RETHINKING DENSE SPATIAL FEATURES: COMPACT SPATIAL QUERIES FOR WORLD ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

World Action Models (WAMs) jointly model future visual observations and robot actions, transferring video priors to robotic manipulation. However, their image-derived representations leave interaction-relevant 3D geometry implicit; dense 3D features can provide complementary spatial cues, but may also introduce manipulation-irrelevant redundancy and additional learning and computational costs. We therefore systematically investigate whether WAMs benefit from additional spatial conditioning, which representations and supervision targets are effective for action generation, and how compact an effective spatial representation can be. To test a compact and manipulation-aligned alternative within this study, we develop **QuestWAM**, which learns latent spatial queries through supervision from future 3D bounding boxes, interaction points, and semantics, organizing spatial information around where task-relevant objects will be and where the robot should interact with them. These queries condition action generation without requiring future annotations or dense 3D reconstruction at inference. QuestWAM achieves 93.14% and 92.18% success on RoboTwin 2.0 Clean and Random, respectively, and improves LIBERO-Plus success from 49.43% for original FastWAM to 53.48%. Compared with online VGGT conditioning at the same token budget on RoboTwin 2.0 Clean, QuestWAM improves success by 1.00 percentage point. Controlled comparisons favor structured supervision over dense-feature distillation and latent queries over decoded attributes. Spatial interventions further show that the queries encode scene-dependent future geometry. These findings suggest that compact and manipulation-aligned spatial conditioning improves WAM action generation, with its value depending on how relevant information is selected and organized, beyond representational richness alone.

1 INTRODUCTION

World Action Models (WAMs) offer a promising path toward general-purpose robot manipulation by jointly modeling visual dynamics and robot actions with rich priors from large-scale video models (Kim et al., 2026; Ye et al., 2026; Yuan et al., 2026). However, accurate scene prediction does not yield representations suited for precise action generation (Han et al., 2026). Physical interaction in 3D requires policies to understand scene semantics and dynamics while identifying task-relevant object locations, spatial extents, and regions for end-effector approach and interaction. Video features may implicitly encode this information, but visual prediction objectives do not necessarily organize it in a form readily usable by an action generator (Zhang et al., 2026b). Translating rich video priors into spatial information useful for physical control therefore remains a key challenge.

Recent work increasingly incorporates explicit geometric and spatiotemporal structure into robot policies (Ze et al., 2024; Huang et al., 2026). Geometric Action Model (Han et al., 2026) builds action prediction upon pretrained geometric representations, while GaussianWAM (Zhang et al., 2026b), WAM4D (Li et al., 2026c), and 4D-WAM (Yang et al., 2026) enhance WAMs through multi-view geometry, spatial registers, or spatiotemporal motion representations. Although these representations preserve rich scene structure, action models must still isolate manipulation-relevant content, while constructing dense geometry can add feature extraction and fusion costs (Wang et al., 2025; Yang et al., 2026). These studies support the value of geometric structure, yet leave a question underexplored: *how rich does a spatial representation need to be for action generation.*

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.

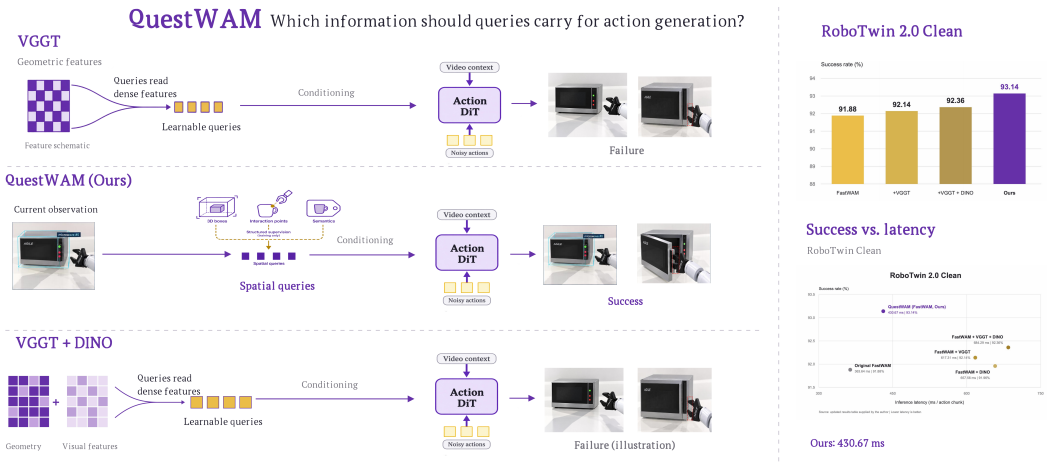


Figure 1: **Motivation and overview of QuestWAM.** QuestWAM learns compact spatial queries through structured 3D supervision as an alternative to dense-feature conditioning. Right: higher success and lower inference latency than the tested dense-conditioning alternatives on RoboTwin 2.0 Clean.

We approach this question by prioritizing manipulation relevance over scene completeness. For object-centric tasks, this suggests organizing future spatial information around task-relevant objects: semantics identifies the relevant entities, future 3D bounding boxes summarize where they will be and the extent they will occupy, and interaction points localize where the robot should engage over the manipulation horizon (Manuelli et al., 2019; Huang et al., 2025). This motivates our central hypothesis: *for action generation in WAMs, the utility of an auxiliary spatial representation depends on selecting and organizing geometry and interaction cues around task-relevant objects rather than maximizing geometric detail.*

To investigate whether compact, manipulation-aligned spatial information can improve action generation, we introduce **QuestWAM**, a spatial-query framework for pretrained WAMs. QuestWAM predicts a small set of latent spatial queries from current multi-view observations, language instructions, and robot states, supervised by future 3D bounding boxes, interaction locations, and semantic states. This design allows us to examine whether organizing spatial information around future object geometry and interaction requirements provides an effective complement to existing video priors through a compact interface. The queries supplement the original video–action pathway without requiring ground-truth future annotations or dense 3D reconstruction at inference. On RoboTwin 2.0 (Chen et al., 2025), QuestWAM achieves **93.14%** and **92.18%** success on Clean and Random, respectively, improving Clean success by 1.26 percentage points over original FastWAM. Under a shared action-fusion interface and matched token budgets, it outperforms the tested dense visual and geometric conditioning alternatives (Siméoni et al., 2025; Wang et al., 2025), improving success by 1.00 percentage point and reducing end-to-end inference latency by 30.2% relative to online VGGT (Wang et al., 2025) conditioning.

We further conduct a systematic representation study examining effective supervision targets, representation construction, action readout, and query capacity. Among the supervision variants, the full objective achieves the highest observed success, with the largest drop when interaction-point supervision is removed. At matched token budgets, structurally supervised queries outperform the tested dense-feature distillation, pooling, and resampling alternatives, while latent queries provide more effective action conditioning than decoded attributes. Capacity comparisons support the compact default configuration, with no further success gains at larger tested budgets. Paired interventions across 50 tasks show that decoded geometry follows donor-object translations and disabling the query branch changes predicted trajectories, demonstrating scene-dependent geometric encoding and action-model sensitivity to query conditioning. Together, these findings support structured spatial supervision and compact latent conditioning as effective complements to existing video priors.

Our contributions are summarized as follows:

- **QuestWAM.** QuestWAM. We introduce compact latent spatial queries supervised by future object geometry, interaction locations, and semantic states to condition WAM action generation without dense 3D reconstruction at inference.
- **Empirical results.** Performance and efficiency. QuestWAM achieves 93.14%/92.18% success on RoboTwin 2.0 Clean/Random and 53.48% on LIBERO-Plus, while outperforming online VGGT conditioning with 30.2% lower inference latency.
- **Systematic representation study.** Systematic representation study. Controlled comparisons reveal the benefits of structured spatial supervision and compact latent conditioning, while paired interventions demonstrate scene-dependent geometric encoding and action sensitivity to spatial queries.

2 Related Work

World Action Models. Video world foundation models learn priors over scene evolution and support embodied visual generation (NVIDIA et al., 2025a;b; Li et al., 2026b). Early video-based policies connect visual dynamics modeling with robot action learning (Wu et al., 2023; Cheang et al., 2024). For world action models (WAMs), one line predicts future visual states alongside actions or conditions actions on imagined futures, bringing scene evolution into the control pathway (Shen et al., 2025; Ye et al., 2026; Li et al., 2026a; Kim et al., 2026). Another line retains video prediction as training supervision while omitting explicit future visual prediction at inference (Yuan et al., 2026). Latent multimodal models support flexible visual and action generation (Bi et al., 2025; Li et al., 2025), while multi-stream designs incorporate geometric and semantic targets and permit action-only inference (Yan et al., 2026). A related direction uses world-model feedback to improve action policies through reinforcement learning (Team et al., 2026; Zhu et al., 2025). These developments motivate examining whether compact, task-aligned representations of future object geometry and interaction regions can effectively condition pretrained WAMs.

Auxiliary Representations. Precise manipulation requires geometry and interaction cues that features learned for recognition or visual prediction may not expose, motivating auxiliary representations for action generation. Beyond WAMs, general vision and geometry models provide dense features and 3D estimates (Siméoni et al., 2025; Wang et al., 2025), although representation utility varies across embodied tasks (Majumdar et al., 2023). Manipulation-specific representations organize spatial information through local descriptors and feature transport, semantic keypoints and relations, or actionability, contact points, and post-contact motion (Florence et al., 2018; Zeng et al., 2020; Manuelli et al., 2019; Huang et al., 2025; Mo et al., 2021; Wu et al., 2025). In WAMs and related predictive models, auxiliary objectives encode scene evolution using future depth, semantics, dynamic regions, or dense 4D correspondences (Zhang et al., 2025; Li et al., 2026c; Zhou et al., 2026). Geometry-aware WAMs further distill multi-view geometry and semantics or align video–action representations with evolving 4D structure and trajectory fields (Zhang et al., 2026b;a; Yang et al., 2026). However, how supervision content, spatial granularity, and conditioning interface affect action generation in pretrained WAMs remains underexplored.

3 Method

3.1 Problem Formulation

We study whether auxiliary spatial conditioning benefits WAM action generation, which representations and supervision targets are effective, and how much representation capacity is needed. Given current multi-view RGB observations o_t, a language instruction ℓ, and robot state s_t, a WAM uses video context H_t^v to predict an action chunk over H future steps. Let C_t denote the auxiliary spatial representation under comparison, giving the conditional policy

$$p_{\theta,\eta}(a_{t:t+H-1} \mid H_t^v, C_t, \ell, s_t)\,, \tag{1}$$

where θ and η are the action-model and spatial-fusion parameters. In QuestWAM (Figure 2), supervision from future 3D boxes, interaction locations, and semantics encourages latent queries to

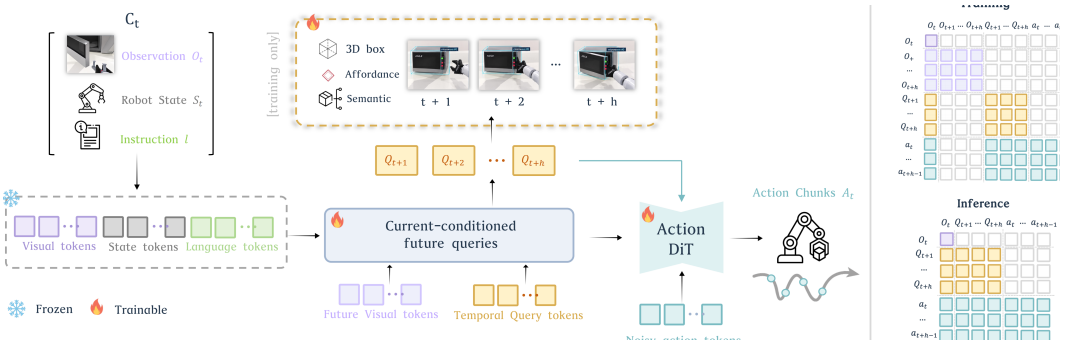


Figure 2: **Overview of QuestWAM.** Current observations, robot states, and instructions condition latent spatial queries supervised by future 3D boxes, interaction points, and semantics. During action adaptation, the frozen query module conditions the Action DiT through gated spatial attention; inference requires neither future annotations nor dense 3D reconstruction. Right: training and inference attention patterns.

encode geometry and interaction information about task-relevant objects. We compare the resulting representations for action generation by varying their sources, supervision targets, readout forms, and token budgets, while sharing the action-fusion architecture across augmented variants.

3.2 Learning Spatial Queries through Structured Supervision

We hypothesize that supervision centered on task-relevant object geometry and interaction locations can produce useful representations for action generation. To examine this hypothesis, we compare queries learned from structured spatial targets with dense-feature conditioning and supervision.

From manipulation requirements to supervision targets. For object-centric manipulation, we select three complementary types of information: future object location and extent, local interaction locations, and the task-relevant objects associated with these attributes. We express these choices through 3D bounding boxes, interaction points, and semantic categories, respectively. Supervising future states within the action horizon encourages the queries to encode spatial information relevant to the upcoming manipulation. This choice lets us test whether supervising these attributes yields better action conditioning than learning to reproduce dense scene features.

Encoding the selected information in latent queries. We use a common query-decoder design to compare supervision targets and examine the effect of representation capacity. Current visual and language features and the projected robot state form a multimodal context M_t, read by learnable object queries $Q = \{q_1, \ldots, q_K\}$. To provide future spatial conditioning over the action horizon from current inputs, a Transformer decoder generates queries autoregressively over future timestamps, conditioned on the context and preceding predictions:

$$\begin{aligned} Z_t^i &= D_\phi(Q, M_t, Z_t^{<i}), \qquad i = 1, \ldots, N_\tau, \\ Z_t &= [Z_t^1; \ldots; Z_t^{N_\tau}]. \end{aligned} \tag{2}$$

Here, $Z_t^i = \{z_j^i\}_{j=1}^K$ contains the tokens for uniformly sampled future timestamp τ_i, and $Z_t^{<i}$ contains preceding predictions, empty at the first step. The groups are concatenated along the token dimension, and all predicted tokens are used for action conditioning.

Learning from structured targets. To apply the selected supervision, an auxiliary head H_ψ decodes each z_j^i into a 3D-box center and size, an interaction-point location, and a semantic-category prediction. Queries are matched to annotated objects independently at each future timestamp, giving

the complete representation-training objective

$$\mathcal{L}_{\mathrm{sp}}^{i,\pi} = \mathcal{L}_{\mathrm{box}}^{i,\pi} + \mathcal{L}_{\mathrm{point}}^{i,\pi} + \lambda_{\mathrm{sem}}\mathcal{L}_{\mathrm{sem}}^{i,\pi},$$
$$\mathcal{L}_{\mathrm{repr}} = \mathbb{E}\left[\omega_r \sum_{i=1}^{N_\tau} \min_{\pi_i} \mathcal{L}_{\mathrm{sp}}^{i,\pi_i}\right]. \tag{3}$$

Here, π_i is the assignment at timestamp τ_i, λ_{sem} balances semantic and geometric supervision, and ω_r balances tasks according to the sampling frequency of task r. Independent matching does not require a query slot to preserve object identity across timestamps. Axis-aligned box centers and interaction points use Smooth-L1 loss on normalized coordinates, box sizes are regressed in normalized logarithmic space, and semantic supervision uses cross-entropy with a no-object category. Only available supervision terms are used when interaction-point annotations are missing. This objective specifies the spatial supervision used to shape the queries. These targets guide representation learning without restricting the information encoded by the latent queries to the supervised attributes.

3.3 Comparing Spatial Representations through a Shared Interface

To compare the usefulness of different spatial representations for action generation, we use a common action readout. We therefore retain the WAM's existing video context and supply additional conditioning through the same fusion architecture for all augmented variants.

Making the representation available to actions. In QuestWAM, the action condition is the predicted latent queries, $C_t = Z_t$, with auxiliary geometric heads used only for training and analysis. Structured targets thus guide representation learning without restricting the action readout to their decoded attributes, a choice tested by comparing latent and explicit conditioning. To provide a separately controllable spatial readout, we map C_t to keys and values at each action layer and fuse them through gated residual attention:

$$(K_C^{(m)}, V_C^{(m)}) = A_\eta^{(m)}(C_t),$$
$$O^{(m)} = O_{\mathrm{base}}^{(m)} + \alpha_m(h^{(m)}) \odot \mathrm{Attn}\left(Q_a^{(m)}, K_C^{(m)}, V_C^{(m)}\right). \tag{4}$$

Here, $O_{\mathrm{base}}^{(m)}$ is the original video–action attention output, $A_\eta^{(m)}$ is a spatial adapter, $Q_a^{(m)}$ denotes action queries, and the gate α_m depends on action hidden state $h^{(m)}$. Spatial attention is normalized separately to avoid competition with video and action tokens in a shared softmax. The separate pathway also provides a defined intervention point for replacing or disabling spatial conditioning while holding the video condition fixed.

Evaluating a learned representation through action adaptation. To compare queries learned under different supervision targets, we first learn the spatial representation and then hold it fixed while training the action model. Stage I freezes the pretrained encoders and trains the query decoder, input projections, learnable queries, and auxiliary heads with $\mathcal{L}_{\mathrm{repr}}$. Stage II freezes the query module and video branch, and trains the Action DiT and spatial adapters on predicted queries. For expert action chunk a, Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$, and noise level σ, let $a_\sigma = (1-\sigma)a + \sigma\epsilon$; action training retains the base WAM objective:

$$\hat{v} = v_{\theta,\eta}(a_\sigma, \sigma \mid H_t^v, Z_t, \ell, s_t),$$
$$\mathcal{L}_{\mathrm{act}} = \mathbb{E}\left[\omega_r w(\sigma) \left\|\hat{v} - (\epsilon - a)\right\|_{\mathrm{valid}}^2\right]. \tag{5}$$

Here, $w(\sigma)$ is the base WAM noise weighting, and errors are averaged over valid action steps and dimensions. Freezing the query module in Stage II prevents the action loss from updating its parameters while the action model learns to use the predicted queries. Action training and inference both use queries predicted from current inputs; inference requires neither future annotations nor additional dense 3D reconstruction.

3.4 Systematic Representation Study

We study spatial representations through the shared action interface along four dimensions: conditioning benefit, supervision and readout, capacity, and action responses.

Benefit and representation sources. We compare QuestWAM with original FastWAM and conditioning derived from VGGT, DINOv3, and their combination. Spatially augmented variants use matched action-conditioning token budgets, with dense features compressed by pooling or learned resampling; task success and end-to-end latency evaluate these choices.

Supervision and action readout. We ablate 3D-box, interaction-point, and semantic supervision individually and compare structured targets with VGGT-feature distillation under the same query architecture. To examine latent conditioning, we compare latent queries with their predicted boxes, interaction points, and optional semantic probabilities. These comparisons examine how supervision and readout choices affect task performance.

Capacity of spatial conditioning. We vary the per-timestamp token count $K \in \{4, 8, 16\}$ while fixing the number of future timestamps and the supervision setting. This tests whether larger conditioning budgets improve performance within the evaluated range.

From encoded geometry to action use. We use paired query interventions to examine the relationship between geometric encoding and action responses. Keeping the policy, source scene, RGB, robot state, instruction, and sampling noise fixed, we substitute donor queries from observations with object translations or illumination changes, or disable the query branch. Decoded-center direction agreement assesses whether query geometry follows donor translations; trajectory RMSE measures changes from the paired Normal baseline; and action direction agreement tests whether end-effector responses align with donor translation. We analyze the first 24 steps across 50 RoboTwin tasks with 20 source scenes per task and average task-level metrics equally. Response slopes with within-task confidence intervals, along with restoration and permutation controls, are examined for opening a microwave and hanging a mug. Evaluation settings, metric definitions, and results are given in Section 4 and Appendix C.

4 Experiments

We first evaluate the benefit of additional spatial conditioning, then examine which supervision signals and action-readout choices yield effective representations, and use query interventions to study the relationship between geometric encoding and action responses.

4.1 Experimental Setup

We evaluate manipulation success on RoboTwin 2.0 Clean and Random (Chen et al., 2025), and zero-shot generalization under seven types of distribution shift in LIBERO-Plus (Fei et al., 2026) for policies trained on LIBERO (Liu et al., 2023). Comparisons include original FastWAM (Yuan et al., 2026), conditioning from VGGT (Wang et al., 2025), DINOv3 (Siméoni et al., 2025), their combination, and QuestWAM. Success rates are reported in % and differences in percentage points (pp).

Spatially augmented variants share the action-fusion interface, with matched total conditioning-token budgets in the main comparison. Here, K denotes tokens per future query group; four future timestamps and the default $K = 4$ give 16 conditioning tokens. Representation comparisons share data splits and use RoboTwin 2.0 Clean at the default budget, except for the capacity sweep. Architecture, two-stage training, and evaluation details are provided in Appendix A.

4.2 Benefit of Additional Spatial Conditioning

To test whether additional spatial conditioning complements existing video priors, we compare QuestWAM with the original model and alternative conditioning schemes. QuestWAM achieves 93.14% and 92.18% on Clean and Random, improving over original FastWAM by 1.26 and 1.30 pp, respectively (Table 1). It also exceeds the strongest dense-conditioning alternative, VGGT+DINOv3, by 0.78 and 0.42 pp. On Clean, QuestWAM takes 430.67 ms per action chunk, a 30.2% reduction from 617.31 ms with online VGGT conditioning. These results support compact spatial conditioning as a complement to existing video representations; the following comparisons examine effective supervision and readout choices.

Table 1: **Main results on RoboTwin 2.0.** SR is reported in %, and end-to-end latency in ms per action chunk, with matched token budgets across spatially augmented methods. **Bold**/underline: best/second best. Resampler maps dense features to a fixed number of conditioning tokens through a learned module.

	Clean		Random	
Method	**SR ↑**	**Latency ↓**	**SR ↑**	**Latency ↓**
Original FastWAM	91.88	**363.64**	90.88	**364.24**
FastWAM + VGGT (resampler)	92.14	617.31	91.54	676.65
FastWAM + DINOv3	91.96	657.56	91.22	660.63
FastWAM + VGGT + DINOv3 (resampler)	92.36	684.29	91.76	697.36
QuestWAM (ours)	**93.14**	430.67	**92.18**	432.79

Table 2: Success rates (%) on LIBERO-Plus for original FastWAM and QuestWAM. The benchmark evaluates zero-shot generalization under seven types of variation. Bold values indicate the better result in each column.

Method	**Camera**	**Robot**	**Lang.**	**Light**	**BG**	**Noise**	**Layout**	**Overall**
Original FastWAM	15.88	43.81	**67.27**	78.63	51.58	38.79	60.13	49.43
QuestWAM (ours)	**21.70**	**45.48**	67.14	**85.81**	**54.74**	**47.91**	**61.90**	**53.48**

On LIBERO-Plus, overall success increases from 49.43% to 53.48%, a gain of 4.05 pp, with improvements in six of seven shift categories (Table 2). These results extend the evidence for useful spatial conditioning to the tested distribution shifts.

Figure 3 shows real-world deployments of QuestWAM across object-centric manipulation tasks. We compare FastWAM and QuestWAM on four real-world tasks—tube insertion, slipper placement, color sorting, and pen placement—with 20 evaluation trials per task for each method. FastWAM achieves success rates of 10%, 25%, 50%, and 35%, while QuestWAM reaches 20%, 30%, 70%, and 65%, respectively. The equal-weight average success rate increases from 30.00% to 46.25%, a gain of 16.25 percentage points, with improvements on all four tasks. Tube insertion and pen placement involve alignment with receptacles, while slipper placement and block sorting involve object transport to language-specified destinations. The sequences illustrate progression from approach and grasping to placement or insertion, showing the object and interaction requirements targeted by the spatial supervision.

4.3 Learning Effective Spatial Conditions

To study which supervision signals help learn action conditions, we remove 3D-box, interaction-point, or semantic supervision individually under the same query architecture while retaining the other terms. The full configuration achieves 93.14%, compared with 92.96%, 92.52%, and 92.98% after removing the respective supervision terms (Table 3(a)). Removing interaction-point supervision yields the largest observed drop of 0.62 pp, and joint supervision gives the highest success rate among these configurations.

To examine structured targets relative to dense-feature supervision, we compare structured supervision with VGGT-feature distillation under the same query architecture and interface budget. Structured supervision exceeds distillation at 91.98% by 1.16 pp and fixed VGGT pooling at 92.04% (Table 3(b)). Together with the resampling comparisons in Table 1, these results support learning compact action conditions from structured targets aligned with manipulation.

To examine how the representation should condition actions, we compare latent queries with their decoded attributes. Latent queries outperform predicted boxes and interaction points by 1.12 pp, and the variant adding semantic probabilities by 1.14 pp (Table 3(c)). This supports using structured

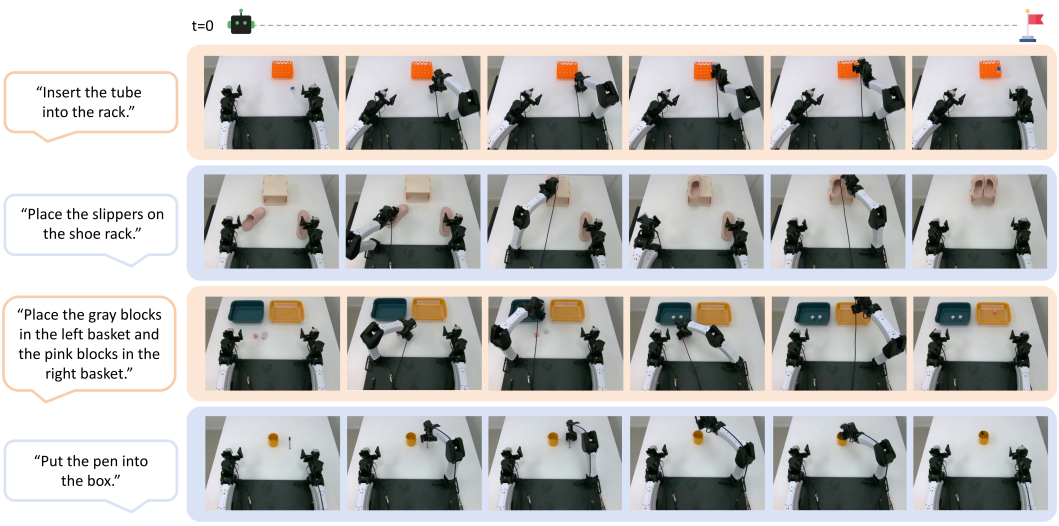


Figure 3: Real-world deployment sequences of QuestWAM for tube insertion, slipper placement, color-based block sorting, and pen placement. Each row follows the instruction on the left, with time progressing from left to right.

attributes as supervision targets while retaining latent queries for action conditioning. Capacity sweeps and full latency comparisons are provided in Appendix B.

Table 3: Supervision components, representation construction, and action readout on RoboTwin 2.0 Clean at the default matched budget. (a) Checkmarks indicate active supervision; ΔSR is the decrease from full QuestWAM in pp. (b) Distillation uses the same query architecture; pooling compresses dense features into conditioning tokens through fixed aggregation. (c) Explicit attributes are predicted by the query module; the highest success rate in each panel is bold.

(a) Supervision components

Training setting	**3D Box**	**Point**	**Semantic**	**SR (%) ↑**	**ΔSR (pp)**
QuestWAM (ours)	✓	✓	✓	**93.14**	0.00
w/o 3D-box supervision	–	✓	✓	92.96	0.18
w/o interaction-point supervision	✓	–	✓	92.52	0.62
w/o semantic supervision	✓	✓	–	92.98	0.16

(b) Supervision / construction

Setting	**SR (%)**
QuestWAM (ours)	**93.14**
VGGT feature distillation	91.98
VGGT (pooling)	92.04

(c) Action readout

Action condition	**SR (%)**
Latent queries	**93.14**
Boxes + points	92.02
+ semantic probabilities	92.00

4.4 Geometric Encoding and Action Responses

To study the geometry encoded in the queries and how actions respond to query changes, we conduct paired interventions across all 50 RoboTwin tasks, using 20 source scenes per task. Using the frozen epoch-2 checkpoint and a gate scale of 0.1, we hold the original inputs fixed, substitute geometry- or illumination-donor queries, or disable the query branch. We analyze predicted actions and end-effector trajectories over the first 24 steps and report macro-averages that give equal weight to each of the 50 tasks.

Table 4: Paired query interventions across all 50 RoboTwin tasks, using 20 source scenes per task (1,000 in total) and analyzing the first 24 steps. All results are equal-weight macro-averages of task-level metrics at the default query budget; trajectory RMSEs are in cm relative to the paired Normal baseline. Evaluation and aggregation details are in Appendix C.

Measurement	**Macro-average (50 tasks)**
Decoded-center direction agreement (%)	100
Branch-off trajectory RMSE (cm)	0.25405
Geometry-donor mean trajectory RMSE (cm)	0.06940
Illumination-donor trajectory RMSE (cm)	0.06735
Action direction agreement: x / y (%)	15.85 / 10.85

The macro-averaged decoded-center direction agreement is 100%, and branch removal yields a macro-averaged trajectory RMSE of 0.25405 cm (Table 4). These results support scene-dependent geometric encoding and an influence of the query branch on action predictions. Geometry- and illumination-donor trajectory RMSEs average 0.06940 and 0.06735 cm, respectively, while action direction agreement averages 15.85% on x and 10.85% on y. These aggregate responses do not establish consistent geometry-guided action changes in the tested initial segments. Evaluation protocols, metric definitions, and task-aggregation details are provided in Appendix C.

5 Conclusion

We investigated how compact spatial conditioning improves WAM action generation through controlled comparisons of representation sources, supervision targets, action readouts, and query capacity. We introduced QuestWAM, which learns latent spatial queries supervised by future 3D boxes, interaction points, and semantics, without requiring future annotations or dense 3D reconstruction at inference. QuestWAM achieves 93.14% and 92.18% success on RoboTwin 2.0 Clean and Random, respectively, and improves LIBERO-Plus success to 53.48%; real-world deployments further illustrate its application to object manipulation. At matched token budgets, structurally supervised queries outperform the tested dense-conditioning and feature-distillation alternatives, while latent queries provide more effective action conditioning than decoded attributes. Supervision ablations favor the joint objective, and capacity comparisons support the compact default configuration. Paired interventions further reveal scene-dependent geometry in the queries and the influence of the query branch on action predictions. Together, these results support learning compact latent representations through structured supervision aligned with manipulation requirements as an effective spatial complement to existing video priors.

AI use statement

In this work, we used generative AI tools to assist with translation, refinement of methodological and experimental descriptions, and the preparation and revision of some scientific figures. These tools were not used to generate the training or evaluation data used in this work, produce experimental results, or independently determine the research conclusions. The authors manually reviewed and revised all AI-assisted text, technical descriptions and figures, and verified them against the system implementation, experimental records, and original sources. The authors take full responsibility for the final content of the paper and all material produced with AI assistance.

Reproducibility Statement

We provide the model architecture, training objectives, data composition, and evaluation protocols in the main paper and appendix.

References

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang

Su, and Jun Zhu. Motus: A unified latent action world model. *arXiv preprint arXiv:2512.13030*, 2025. URL https://arxiv.org/abs/2512.13030.

Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, Hanbo Zhang, and Minzhao Zhu. GR-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024. URL https://arxiv.org/abs/2410.06158.

Tianxing Chen, Zanxin Chen, Baijun Chen, et al. RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025. URL https://arxiv.org/abs/2506.18088.

Senyu Fei, Siyin Wang, Junhao Shi, et al. LIBERO-Plus: A progressive robustness benchmark for visual-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 38574–38583, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Fei_LIBERO-Plus_A_Progressive_Robustness_Benchmark_for_Visual-Language-Action_Models_CVPR_2026_paper.html.

Peter R. Florence, Lucas Manuelli, and Russ Tedrake. Dense object nets: Learning dense visual object descriptors by and for robotic manipulation. *arXiv preprint arXiv:1806.08756*, 2018. URL https://arxiv.org/abs/1806.08756.

Jisang Han, Seonghu Jeon, Jaewoo Jung, et al. Geometric action model for robot policy learning. *arXiv preprint arXiv:2606.17046*, 2026. URL https://arxiv.org/abs/2606.17046.

Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. ReKep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 4573–4602. PMLR, 2025. URL https://proceedings.mlr.press/v270/huang25g.html.

Wenlong Huang, Yu-Wei Chao, Arsalan Mousavian, et al. PointWorld: Scaling 3D world models for in-the-wild robotic manipulation. *arXiv preprint arXiv:2601.03782*, 2026. URL https://arxiv.org/abs/2601.03782.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026. URL https://arxiv.org/abs/2601.16163.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026a. URL https://arxiv.org/abs/2601.21998.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified Video Action Model. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.074.

Xinghang Li, Jun Guo, Qiwei Li, Long Qian, Hang Lai, Yueze Wang, Hongyu Yan, Jiahang Cao, Xi Chen, Jingen Qu, Jiaxi Song, Nan Sun, Hanye Zhao, Futeng Liu, Wanli Peng, Heyun Wang, Yunhong Wang, Caoyu Xia, Jack Zhao, Diyun Xiang, Hangjun Ye, Heng Qu, Huaping Liu, and Jason Li. Xiaomi-Robotics-U0: Unified embodied synthesis with world foundation model. *arXiv preprint arXiv:2607.11643*, 2026b. URL https://arxiv.org/abs/2607.11643.

Ying Li, Xiaobao Wei, Jiajun Cao, Hao Wang, Xiaowei Chi, Chengyu Bai, Qianpu Sun, Jiajun Li, Xiaojie Zhang, Peidong Jia, Jian Tang, Sirui Han, and Shanghang Zhang. WAM4D: Fast 4D world action model via spatial register tokens. *arXiv preprint arXiv:2606.14048*, 2026c. URL https://arxiv.org/abs/2606.14048.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems*, 2023. URL

https://proceedings.neurips.cc/paper_files/paper/2023/hash/8c3c666820ea055a77726d66fc7d447f-Abstract-Datasets_and_Benchmarks.html.

Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Yecheng Jason Ma, Claire Chen, Sneha Silwal, Aryan Jain, Vincent-Pierre Berges, Pieter Abbeel, Jitendra Malik, Dhruv Batra, Yixin Lin, Oleksandr Maksymets, Aravind Rajeswaran, and Franziska Meier. Where are we in the search for an artificial visual cortex for embodied intelligence? *arXiv preprint arXiv:2303.18240*, 2023. URL https://arxiv.org/abs/2303.18240.

Lucas Manuelli, Wei Gao, Peter Florence, and Russ Tedrake. kPAM: Keypoint affordances for category-level robotic manipulation. *arXiv preprint arXiv:1903.06684*, 2019. URL https://arxiv.org/abs/1903.06684.

Kaichun Mo, Leonidas J. Guibas, Mustafa Mukadam, Abhinav Gupta, and Shubham Tulsiani. Where2Act: From pixels to actions for articulated 3D objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6813–6823, 2021. URL https://openaccess.thecvf.com/content/ICCV2021/html/Mo_Where2Act_From_Pixels_to_Actions_for_Articulated_3D_Objects_ICCV_2021_paper.html.

NVIDIA et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025a. URL https://arxiv.org/abs/2501.03575.

NVIDIA et al. World simulation with video foundation models for physical ai. *arXiv preprint arXiv:2511.00062*, 2025b. URL https://arxiv.org/abs/2511.00062.

Yichao Shen, Fangyun Wei, Zhiying Du, Yaobo Liang, Yan Lu, Jiaolong Yang, Nanning Zheng, and Baining Guo. VideoVLA: Video generators can be generalizable robot manipulators. *arXiv preprint arXiv:2512.06963*, 2025. URL https://arxiv.org/abs/2512.06963.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025. URL https://arxiv.org/abs/2508.10104.

GigaBrain Team, Boyuan Wang, Bohan Li, Chaojun Ni, Guan Huang, Guosheng Zhao, Hao Li, Jie Li, Jindi Lv, Jingyu Liu, Lv Feng, Mingming Yu, Peng Li, Qiuping Deng, Tianze Liu, Xinyu Zhou, Xinze Chen, Xiaofeng Wang, Yang Wang, Yifan Li, Yifei Nie, Yilong Li, Yukun Zhou, Yun Ye, Zhichao Liu, and Zheng Zhu. Gigabrain-0.5m*: a vla that learns from world model-based reinforcement learning, 2026. URL https://arxiv.org/abs/2602.12099.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wang_VGGT_Visual_Geometry_Grounded_Transformer_CVPR_2025_paper.html.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. *arXiv preprint arXiv:2312.13139*, 2023. URL https://arxiv.org/abs/2312.13139.

Shijie Wu, Yihang Zhu, Yunao Huang, Kaizhen Zhu, Jiayuan Gu, Jingyi Yu, Ye Shi, and Jingya Wang. AffordDP: Generalizable diffusion policy with transferable affordance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6971–6980, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wu_AffordDP_Generalizable_Diffusion_Policy_with_Transferable_Affordance_CVPR_2025_paper.html.

Ge Yan, Jinghao Liu, Yuzhi Fan, Lei Cai, Minwen Liao, Jesse Zhang, and Dieter Fox. Flex-π: A multi-stream world-action model with compute flexibility. *arXiv preprint arXiv:2608.10860*, 2026. URL https://arxiv.org/abs/2608.10860.

Lishan Yang, Wenxuan Song, Xi Wang, Pingyue Sheng, Zheng Fang, Ziyang Zhou, Junjie He, Haodong Yan, Jiayi Chen, Nan Sun, Qiao Sun, Pengwei Wang, Lingqiao Liu, Yan Wang, Yuxiang Gao, Feras Dayoub, and Haoang Li. 4D-WAM: Infusing spatiotemporal awareness into world action models through trajectory fields. *arXiv preprint arXiv:2608.08023*, 2026. URL https://arxiv.org/abs/2608.08023.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026. URL https://arxiv.org/abs/2602.15922.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-WAM: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026. URL https://arxiv.org/abs/2603.16666.

Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3D Diffusion Policy: Generalizable visuomotor policy learning via simple 3D representations. In *Proceedings of Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.067. URL https://www.roboticsproceedings.org/rss20/p067.html.

Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Ayzaan Wahid, Vikas Sindhwani, and Johnny Lee. Transporter networks: Rearranging the visual world for robotic manipulation. *arXiv preprint arXiv:2010.14406*, 2020. URL https://arxiv.org/abs/2010.14406.

Jianjun Zhang, Jian Zhu, Taiyi Su, Chong Ma, Zitai Huang, Yi Xu, and Hanli Wang. Learning 4D geometric priors for inference-efficient world action models. *arXiv preprint arXiv:2607.05468*, 2026a. URL https://arxiv.org/abs/2607.05468.

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, Fan Lu, He Wang, Zhizheng Zhang, Li Yi, Wenjun Zeng, and Xin Jin. DreamVLA: A vision-language-action model dreamed with comprehensive world knowledge. *arXiv preprint arXiv:2507.04447*, 2025. URL https://arxiv.org/abs/2507.04447.

Zijian Zhang, Yuqing Jiang, Weitao Zhou, Minglei Li, Jinhao Zhang, Yao Mu, Xiaofan Li, Hao Zhao, and Haibao Yu. GaussianWAM: Distilling geometry and semantics from 3D gaussian fields into world-action models. *arXiv preprint arXiv:2608.24714*, 2026b. URL https://arxiv.org/abs/2608.24714.

Kaichen Zhou, Yuzhen Chen, Fangneng Zhan, Hang Hua, Grace Chen, Xinhai Chang, Ao Qu, Yilun Du, Zhuang Liu, Paul Pu Liang, and Mengyu Wang. GEM-4D: Geometry-enhanced video world models for robot manipulation. *arXiv preprint arXiv:2605.22882*, 2026. URL https://arxiv.org/abs/2605.22882.

Fangqi Zhu, Zhengyang Yan, Zicong Hong, Quanxin Shou, Xiao Ma, and Song Guo. Wmpo: World model-based policy optimization for vision-language-action models, 2025. URL https://arxiv.org/abs/2511.09515.

A Implementation and Evaluation Details

Architecture and optimization

We retain the video expert, action expert, and MoT interaction pathway; RoboTwin experiments initialize from the public three-camera checkpoint. The query module is a four-layer Transformer decoder with hidden dimension 512 and eight attention heads; it predicts autoregressively over future timestamps and supplies all outputs to the 30 action layers through residual spatial attention.

For QuestWAM, Stage I learns spatial queries with frozen encoders; Stage II freezes the query module and video branch while training the action model and fusion modules with the original

action objective. Training uses eight H20 96GB GPUs, BF16, AdamW, and an effective batch size of 128, with linear warm-up for the first 5% of updates followed by cosine decay. Peak learning rates are 2×10^{-4} for query training and 5×10^{-5} for fusion training; action adaptation follows the backbone setting. Controlled representation comparisons share data splits, sampling rules, and batch size; spatial query pretraining remains an additional stage.

Inference predicts 32-step action chunks, executes the first 24 steps before replanning, and uses 10 denoising steps. Latency is measured on one H20 with batch size 1 and BF16; separate architecture profiles and their distinction from policy evaluation are documented in Appendix D.

Conditioning and evaluation

Original FastWAM uses its original action-training objective without the added spatial-query branch or auxiliary spatial supervision. Online VGGT, DINOv3, and combined features are compressed by pooling or learned resampling to the action-interface budget used for comparison. Here, dense describes the feature source or teacher target, rather than more conditioning tokens supplied to the action model. The VGGT-distillation control uses the same query architecture as QuestWAM and does not use the teacher at inference.

LIBERO-Plus covers changes in camera viewpoint, robot appearance, language, lighting, background, visual noise, and layout. Success-rate differences are expressed in percentage points; point-estimate differences alone are not used to infer statistical significance. Per-task Clean results and their evaluation provenance are provided in Appendix E. The real-world deployment sequences in Figure 3 illustrate progression from approach to object manipulation across tasks.

B Extended Representation Comparisons

Supervision and representation construction

The supervision ablations in Table 3(a) remove 3D-box, interaction-point, or semantic supervision individually while retaining the other terms. Removing a supervision term changes the training objective without ensuring that the corresponding information is absent from the latent queries.

The construction comparison includes online feature resampling, pooling, and training-only teacher distillation (Table 5). VGGT distillation reaches 91.98% versus 93.14% for structured supervision; this comparison examines supervision targets under the same query architecture. Representation-construction latency is measured separately for each comparison.

Table 5: Full representation-construction comparison on RoboTwin 2.0 Clean at the default $K = 4$, including latency from the corresponding studies. QuestWAM and the distillation control use the same query architecture; online pooling and resampling share the action-fusion interface at the matched budget. Pooling uses fixed aggregation and a resampler uses a learned module to convert dense features into a fixed number of conditioning tokens. Latency is ms per action chunk.

Representation / construction	K	**SR (%)** ↑	**Latency** ↓
Structured sparse queries (QuestWAM)	4	**93.14**	560.56
Dense VGGT + resampler	4	92.14	597.13
Dense VGGT+DINOv3 + resampler	4	92.36	698.69
Dense VGGT + pooling	4	92.04	667.58
Queries distilled from dense VGGT	4	91.98	572.07

Dense refers to the feature source or teacher target; K follows the definition in the setup. The VGGT teacher is absent at inference for the distillation row. QuestWAM latency is 560.56 ms in the distillation comparison and 553.96 ms in the pooling/resampling comparison; each serves as the reference for its respective study.

Representation capacity

Table 6 compares success rates and associated latencies across conditioning-token budgets. K denotes tokens per future query group, with the future timestamp count fixed; each method's supervi-

sion and training budget are held fixed across the sweep. None of the three representations achieves higher success at the larger tested budgets; smaller budgets were not tested.

Table 6: Representation capacity on RoboTwin 2.0 Clean; K denotes tokens per future query group, with the future timestamp count fixed. Resampler denotes a learned module that converts dense features into a fixed number of conditioning tokens. Original FastWAM has no additional spatial-query branch; bold marks the highest success rate and lowest latency.

Method	K	**SR (%)** ↑	**Latency (ms)** ↓
Original FastWAM	–	91.88	**363.64**
FastWAM + VGGT (resampler)	4	92.14	617.31
	8	91.98	659.38
	16	91.92	651.36
FastWAM + VGGT + DINOv3 (resampler)	4	92.36	684.29
	8	92.26	701.21
	16	92.18	680.07
QuestWAM (ours)	4	**93.14**	430.67
	8	92.56	588.15
	16	92.56	585.61

Latent and explicit action conditions

Explicit conditions comprise predicted boxes, interaction points, and optional semantic probabilities from the query module, without using future ground truth as action input. Table 7 reports success rates and latency for this action-readout comparison. The advantage of latent queries may involve additional information, decoding errors, or action-readout difficulty; this comparison does not isolate these factors.

Table 7: Action-conditioning representations on RoboTwin 2.0 Clean at the default matched budget. Explicit attributes are predicted by the query module rather than supplied as ground truth. Latencies are from this representation comparison, in ms per action chunk.

Action condition	K	**SR (%)** ↑	**Latency (ms)** ↓
Predicted boxes + points	4	92.02	592.41
+ semantic probabilities	4	92.00	559.50
Latent queries (QuestWAM)	4	**93.14**	553.96

C Paired Query Interventions

Evaluation scope and aggregation

We conduct paired query interventions across all 50 RoboTwin tasks, using 20 fixed source scenes per task for a total of 1,000 source scenes. All interventions use the frozen epoch-2 checkpoint, the default $K = 4$ configuration, and a gate scale of 0.1, analyzing actions and end-effector trajectories over the first 24 steps. Source RGB observations, robot states, language instructions, and sampling noise remain fixed while we substitute geometry- or illumination-donor queries or disable the query branch.

Table 4 reports equal-weight macro-averages over the 50 tasks: each metric is computed within each task and then averaged across tasks. Trajectory RMSEs are computed relative to the paired Normal baseline, and the aggregate is the mean of task-level RMSEs. The analyzed arm is selected by motion in the Normal trajectory; action direction agreement requires end-effector displacement greater than 10^{-5} m and an angle of at most 60° to donor-object translation.

TASK-LEVEL SLOPE ANALYSIS AND CONTROLS

Alongside the 50-task aggregate, Table 8 reports axis-specific response slopes and within-task confidence intervals for opening a microwave and hanging a mug. Each of these two tasks uses 20 source scenes with 12 translated-object donors per scene, giving 240 geometry interventions clustered within 20 scenes per task. Restore and permutation controls exactly reproduce the original actions and end-effector trajectories in all 40 source scenes of these two tasks.

We estimate slopes of same-axis end-effector displacement against donor-object translation using 10,000 source-scene-clustered bootstrap resamples within each task. Each resample retains all translation magnitudes within the sampled scenes; nominal 95% confidence intervals are not corrected for the four task–axis comparisons. These intervals characterize the corresponding task-specific slopes, not uncertainty in the 50-task macro-average. The microwave response is near zero on x and weakly negative on y, while both mug axes have small positive slopes, showing differing directional responses in these task-level analyses.

The intervention analyses concern the first 24 steps and do not measure full-task success; source RGB may conflict with donor queries, and complete-query replacement can transfer non-geometric changes.

Table 8: Task-specific slopes of same-axis end-effector displacement against donor-object translation for opening a microwave and hanging a mug, in cm/cm. Each task uses 20 source scenes; nominal 95% confidence intervals are estimated by source-scene-clustered bootstrap within each task. The equal-weight 50-task aggregate is reported in Table 4.

Task	**Axis**	**Slope (cm/cm)**	**Nominal 95% CI**
Open microwave	x	-0.000000706	$[-0.00110247,\ 0.00106724]$
Open microwave	y	-0.00123665	$[-0.00233496,\ -0.00006262]$
Hanging mug	x	0.000692534	$[0.00006866,\ 0.00133468]$
Hanging mug	y	0.00151799	$[0.00016860,\ 0.00301455]$

The 240 geometry interventions per task are clustered within 20 source scenes; confidence intervals use 10,000 bootstrap resamples without correction for multiple comparisons.

D RUNTIME MEASUREMENTS

Table 1 reports end-to-end latency per action chunk for the evaluated policies, using a common timing statistic and including online feature extraction. QuestWAM takes 430.67 and 432.79 ms on Clean and Random, respectively, compared with 363.64 and 364.24 ms for original FastWAM.

The separate architecture profiles use B for total tokens, distinct from the per-future-timestamp K in the policy-capacity sweep. The combined-feature variant allocates $B/2$ tokens to each branch. Only VGGT at $B = 4$ uses correspondingly trained weights; the other configurations were not retrained, so these results characterize architecture runtime only. This experiment characterizes architecture runtime, with policy success evaluated in the manipulation experiments.

Table 9: Architecture profiles for dense-feature conditioning, including online feature extraction; a resampler learns to convert features into a fixed number of conditioning tokens. B follows the total-token count in the profiling report and is distinct from the per-timestamp K used in the policy-capacity study. Measurements use one H20, batch size 1, BF16, 10 denoising steps, and 32-step action chunks, with 100 warm-up iterations and 1,000 timed iterations per setting. Only VGGT at $B = 4$ uses correspondingly trained weights; the other settings are architecture profiles without retraining, and these timings are not paired with main-table success rates.

Conditioning	B	**Mean $\pm$ SD (ms)**	**P95 (ms)**
VGGT + resampler	4	614.84 ± 2.73	617.31
VGGT + resampler	8	656.29 ± 2.25	659.38
VGGT + resampler	16	648.17 ± 3.48	651.36
VGGT+DINOv3 + resampler	4	681.43 ± 2.80	684.29
VGGT+DINOv3 + resampler	8	696.41 ± 3.32	701.21
VGGT+DINOv3 + resampler	16	677.54 ± 2.96	680.07

The combined variant allocates $B/2$ tokens to each feature branch. Standard deviations describe repeated inference timings, not uncertainty in task success or training randomness.

E PER-TASK ROBOTWIN RESULTS

Table 10: Per-task success rates (%) on RoboTwin 2.0 Clean. Results for $\pi_{0.5}$, Motus, and FastWAM are taken from the per-task evaluation reported by FastWAM (Yuan et al., 2026). QuestWAM uses a development-selected checkpoint evaluated on 100 independent episodes per task. The highest score in each row is highlighted in bold, including ties.

Task	H	$\pi_{0.5}$	Motus	FastWAM	QuestWAM
Adjust Bottle	1	**100**	89	**100**	**100**
Beat Block Hammer	1	96	95	**99**	96
Blocks Ranking RGB	3	92	99	**100**	**100**
Blocks Ranking Size	3	49	75	**94**	93
Click Alarmclock	1	98	**100**	**100**	99
Click Bell	1	99	**100**	**100**	**100**
Dump Bin Bigbin	1	92	95	**97**	96
Grab Roller	1	**100**	**100**	**100**	**100**
Handover Block	2	66	86	**95**	**95**
Handover Mic	2	98	78	**99**	**99**
Hanging Mug	2	18	38	58	**81**
Lift Pot	1	96	96	**100**	**100**
Move Can Pot	1	51	34	90	**96**
Move Pillbottle Pad	1	84	93	**100**	**100**
Move Playingcard Away	1	96	**100**	**100**	99
Move Stapler Pad	1	56	83	77	**84**
Open Laptop	1	90	95	**98**	96
Open Microwave	1	34	**95**	62	68
Pick Diverse Bottles	2	81	90	80	**91**
Pick Dual Bottles	2	93	96	**100**	**100**
Place A2B Left	1	87	88	**95**	**95**
Place A2B Right	1	87	91	93	**94**
Place Bread Basket	1	77	91	91	**93**
Place Bread Skillet	2	85	86	90	**96**
Place Burger Fries	2	94	**98**	96	95
Place Can Basket	2	62	**81**	71	72
Place Cans Plasticbox	2	94	98	**99**	**99**
Place Container Plate	1	99	98	96	**100**
Place Dual Shoes	2	75	93	**94**	91
Place Empty Cup	1	**100**	99	**100**	**100**
Place Fan	1	87	91	**96**	95
Place Mouse Pad	1	60	66	83	**86**
Place Object Basket	2	80	81	**89**	86
Place Object Scale	1	86	88	90	**93**
Place Object Stand	1	91	**98**	90	**98**
Place Phone Stand	1	81	87	**97**	96
Place Shoe	1	92	**99**	96	94
Press Stapler	1	87	**93**	90	71
Put Bottles Dustbin	3	84	81	**95**	93
Put Object Cabinet	2	80	88	**94**	91
Rotate QRcode	1	89	89	**93**	87
Scan Object	2	72	67	**89**	**89**
Shake Bottle Horizontally	1	99	**100**	**100**	**100**
Shake Bottle	1	99	**100**	**100**	**100**
Stack Blocks Three	3	91	91	95	**98**
Stack Blocks Two	2	97	91	**100**	**100**
Stack Bowls Three	3	77	79	80	**85**
Stack Bowls Two	2	95	**98**	92	95
Stamp Seal	1	79	93	90	**96**
Turn Switch	1	62	**84**	61	76
Macro average	–	82.74	88.66	91.88	**93.14**