}^{M} \mathrm{sg}(A_i)\log\pi_i \\ &\quad + \lambda_{\mathrm{sft}}\mathrm{SmoothL1}(\mu, p^*) \\ &\quad + \lambda_{\mathrm{kl}} D_{\mathrm{ref}} + \lambda_\sigma \mathcal{L}_\sigma. \end{aligned}$$

Here sg stops advantage gradients and p^* is the supervised point target; refinement acts directly on the continuous spatial head.

Experiments

Evaluation Questions and Protocol

Four questions structure the evaluation.

Q1: Geometry. Do typed heads preserve task-appropriate geometry?

Q2: Phase specialization. Does source-conditioned OFG resolve PICK source selection while preserving the benefits of a fixed OFG-PICK/Pointing-PLACE contract?

Q3: Transfer and efficiency. Does the readout transfer across datasets and VLA backbones while reducing inference cost?

Q4: Deployment. Do the gains persist in simulation and real-robot deployment?

Native and external evaluations answer Q1; source-conditioning and interface-composition ablations answer Q2. These controlled comparisons isolate component-level grounding quality, while the deployment studies measure complete robot-task success. Cross-dataset and cross-backbone transfer plus known-head runtime address Q3; controlled simulation and real-robot studies on shared task sets and executor families address Q4.

Evaluation Protocol and Metrics. We evaluate each output at the level of the interface it is designed to serve. Pointing and OFG predictions are converted to image coordinates and scored by point-in-mask (PIM) or point-in-box (PIB) sample success; VTG is measured by normalized trajectory root mean squared error (RMSE), average displacement error (ADE), and final displacement error (FDE). Robot studies require full task completion. Within each experiment, all compared methods use the same samples or episode identities, ensuring that measured differences arise from the spatial interface or adaptation strategy. Experiments used 40-GB NVIDIA A100 and dual 48-GB RTX A6000 GPUs; complete hardware, software, and seed records appear in the Supplement.

Native Spatial-Head Results

Table 1 answers the first question. Pointing-VLA matches Embodied-R1 on sparse referring-point grounding at 64.3%, while OFG raises full Part-Affordance-2K accuracy from 40.9% to 57.3%. This 16.4-point improvement demonstrates geometry specialization: Pointing preserves sparse target accuracy, whereas OFG substantially strengthens part-level functional grounding by retaining the spatial extent required for contact prediction.

Model	VABench-P acc. ↑	Part-Afford.2K acc. ↑
GPT-4o	9.3	10.2
ASMv2	10.1	13.8
RoboBrain	7.0	25.3
Qwen2.5-VL	9.9	23.4
RoboPoint (Yuan et al. 2025a)	19.1	27.6
FSD (Yuan et al. 2026)	61.8	9.6
Embodied-SFT (Yuan et al. 2025b)	50.5	39.4
Embodied-R1 (Yuan et al. 2025b)	63.7	40.9
Pointing-VLA (ours)	**64.3**	**57.3**

Table 1: Native point-in-region grounding accuracy (%). Pointing-VLA uses its Pointing head for VABench-P and its OFG head for the full Part-Affordance-2K run.

Sequential geometry. VTG completes the native evaluation for ordered spatial outputs. On 300 VABench-V examples, the trajectory readout records 0.1042 RMSE, 0.1368 ADE, and 0.1493 FDE in normalized image coordinates. These measures evaluate the waypoint sequence and its terminal target rather than collapsing trajectory quality into a single point-in-region score.

Mechanism and Fixed-Contract Ablations

PICK requires both source-instance disambiguation and a contact location within the selected object. Source-conditioned OFG preserves this dense functional extent and exposes its peak as the grasp cue. PLACE instead requires a compact target in the destination region, for which Pointing provides a direct

normalized coordinate. The fixed OFG-PICK/Pointing-PLACE contract therefore aligns each execution stage with the geometry it requires, without introducing an inference-time selector.

Mechanism analysis. Panel (a) identifies the effective capacity regime of the spatial readout. Eight queries and rank-32 LoRA produce the strongest PIB within their respective sweeps, while GRPO refinement raises PIB from 61.3% to 64.3%. The non-monotonic query and rank trends demonstrate that performance is governed by the spatial-readout design rather than parameter count alone.

Fixed-contract validation. Panel (b) validates the phase-aligned assignment: the deployed contract completes 70/96 episodes (72.9%) under collision-enabled CuRobo execution, outperforming the strongest alternative composition by 14.6 percentage points while completing all Stack episodes and 22/24 Eggplant episodes.

The controlled frozen multicolor Stack evaluation further verifies this mechanism. Source-conditioned OFG selects the instructed source in all 48 online cases and completes 43/48 episodes, versus 40/48 for Attention PICK under identical Pointing PLACE targets. Perfect source selection establishes OFG as a direct PICK readout that resolves the instructed source while preserving functional contact grounding.

External Diagnostics and Cross-Backbone Transfer

Cross-Dataset Geometry. Table 3 asks whether performance follows the geometry exposed by each interface. These diagnostics standardize outputs as points and score PIM/PIB, isolating spatial grounding from downstream control. Pointing is strongest on RefCOCO and RoboAff, whereas OFG is strongest on AGD20K and raises full Part-Affordance-2K PIM from 40.9% with Embodied-R1 text generation to 57.3%. The crossover supports expert-specific output geometries rather than one serialized coordinate interface. The Supplement provides qualitative OFG heatmaps.

Pointing, OFG, and Embodied-R1 text generation are evaluated on the same cross-dataset samples. Each trained head emits one point, while text generation can emit multiple points; sample-level success therefore keeps the comparison independent of candidate count. Dataset composition and deterministic split construction are detailed in the Supplement and released evaluation code. This shared point protocol enables a controlled cross-dataset comparison of spatial interfaces under identical output and evaluation rules.

Suite	Metric	Pointing-VLA Point	Pointing-VLA OFG	Embodied-R1 Text Generation
RefCOCO	PIM	**40.7%**	22.2%	10.6%
RefCOCO	PIB	**56.1%**	36.0%	25.9%
AGD20K	PIM	53.9%	**64.4%**	39.9%
Part-Aff.2K	PIM	–	**57.3%**	40.9%
RoboAff.	PIM	**11.5%**	9.8%	2.4%

Table 3: Cross-dataset PIM/PIB accuracy (%).

System	Total (s)	Per sample (s)	Speedup
OFG head	1,274.61	0.434	6.90×
Pointing head	1,316.29	0.448	6.68×
Embodied-R1 text decoder	8,798.06	2.993	1.00×

Method	Pose	Success	Time (s)	Speedup
NORA base	Horizontal	100.0	107.11	1.0×
+ OFG/contact	Horizontal	**100.0**	**4.85**	**22.1**×
NORA base	Laid vert.	89.0	111.57	1.0×
+ OFG/contact	Laid vert.	**95.0**	**5.44**	**20.5**×

Table 4: Known-head runtime (top) and NORA-1.5 transfer (bottom).

Interface-Level Runtime. Table 4 measures inference from multimodal hidden states to typed spatial outputs under the shared external protocol. The geometric readouts are 6.68–6.90× faster than autoregressive text decoding, establishing the interface-level efficiency gained by eliminating coordinate serialization, token-by-token generation, and post-hoc parsing.

NORA-1.5 Transfer. Table 4 shows that the frozen NORA-1.5 backbone with the transferred OFG/contact readout preserves horizontal-pose success, improves laid-vertical success from 89.0% to 95.0%, and cuts recorded controller time by more than 20× under the shared wrapper.

System	Spoon	Carrot	Stack	Eggpl.	Avg.
Octo-S	47.2	9.7	4.2	56.9	30.0
SpatialVLA (FT)	16.7	25.0	29.2	**100.0**	42.7
SoFar	55.5	56.9	62.5	40.2	53.8
MemoryVLA	75.0	75.0	37.5	**100.0**	71.9
Embodied-R1 + CuRobo	20.8	45.8	62.5	79.2	52.1
Pointing-VLA + CuRobo	50.0	50.0	**100.0**	91.7	**72.9**

Table 5: Bridge/WidowX success (%; 24 episodes/task for Embodied-R1 and Pointing-VLA). Pointing-VLA uses fixed OFG-PICK/Pointing-PLACE with collision-enabled CuRobo.

Deployment Studies

Table 5 evaluates four Bridge/WidowX manipulation structures. The Pointing-VLA condition contains 24 episodes per task, and success requires complete task execution. Its fixed OFG PICK and Pointing PLACE composition reaches 72.9% with collision checking active in every rollout; CuRobo completes all planned motion segments without planner failure or fallback. The tasks stress distinct spatial bottlenecks: thin-object

(a) Spatial-head configuration

Component	Settings	PIB
Query tokens	4 / 8 / 16	44.7 / **50.3** / 45.7
LoRA rank	none / 16 / 32 / 64	45.3 / 47.3 / **53.3** / 50.3
Refinement	SFT / GRPO	61.3 / **64.3**

(b) Interface-composition ablation

Configuration	Spoon on towel	Carrot on plate	Stack cube	Eggplant basket	Avg.
Attention → Attention	16.7	41.7	100.0	62.5	55.2
Attention → Pointing	12.5	50.0	100.0	70.8	58.3
OFG → Attention	16.7	54.2	16.7	58.3	36.5
OFG → Pointing	**50.0**	**50.0**	**100.0**	**91.7**	**72.9**

Table 2: Mechanism and fixed-contract results. (a) VABench-P PIB (%) across spatial-head configurations. (b) Bridge/WidowX success (%) for fixed PICK/PLACE interfaces (24 episodes/task); the final row is the deployed OFG-PICK/Pointing-PLACE contract with collision-enabled CuRobo.

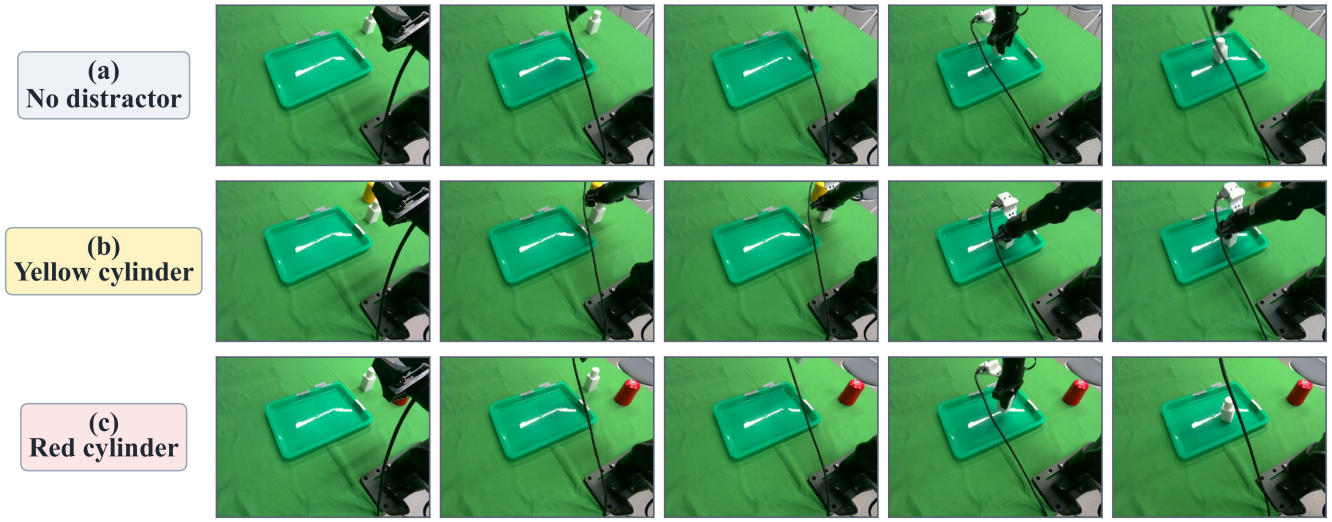

Figure 4: Representative successful real-robot rollouts across three visual contexts, each ordered from initialization to stable release.

contact for Spoon, surface targeting for Carrot, source-instance disambiguation for Stack, and container placement for Eggplant. The source-conditioned OFG expert preserves 100.0% Stack success as the direct PICK readout, while Pointing supplies an explicit placement target. Together, these results demonstrate reliable four-task deployment through a collision-aware motion-planning backend.

Real-Robot Deployment

Scene success	$\pi_{0.5}$	$\pi_{0.5}$ + P-VLA	Δ
No distractor	20/50 (40.0)	**36/50 (72.0)**	+32.0
Yellow cylinder	26/50 (52.0)	**42/50 (84.0)**	+32.0
Red cylinder	33/50 (66.0)	**43/50 (86.0)**	+20.0
All three scenes	79/150 (52.7)	**121/150 (80.7)**	**+28.0**

Failure stage	Grasp	Transfer	Tray	Uncertain
Count (/150)	47→16	8→9	13→4	3→0
Δ (pp)	**-20.7**	+0.7	-6.0	-2.0

Table 6: Autonomous PiPER outcomes. Success is count/trials (%); failures show baseline→P-VLA for pre-lift grasp, post-grasp transfer/drop, tray arrival without upright release, and uncertain cases.

We deploy Pointing-VLA on an AgileX PiPER and compare it with a $\pi_{0.5}$ baseline (Physical Intelligence et al. 2025) through the same dual-camera perception-to-control stack. Pointing-VLA retains $\pi_{0.5}$ as the action-generating policy and inserts typed spatial guidance before execution. During PICK, the OFG head predicts an affordance heatmap over the lower rectangular part of the white composite object, and the external wrapper converts its peak into a robot-frame grasp cue. During PLACE, the Pointing head predicts a normalized image-space placement coordinate inside the green tray, which the wrapper converts into the placement target. The same deterministic OFG-PICK/Pointing-PLACE contract is used throughout all three visual contexts: no distractor, a yellow-cylinder distractor, and a red-cylinder distractor. We evaluate both systems over 50 trials per context. The reported endpoint is full task completion: the designated part must be grasped and lifted, positioned upright in the tray, and stably released.

Across all three visual contexts, typed spatial guidance consistently improves autonomous completion over the action-policy baseline. Failure-stage analysis confirms that the gains concentrate at the intended control boundaries: pre-lift grasp failures fall from 47 to 16, and tray-arrival failures fall from 13 to 4. Post-grasp transfer remains the primary residual bottleneck. These reductions verify the stage-aligned contribution of OFG-PICK and Pointing-PLACE guidance; Figure 4 traces representative successes from approach through stable release.

Discussion and Conclusion

Pointing-VLA treats referring points, functional regions, and waypoint traces as distinct robot-facing geometries. Embodied-R1 provides shared multimodal states, while lightweight geometry heads expose normalized points, heatmaps, and trajectories directly. Explicit deployment contracts align these outputs with execution stages and preserve the geometry required by each robot-facing slot. Together, the native, cross-dataset, runtime, transfer, and deployment results establish geometry-appropriate typed readouts as an efficient and inspectable interface between embodied VLM reasoning and robot execution.

The fixed pick/place scaffold provides a controlled deployment setting that verifies the effectiveness of typed spatial guidance. Task-dependent variation, most visible on spoon-on-towel, identifies closed-loop geometry-aware execution as the next leverage point. Future work will extend this interface with visual correction, broader structured task decompositions, and cross-robot transfer.

References

Andersen, R. A.; and Buneo, C. A. 2002. Intentional Maps in Posterior Parietal Cortex. *Annual Review of Neuroscience*, 25: 189–220.

Black, K.; Brown, N.; Driess, D.; Esmail, A.; Equi, M.; Finn, C.; Fusai, N.; Groom, L.; Hausman, K.; Ichter, B.; Jakubczak, S.; Jones, T.; Ke, L.; Levine, S.; Li-Bell, A.; Mothukuri, M.; Nair, S.; Pertsch, K.; Shi, L. X.; Tanner, J.; Vuong, Q.; Walling, A.; Wang, H.; and Zhilinsky, U. 2025. π_0: A Vision-Language-Action Flow Model for General Robot Control. In *Proceedings of Robotics: Science and Systems*.

Brohan, A.; Brown, N.; Carbajal, J.; Chebotar, Y.; Dabis, J.; Finn, C.; Gopalakrishnan, K.; Hausman, K.; Herzog, A.; Hsu, J.; Ibarz, J.; Ichter, B.; Irpan, A.; Jackson, T.; Jesmonth, S.; Joshi, N.; Julian, R.; Kalashnikov, D.; Kuang, Y.; Leal, I.; Lee, K.-H.; Levine, S.; Lu, Y.; Malla, U.; Manjunath, D.; Mordatch, I.; Nachum, O.; Parada, C.; Peralta, J.; Perez, E.; Pertsch, K.; Quiambao, J.; Rao, K.; Ryoo, M.; Salazar, G.; Sanketi, P.; Sayed, K.; Singh, J.; Sontakke, S.; Stone, A.; Tan, C.; Tran, H.; Vanhoucke, V.; Vega, S.; Vuong, Q.; Xia, F.; Xiao, T.; Xu, P.; Xu, S.; Yu, T.; and Zitkovich, B. 2023. RT-1: Robotics Transformer for Real-World Control at Scale. In *Proceedings of Robotics: Science and Systems*.

Chi, C.; Xu, Z.; Feng, S.; Cousineau, E.; Du, Y.; Burchfiel, B.; Tedrake, R.; and Song, S. 2023. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. In *Robotics: Science and Systems*.

Cisek, P. 2007. Cortical Mechanisms of Action Selection: The Affordance Competition Hypothesis. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1485): 1585–1599.

Driess, D.; Xia, F.; Sajjadi, M. S. M.; Lynch, C.; Chowdhery, A.; Ichter, B.; Wahid, A.; Tompson, J.; Vuong, Q.; Yu, T.; Huang, W.; Chebotar, Y.; Sermanet, P.; Duckworth, D.; Levine, S.; Vanhoucke, V.; Hausman, K.; Toussaint, M.; Greff, K.; Zeng, A.; Mordatch, I.; and Florence, P. 2023. PaLM-E: An Embodied Multimodal Language Model. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, 8469–8488. PMLR.

Goodale, M. A.; and Milner, A. D. 1992. Separate Visual Pathways for Perception and Action. *Trends in Neurosciences*, 15(1): 20–25.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685.

Kim, M. J.; Pertsch, K.; Karamcheti, S.; Xiao, T.; Balakrishna, A.; Nair, S.; Rafailov, R.; Foster, E.; Lam, G.; Sanketi, P.; Vuong, Q.; Kollar, T.; Burchfiel, B.; Tedrake, R.; Sadigh, D.; Levine, S.; Liang, P.; and Finn, C. 2025. OpenVLA: An Open-Source Vision-Language-Action Model. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, 2679–2713. PMLR.

Luo, H.; Zhai, W.; Zhang, J.; Cao, Y.; and Tao, D. 2022. Learning Affordance Grounding from Exocentric Im-

ages. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2252–2261.

Octo Model Team; Ghosh, D.; Walke, H.; Pertsch, K.; Black, K.; Mees, O.; Dasari, S.; Hejna, J.; Kreiman, T.; Xu, C.; Luo, J.; Tan, Y. L.; Chen, L. Y.; Sanketi, P.; Vuong, Q.; Xiao, T.; Sadigh, D.; Finn, C.; and Levine, S. 2024. Octo: An Open-Source Generalist Robot Policy. In *Robotics: Science and Systems*.

Open X-Embodiment Collaboration; O'Neill, A.; Rehman, A.; Gupta, A.; Maddukuri, A.; Gupta, A.; Padalkar, A.; Lee, A.; Pooley, A.; Gupta, A.; et al. 2024. Open X-Embodiment: Robotic Learning Datasets and RT-X Models. In *Proceedings of the IEEE International Conference on Robotics and Automation*.

Physical Intelligence; Black, K.; Brown, N.; Darpinian, J.; Dhabalia, K.; Driess, D.; Esmail, A.; Equi, M.; Finn, C.; Fusai, N.; Galliker, M. Y.; Ghosh, D.; Groom, L.; Hausman, K.; Ichter, B.; Jakubczak, S.; Jones, T.; Ke, L.; LeBlanc, D.; Levine, S.; Li-Bell, A.; Mothukuri, M.; Nair, S.; Pertsch, K.; Ren, A. Z.; Shi, L. X.; Smith, L.; Springenberg, J. T.; Stachowicz, K.; Tanner, J.; Vuong, Q.; Walke, H.; Walling, A.; Wang, H.; Yu, L.; and Zhilinsky, U. 2025. $\pi_{0.5}$: A Vision-Language-Action Model with Open-World Generalization. arXiv:2504.16054.

Shridhar, M.; Manuelli, L.; and Fox, D. 2022. CLIPort: What and Where Pathways for Robotic Manipulation. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, 894–906. PMLR.

Shridhar, M.; Manuelli, L.; and Fox, D. 2023. Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation. In *Proceedings of the 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, 785–799. PMLR.

Yuan, W.; Duan, J.; Blukis, V.; Pumacay, W.; Krishna, R.; Murali, A.; Mousavian, A.; and Fox, D. 2025a. RoboPoint: A Vision-Language Model for Spatial Affordance Prediction in Robotics. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, 4005–4020. PMLR.

Yuan, Y.; Cui, H.; Chen, Y.; Dong, Z.; Ni, F.; Kou, L.; Liu, J.; Li, P.; Zheng, Y.; and Hao, J. 2026. From Seeing to Doing: Bridging Reasoning and Decision for Robotic Manipulation. arXiv:2505.08548.

Yuan, Y.; Cui, H.; Huang, Y.; Chen, Y.; Ni, F.; Dong, Z.; Li, P.; Zheng, Y.; Tang, H.; and Hao, J. 2025b. Embodied-R1: Reinforced Embodied Reasoning for General Robotic Manipulation. arXiv:2508.13998.

Zeng, A.; Florence, P.; Tompson, J.; Welker, S.; Chien, J.; Attarian, M.; Armstrong, T.; Krasin, I.; Duong, D.; Sindhwani, V.; and Lee, J. 2021. Transporter Networks: Rearranging the Visual World for Robotic Manipulation. In *Proceedings of the 4th Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, 726–747. PMLR.

Zhao, T. Z.; Kumar, V.; Levine, S.; and Finn, C. 2023. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. In *Proceedings of Robotics: Science and Systems*.

Zitkovich, B.; Yu, T.; Xu, S.; Xu, P.; Xiao, T.; Xia, F.; Wu, J.; Wohlhart, P.; Welker, S.; Wahid, A.; Vuong, Q.; Vanhoucke, V.; Tran, H.; Soricut, R.; Singh, A.; Singh, J.; Sermanet, P.; Sanketi, P.; Salazar, G.; Ryoo, M.; Reymann, K.; Rao, K.; Pertsch, K.; Mordatch, I.; Michalewski, H.; Lu, Y.; Levine, S.; Lee, T.-W. E.; Lee, L.; Leal, I.; Kuang, Y.; Kalashnikov, D.; Julian, R.; Joshi, N.; Irpan, A.; Ichter, B.; Hsu, J.; Herzog, A.; Hausman, K.; Gopalakrishnan, K.; Fu, C.; Florence, P.; Finn, C.; Dubey, A.; Driess, D.; Ding, T.; Choromanski, K.; Chen, X.; Chebotar, Y.; Carbajal, J.; Brown, N.; Brohan, A.; Arenas, M. G.; and Han, K. 2023. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, 2165–2183. PMLR.