DO WORLD-ACTION MODELS NEED DENSE WORLD CONTEXT? PROGRESSIVE SPARSE FORCING FOR ACTION DITS

Anonymous authors
Paper under double-blind review

ABSTRACT

World-Action Models (WAMs) built on Diffusion Transformers (DiTs) have become a powerful framework for embodied control, generating actions by denoising action tokens conditioned on visual world states. Across diverse visual conditioning layouts, existing WAMs typically ground action prediction through dense action-to-world attention at every DiT layer. This raises an underexplored question: which visual tokens do noisy action tokens attend to during denoising? Analyzing WAMs under mainstream conditioning layouts, we find that dense conditioning often becomes homogenized across depth and task phases, and frequently anchors on task-irrelevant regions rather than action-relevant physical cues. To make visual grounding adaptive, we introduce Progressive Sparse Forcing (PSF), a general action-to-world conditioning framework that converts dense visual-context aggregation into active visual focusing. PSF couples a depth-decaying sparsity schedule with an end-to-end learned action-world router, forcing noisy action tokens to select denoising-relevant visual context from the original training objective. On robotic manipulation, PSF improves closed-loop success on LIBERO and substantially strengthens zero-shot spatial robustness on LIBERO-Plus. On autonomous driving, it improves long-horizon planning performance across 4K-100K training clips from the PhysicalAI-AV dataset. Beyond accuracy, PSF reaches strong performance *with fewer action denoising steps* and reduces inference computation via sparse conditioning. Code will be publicly released.

1 INTRODUCTION

World-Action Models (WAMs) built on Diffusion Transformers (DiTs) (Peebles & Xie, 2023) have emerged as a powerful paradigm for embodied AI, enabling visuomotor policies for robotic manipulation (Agarwal et al., 2026; Ye et al., 2026b) and autonomous driving (Gui et al., 2026; Liu et al., 2026; Shi et al., 2026). By casting action generation as a denoising process conditioned on visual world states, these models provide a unified interface between high-dimensional observations and executable controls (Wang et al., 2026; Shen et al., 2026). Existing WAMs differ mainly in how visual context is exposed to the action branch. In *first-frame conditioning*, noisy action tokens are denoised conditioned only on the initial observation (Yuan et al., 2026); in *future-state conditioning*, the action branch additionally leverages future video tokens, either through clean teacher-forced states for inverse dynamics (Pai et al., 2025; Li et al., 2026) or through jointly denoised video-action streams (Ye et al., 2026b). Despite these architectural differences, most WAMs share a common design principle: action tokens densely attend to their designated visual tokens across DiT layers.

This dense action-to-world conditioning has been highly effective, yet it leaves a basic aspect of WAMs underexplored: *which visual details are actually used by deep action tokens to denoise executable controls?* Precise control often depends on highly localized physical cues, such as object identity, grasp/contact regions, target placement, nearby traffic participants, or lane-level geometry. However, standard WAMs typically rely on stacking dense action-to-world attention layers, where every action token is given access to the full designated visual sequence throughout the network. This layer-invariant visibility pattern leaves the burden of evidence selection entirely to the learned attention weights. It remains unclear whether *such passive dense access* is sufficient for noisy action tokens at *different depths and denoising states* to isolate the physical cues needed for accurate con-

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.

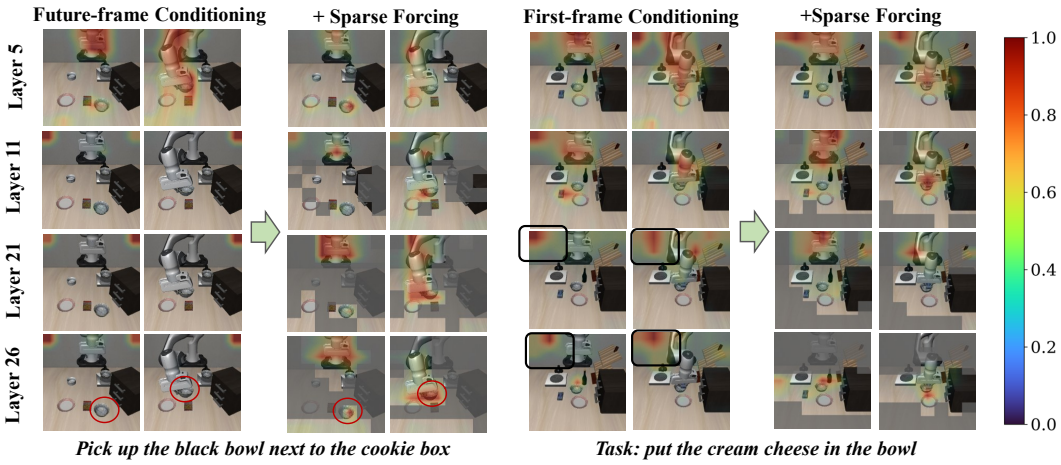


Figure 1: **Action-to-world attention maps in dense WAMs.** We visualize the attention values from action tokens to visual tokens in two representative dense WAM architectures: a first-frame conditioned FastWAM (Yuan et al., 2026) model (Right) and a non-autoregressive future-state conditioning variant following Lingbot-VA (Li et al., 2026) (Left). In both architectures, deep action-to-world attention remains similar across layers and execution phases, and often concentrates on salient static regions (e.g., the robot base or table edges) rather than task-relevant objects or contact regions.

trol. Prior studies on video-generation DiTs show that dense attention can become structurally redundant and dominated by repeated local patterns (Yuan et al., 2024; Chen et al., 2026). Meanwhile, recent analyses of Vision-Language-Action models suggest that action representations often exhibit dispersed or weakly grounded visual attention (Song et al., 2026). Together, these findings raise doubts that simply stacking dense action-to-world attention layers reliably produces action-aligned visual grounding. However, the action-to-world pathway in WAMs remains largely uncharacterized, despite the action-relevant visual context varying with layer depth, denoising state, and task phase.

To investigate this issue, we analyze action-to-world attention maps across representative first-frame and future-state WAMs. As shown in Figure 1, the dense conditioning pathway exhibits two recurring structural limitations across architectures and task phases.

- **Attention Homogenization across Depth and Time**. As illustrated in Figure 1, the action-to-world attention patterns exhibit high consistency across the depth of the network. Across distinct deep layers in both WAMs, the models' focus remains remarkably similar, suggesting limited depth-wise differentiation in how visual context is aggregated. Furthermore, this structural similarity persists throughout the manipulation task execution process. Together, these observations indicate that under dense coupling, deep noisy action tokens often remain in a static, broad conditioning regime, motivating a mechanism that progressively refines their visual focus.
- **"A-to-V attention sinks" in Deep Layers**. More critically, deep action tokens often do not concentrate on localized, task-relevant regions. Instead of consistently attending to instruction-specified objects, contact points, or goal locations, a substantial portion of action-to-world attention is repeatedly assigned to visually salient but task-agnostic regions, such as the empty background areas in Fig. 1. We refer to these as "action-to-visual (A-to-V) attention sinks": visual tokens that repeatedly attract action attention across layers or task phases. Such sinks are not necessarily erroneous, as they may carry aggregated scene information after visual self-attention. However, sink-dominated conditioning provides limited spatial specificity for physical control, and can underemphasize the local object or contact cues that determine precise actions. This motivates a mechanism that preserves global context while promoting task-aligned visual focus.

These findings reveal that dense WAMs do not consistently develop adaptive visual focus. In common instantiations of WAMs, policies are trained with flow matching toward a supervised velocity target. Our key insight is that this structure naturally supports depth-adaptive visual focusing: *deeper noisy action states lie closer in the residual computation path to the supervised velocity read-*

out. Since this readout determines the clean-action estimate, these deeper action states can provide stronger cues for selecting denoising-relevant visual tokens. Appendix A.3 formalizes this intuition with a conditional flow-matching bound. This leads to the core question addressed in this work:

Can we adapt action-to-world attention to action-relevant visual tokens by progressively enforcing sparsity along depth in action DiTs?

To answer this question, we propose Progressive Sparse Forcing (PSF), an action-centric conditioning mechanism that reconfigures action-to-world attention inside action DiTs. PSF introduces two synergistic components. First, the Depth-Aware Action-World Router (DAAR) actively estimates visual-token relevance from the denoising action state along DiT depth. By integrating these routing scores into the main action-to-world attention logits, DAAR allows the original denoising loss to directly shape sparse visual-token routing. Second, the Progressive Sparsity Schedule (PSS) orchestrates this mechanism by systematically shrinking the retained visual context as the network deepens. Together, PSF transforms dense visual-context aggregation into an end-to-end learned visual focusing process, requiring no external grounding labels or auxiliary reconstruction objectives. Our core contributions are summarized as follows:

- We *reveal a shared structural limitation of WAMs*: deep action-to-world attention becomes homogenized across layers and task phases, and frequently anchors on visually salient but task-agnostic regions rather than action-relevant physical cues.
- We *introduce Progressive Sparse Forcing* (PSF), a general action-to-world conditioning framework for action DiTs across WAM layouts. By progressively enforcing sparse action-to-world visibility through an end-to-end learned router, PSF converts dense visual aggregation into active visual focusing, enabling noisy action tokens to select denoising-relevant visual context.
- We *instantiate PSF across mainstream WAM layouts*, including first-frame and future-state conditioning, and validate it across *indoor robotic manipulation and outdoor autonomous driving*. PSF improves closed-loop manipulation on LIBERO and substantially strengthens zero-shot spatial robustness on LIBERO-Plus. For long-horizon autonomous driving, PSF consistently improves trajectory prediction across *data scales from 4K to 100K clips*. Moreover, with only two action-denoising steps, DriveWAM+PSF *achieves a* $2.94\times$ *speedup* over the default DriveWAM schedule while *improving FDE@4s by 9.7%*, and its sparse action-to-world pathway scales favorably under large visual contexts.

2 Action-to-World Conditioning in WAMs

We consider WAMs built on DiTs, where a sequence of noisy action tokens is iteratively refined into executable actions under visual conditioning. Let $A \in \mathbb{R}^{N_a \times D}$ denote action tokens and $V \in \mathbb{R}^{N_v \times D}$ denote visual tokens. The visual sequence can be partitioned as $V = [V_o, V_f]$, where V_o denotes the observed visual state and V_f denotes future visual tokens when available. Mainstream WAMs differ primarily in the candidate visual set exposed to the action DiTs: first-frame conditioned models (Yuan et al., 2026; Ye et al., 2026a) use $V_c = V_o$, whereas future-state conditioned models use $V_c = V_o \cup V_f$. The future tokens V_f can be provided as clean future states in an inverse-dynamics fashion (Li et al., 2026; Pai et al., 2025), or jointly denoised with actions in a joint-denoising fashion (Ye et al., 2026b; Bi et al., 2026). Across these layouts, action prediction is grounded by an action-to-world attention pathway. At each DiT layer, action queries attend to visual keys and values under a base visibility mask M_{dense}, which encodes the valid candidate tokens in V_c. This pathway is expressed as

$$\text{Attn}_{\text{dense}}(A, V_c) = \text{Softmax}\left(\frac{(AW_Q)(V_cW_K)^\top}{\sqrt{d_k}} + M_{\text{dense}}\right)(V_cW_V), \tag{1}$$

where $M_{\text{dense}} \in \mathbb{R}^{N_a \times |V_c|}$ is written as an additive mask for mathematical convenience, assigning $-\infty$ to invalid visual tokens. Efficient boolean or block-mask implementations are mathematically equivalent. In standard WAMs, this dense visibility pattern is reused across layers, leaving visual-context aggregation to the soft cross-attention weights alone. PSF modifies this shared conditioning pathway by replacing the static mask M_{dense} with a layer-dependent, action-conditioned sparse visibility mask $\mathcal{M}_l$. Unlike the dense mask, $\mathcal{M}_l$ is actively routed from the current action state and optimized through the primary denoising loss, allowing deep action layers to access a compact set of denoising-relevant visual tokens.

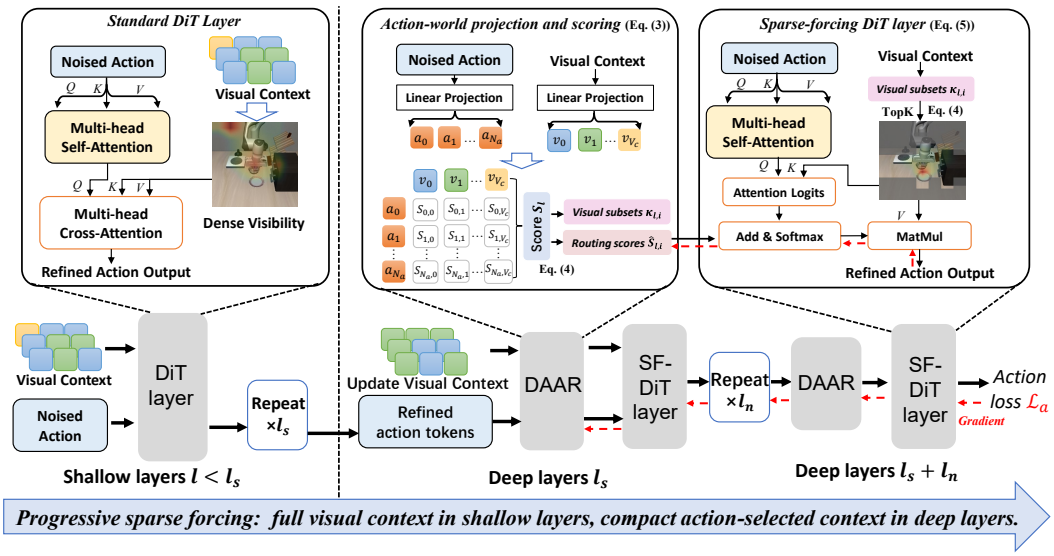


Figure 2: **Overview of Progressive Sparse Forcing.** Standard dense WAM layers expose the full visual context to action tokens. PSF keeps this dense visibility in shallow layers ($l < l_s$), and progressively sparsifies action-to-world attention in deeper layers. At sparsification boundaries, DAAR computes action-world relevance from current-layer features (Eq. 3), gathers compact visual subsets, and injects selected routing scores into sparse-forcing attention logits (Eq. 5).

Preliminaries and related work. Detailed preliminaries on DiTs, flow matching, and first-frame/future-state WAM layouts are provided in Appendix A. Appendix B situates PSF relative to existing WAMs and attention-related issues in VLAs and video-generation DiTs.

3 Progressive Sparse Forcing

Here, we formalize Progressive Sparse Forcing (PSF), which sparsifies the fixed action-to-world conditioning M_{dense} in deep DiT layers via layer-dependent visibility masks $\mathcal{M}_l$. We first describe the design principle of this reconfiguration, then introduce the Depth-Aware Action-World Router (DAAR) and the Progressive Sparsity Schedule (PSS) that jointly construct $\mathcal{M}_l$. Finally, we discuss how the routing is trained end-to-end through the original action denoising objective.

Design principle. PSF is motivated by a residual-path view of flow-matching action DiTs. *Within one denoising forward pass, action states are refined through depth before the final velocity readout.* Thus, deeper action states are closer in the computation graph to the supervised readout that determines the clean-action estimate. Let h_l be the action state at layer l and let P denote the final velocity readout applied diagnostically to h_l. Under the linear flow-matching path, the corresponding clean-action endpoint estimate is $\hat{x}_{1,l} = x_t + (1-t)P(h_l)$. Under a Lipschitz readout and bounded residual updates $h_{k+1} = h_k + \Delta_k$, Appendix A.3 shows the conditional bound

$$\left(\mathbb{E}\left[\|\hat{x}_{1,l} - x_1\|^2 \mid t\right]\right)^{1/2} \le (1-t)\left(\epsilon_t + L_P \sum_{k=l}^{L-1} d_{k,t}\right), \tag{2}$$

where ϵ_t is the final velocity error and $d_{k,t}$ bounds the residual update at layer k. The decreasing residual-tail term motivates using deeper action states for selective visual routing, without guaranteeing monotonic prediction improvement across layers. PSF therefore retains broad visual access in shallow layers and progressively restricts action-to-world visibility in deeper layers. The resulting visual focusing behavior is evaluated both quantitatively and qualitatively in Sec. 4.

Depth-Aware Action-World Router (DAAR). As illustrated in Fig. 2, DAAR constructs the layer-wise sparse visibility pattern $\mathcal{M}_l$. At a sparsification layer l, it dynamically selects the subset of

candidate visual tokens most relevant to each action token. To make routing aware of the current denoising state, DAAR uses the current-layer, timestep-modulated hidden states before attention. Let $h^a_{l,i}$ and $h^v_{l,j}$ denote those of action token a_i and visual token $v_j \in V_c$, respectively.

Action-world projection and scoring. Motivated by token-level relevance estimation in vision-language alignment (Radford et al., 2021), DAAR scores how useful each visual token is for the current action denoising state. Rather than using a single similarity head, we adopt a lightweight multi-head routing design (Liu et al., 2025), allowing different heads to capture complementary action-world relations such as object identity, contact geometry, and goal regions. Specifically, DAAR first projects action and visual features into H_r routing heads:

$$[q^1_{l,i}, \dots, q^{H_r}_{l,i}] = h^a_{l,i} W^r_A, \quad [k^1_{l,j}, \dots, k^{H_r}_{l,j}] = h^v_{l,j} W^r_V, \quad \omega_{l,i} = h^a_{l,i} W^r_\omega.$$

It then computes the action-world relevance matrix $S_l \in \mathbb{R}^{N_a \times |V_c|}$ as

$$S_{l,ij} = \log \sum_{h=1}^{H_r} \exp\left(\frac{\langle q^h_{l,i}, k^h_{l,j} \rangle}{\sqrt{d_r}} + \omega^h_{l,i} \right), \tag{3}$$

where $q^h_{l,i}, k^h_{l,j} \in \mathbb{R}^{d_r}$ are projected routing features, $\omega^h_{l,i}$ is an action-dependent head weight, and W^r_A, W^r_V, W^r_ω are learnable routing projections. The log-sum-exp aggregation acts as a soft head-wise vote, allowing each action token to adaptively emphasize the relevance heads most useful for its visual selection. Thus, $S_{l,ij}$ measures action-to-world relevance under the current denoising state.

Compact subset selection. At layer l, PSS assigns a retention ratio $r_l \in (0, 1)$, which specifies the fraction of valid visual candidates retained for routing. To preserve the temporal and conditioning constraints of the original WAM layout, DAAR applies Top-K selection only within the valid candidate set $\mathcal{V}_i = \{j \mid M_{\text{dense},ij} \neq -\infty\}$ for each action token a_i:

$$\mathcal{K}_{l,i} = \text{TopK}_{j \in \mathcal{V}_i} \left(S_{l,ij}, N_{l,i} \right), \qquad \hat{S}_{l,i} = S_{l,i}[\mathcal{K}_{l,i}]. \tag{4}$$

Here, $N_{l,i} = \lfloor r_l |\mathcal{V}_i| \rfloor$ is the number of retained visual tokens, $\mathcal{K}_{l,i}$ denotes the selected visual token indices for action token a_i, and $\hat{S}_{l,i}$ denotes the corresponding selected routing scores. DAAR then gathers the compact key-value subset indexed by $\mathcal{K}_{l,i}$ for the subsequent sparse-forcing layer.

Sparse-forcing DiT layer. The Top-K indices $\mathcal{K}_{l,i}$ determine which visual tokens remain visible to action token a_i, but this discrete routing alone does not provide a continuous gradient path to guide the relevance computation in Eq. 3. To let the sparse routing behavior be guided by the original denoising objective, PSF converts the original action-to-world attention into a sparse-forcing attention operation: it gathers only the selected visual hidden states and injects the corresponding DAAR relevance scores into the main action-to-world attention logits. Let $\hat{H}^v_{l,i} = h^v_{l,\mathcal{K}_{l,i}}$ denote the ordered selected visual hidden states for action token a_i. The sparse-forcing attention is

$$\text{Attn}_{\text{PSF}}(a_i, V_c) = \text{Softmax}\left(\frac{(h^a_{l,i} W_Q)(\hat{H}^v_{l,i} W_K)^\top}{\sqrt{d_k}} + \lambda \hat{S}_{l,i} \right) (\hat{H}^v_{l,i} W_V), \tag{5}$$

where W_Q, W_K, W_V are attention projections of backbone DiT and learnable λ controls the forcing strength. Gradients from the denoising loss reach DAAR via $\hat{S}_{l,i}$. With at most K tokens per action, the attention cost drops from $\mathcal{O}(N_a|V_c|)$ to $\mathcal{O}(N_a K)$.

Progressive Sparsity Schedule (PSS). While DAAR defines how to route visual tokens at an activated layer, PSS specifies *when* routing is activated and *how* the retained visual context shrinks with depth. Let $\mathcal{B} = \{b_1, \dots, b_m\}$ denote the sparsification boundaries after the dense shallow stage ($b_1 = l_s$), and let $1 > \rho_1 > \cdots > \rho_m > 0$ be their target retention ratios. During the forward pass, when the network reaches boundary b_k, DAAR is activated with retention ratio ρ_k to compute the selected indices $\mathcal{K}_{b_k,i}$ and routing scores $\hat{S}_{b_k,i}$ for each action token according to Eqs. 3 and 4. This sparse configuration is then reused until the next boundary. For any layer $l \in [b_k, b_{k+1})$, PSS sets

$$r_l = \rho_k, \qquad \mathcal{K}_{l,i} = \mathcal{K}_{b_k,i}, \qquad \hat{S}_{l,i} = \hat{S}_{b_k,i}. \tag{6}$$

This interval-wise persistence avoids unstable layer-by-layer re-selection and allows action representations to integrate the selected visual context over multiple DiT blocks. The monotonic decay of

ρ_k implements progressive sparse forcing: earlier sparse stages retain broader visual context, while deeper stages restrict $\mathcal{M}_l$ to increasingly compact visual context. Thus, later action layers must operate under progressively tighter visual visibility, enforcing DAAR to retain evidence most relevant to action prediction. The full forward pass is summarized in Algorithm 1 in Appendix C.1.

End-to-end training. PSF is trained with the original WAM objective in a standard end-to-end fashion, without any auxiliary routing loss:

$$\mathcal{L}_{\text{WAM}} = \mathcal{L}_a + \beta_v \mathcal{L}_v, \tag{7}$$

where $\mathcal{L}_a$ and $\mathcal{L}_v$ denote action and video denoising losses, and β_v follows the baseline WAM. Since the selected routing scores are injected into Eq. 5, DAAR lies on the original computation graph and is optimized end-to-end by the same denoising signal. In this way, the original objective drives sparse forcing under progressively restricted visual visibility, without any auxiliary supervision.

Instantiation across WAM layouts. PSF is a general action-DiT reconfiguration that can be integrated across different WAMs through their shared action-to-world attention pathway. Across all cases, PSF preserves the backbone, training hyperparameters, and original visual layout of the dense baseline, including temporal or causal visibility constraints. Details are provided in Appendix C.

4 Experiments

4.1 Experimental Setup

Baselines and integration. We evaluate PSF on representative WAM layouts. For robotic manipulation, we follow the FastWAM codebase (Yuan et al., 2026) and instantiate all baselines with a pretrained Wan2.2-5B (Wan et al., 2025) DiT backbone. Specifically, FastWAM serves as the first-frame conditioned baseline; IDM-WAM is implemented as a non-autoregressive inverse-dynamics variant following the Lingbot-VA architectural design (Li et al., 2026); and JD-WAM is implemented as a joint video-action denoising variant following the DreamZero-style coupled denoising design (Ye et al., 2026b). For autonomous driving, we adopt DriveWAM as a strong IDM-style baseline. In all cases, the WAM layout, denoising/noising strategy (e.g., asymmetric schedules (Ye et al., 2026b) or future-state noising (Li et al., 2026)), training objective, and hyperparameters are kept identical to the corresponding baseline. Detailed configurations are provided in Appendix D.

Domains and protocols. For robotic manipulation, we evaluate closed-loop execution on LIBERO (Liu et al., 2023) and out-of-distribution (OOD) test on LIBERO-Plus (Fei et al., 2025). For autonomous driving, we evaluate long-horizon trajectory planning on the 1,700-hour PhysicalAI-Autonomous-Vehicles dataset (Wang et al., 2025), using 4K, 20K, and 100K training clips and reporting ADE/FDE (Average/Final Displacement Error) at 3s and 4s horizons.

Evaluation Topics. With the above baselines, our evaluation is organized around four questions:

Q1. Generality.	Does PSF improve action prediction across WAM layouts and domains?
Q2. Action Grounding.	Does PSF help action denoising focus on useful visual context?
Q3. OOD Robustness.	Does PSF compromise zero-shot spatial robustness under OOD shifts?
Q4. Ablation & efficiency.	Which designs drive the gains, and how does sparsity affect inference cost?

4.2 Experimental Results

Answer 1: ***PSF generalizes across WAM layouts and domains***.
Robot Manipulation. Table 1 shows that PSF improves the overall closed-loop success rate for every evaluated WAM layout. These paired gains are obtained without changing the backbone or training hyperparameters, indicating that reconfiguring action-to-world attention improves policy execution even on strong baselines. Notably, improvements in the Long-task for FastWAM and JD-WAM suggest that focused deep-layer conditioning can be beneficial in multi-stage manipulation.
Autonomous driving. We next test whether the same reconfiguration scales to real-world autonomous driving. Table 2 reports trajectory planning results on the PhysicalAI-Autonomous-Vehicles dataset. Following the DriveWAM protocol, we use its released 4K, 20K, and 100K training splits and the same evaluation setting. PSF improves trajectory prediction accuracy at both 3s and 4s horizons. The gains also persist across all three data scales, suggesting that our action-conditioned sparse forcing remains beneficial as the amount of training data increases from 4K to 100K clips.

Table 1: **Closed-loop manipulation success rates (%) on LIBERO.** † denotes our reproduced baseline under the same training settings.

Method	Spatial	Object	Goal	Long	Overall
$\pi_{0.5}$ (Intelligence et al., 2025)	98.8	98.2	98.0	92.4	96.9
Motus (Bi et al., 2026)	96.8	99.8	96.6	97.6	97.7
Abot-M0 (Yang et al., 2026)	98.8	99.8	99.0	96.6	98.6
Future-state conditioning (inverse dynamics / joint denoising)					
IDM-WAM†	98.4	98.6	98.0	97.8	98.2
IDM-WAM + PSF	**99.4**	**98.8**	**98.4**	**98.0**	**98.7**
JD-WAM†	**99.0**	**100.0**	98.4	95.6	98.3
JD-WAM + PSF	98.6	**100.0**	**99.0**	**97.8**	**98.9**
First-frame conditioning					
FastWAM†	96.6	99.2	97.6	92.8	96.6
FastWAM + PSF	**97.0**	**99.8**	**98.0**	**95.4**	**97.6**

Table 2: **Autonomous driving performance on PhysicalAI-Autonomous-Vehicles. Left:** 3s and 4s trajectory prediction on the 100K split. The suffix −b denotes the best result across different denoising-step settings. Published baselines are included for context (Bartoccioni et al., 2025; Wang et al., 2025). **Right:** 4s trajectory prediction across 4K, 20K, and 100K training clips. lower is better.

Method	3s Horizon		4s Horizon	
	ADE	FDE	ADE	FDE
VaVAM	2.31	4.32	-	-
Alpamayo-1.5	0.80	2.31	1.44	4.18
DriveWAM†	0.48	1.35	0.84	2.48
+PSF	**0.44**$_{-8.3\%}$	**1.29**$_{-4.4\%}$	**0.79**$_{-6.0\%}$	**2.39**$_{-3.6\%}$
DriveWAM−b†	0.48	1.31	0.83	2.38
+PSF−b	**0.44**$_{-8.3\%}$	**1.21**$_{-7.6\%}$	**0.76**$_{-8.4\%}$	**2.21**$_{-7.1\%}$

Data Size	Method	ADE@4s	FDE@4s
4k	DriveWAM†	1.01	2.95
	+PSF	**0.95**$_{-5.9\%}$	**2.69**$_{-8.8\%}$
20k	DriveWAM†	0.94	2.65
	+PSF	**0.89**$_{-5.3\%}$	**2.42**$_{-8.6\%}$
100k	DriveWAM†	0.84	2.48
	+PSF	**0.79**$_{-6.0\%}$	**2.39**$_{-3.6\%}$

Answer 2: ***PSF makes visual conditioning more effective for action denoising***
The driving results in Table 2 use the default DriveWAM inference schedule, with 10 action and 3 video denoising steps. We test the hypothesis: ***If PSF provides better visual grounding for the action branch, strong action prediction could be achievable with fewer denoising steps***.

Quantitative: fewer video denoising steps. We first ablate the inference-time video denoising budget in Appendix D.2. Reducing it from 3 to 1 step slightly improves the dense baseline (FDE 2.47→2.45), while yielding a larger gain for PSF (2.39→2.32). This suggests that probabilistic future-state noising makes fewer video-denoising steps sufficient for both models, while PSF benefits more through action-conditioned visual focusing. Thus, all subsequent action-denoising ablations use 1 video denoising step.

Table 3: **Latency under reduced denoising steps.** video-denoising steps fixed to 1.

Setting	Latency (ms)↓	ADE@4s↓	FDE@4s↓
DriveWAM	1285.6	0.84	2.48
DriveWAM−b	936.9	0.83	2.38
+PSF 2-step	**436.7**	0.79$_{-6.3\%}$	2.24$_{-9.7\%}$
+PSF−b	573.1	**0.76**$_{-9.5\%}$	**2.21**$_{-10.9\%}$

Quantitative: fewer action denoising steps. Figure 3 further varies the action-denoising steps from 10 to 2. PSF achieves lower trajectory error with substantially fewer action refinements. At two action-denoising steps, DriveWAM+PSF surpasses the best DriveWAM result over the entire sweep, reducing ADE@4s and FDE@4s by 3.5% and 6.0%, respectively. The best PSF setting further improves over the best dense baseline by 8.2% in ADE@4s and 7.2% in FDE@4s. Table 3 confirms the latency benefit: PSF 2-step is nearly 3× faster than default DriveWAM while improving ADE/FDE, and the best PSF setting remains faster and more accurate than the best baseline setting. Together, these results indicate that compact, action-selected visual conditioning makes each action refinement step more effective, allowing the action branch to reach strong long-horizon prediction accuracy with fewer denoising iterations.

Qualitative: visual grounding analysis. Beyond quantitative evaluation, we visualize PSF attention maps across manipulation rollouts to examine how the selected visual context evolves over time. As shown in Fig. 4, the sparse masks $\mathcal{M}_l$ induced by Eq. 4 evolve with execution, focusing on

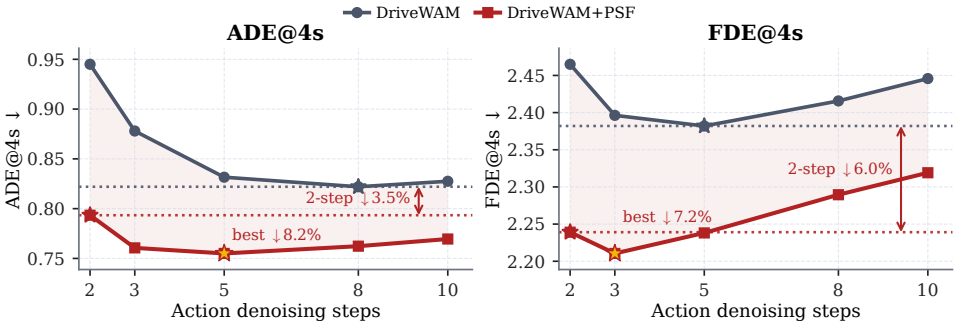


Figure 3: **Action denoising-step efficiency on PhysicalAI-Autonomous-Vehicles.** We vary the action-denoising budget at inference from 10 to 2 steps. PSF reaches strong 4s trajectory prediction accuracy with substantially fewer action steps.

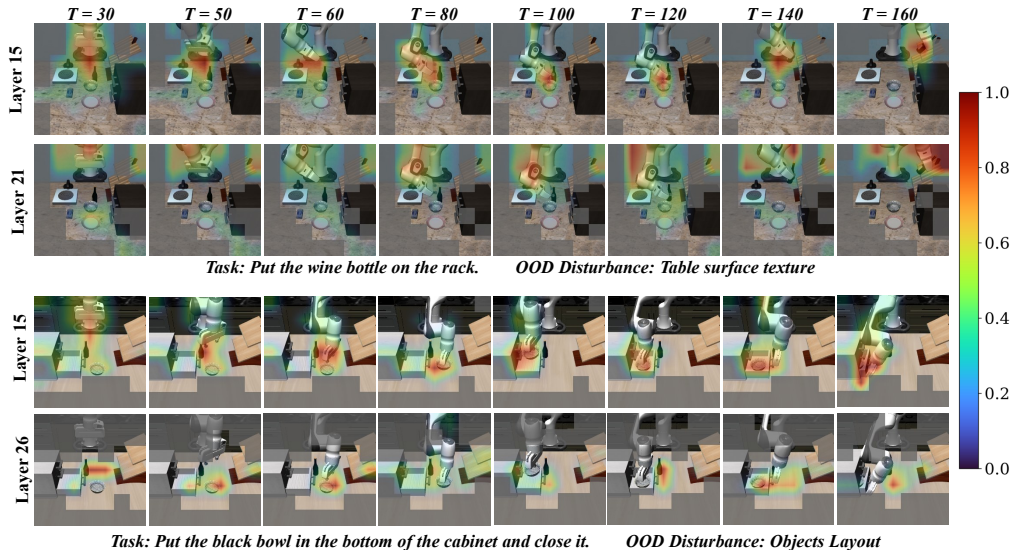


Figure 4: **Qualitative PSF grounding.** Visualization of PSF masks across manipulation rollouts. The gray grids denote visual tokens masked out by PSF, while the heatmap intensity represents normalized attention values from action tokens to the unmasked visual tokens.

task-relevant objects, interaction regions, and goal areas, even under texture disturbances. This supports our interpretation that PSF induces action-aligned visual focus without auxiliary supervision. Additional visualizations across tasks and disturbances are provided in the Appendix D.3.

Answer 3: ***PSF preserves and improves zero-shot robustness under OOD shifts***
Zero-shot robustness. We evaluate whether restricting visual visibility compromises zero-shot generalization. Table 4 evaluates LIBERO-Plus across seven perturbation axes. The improvements are most pronounced under spatial shifts such as camera viewpoint and object layout, where the locations of action-relevant evidence change. On camera and layout shifts, PSF improves FastWAM by +24.0/+21.3, IDM-WAM by +11.2/+6.5, and JD-WAM by +12.5/+0.8, leading to OOD gains of +13.8, +5.0, and +1.1, respectively. These results show that PSF improves spatial OOD robustness across WAM layouts, with the largest gains on FastWAM and IDM-WAM, while preserving the zero-shot robustness of JD-WAM. On the full 10,030-task LIBERO-Plus, PSF improves FastWAM and IDM-WAM by 9.2 and 2.2 points, respectively, while JD-WAM remains near parity (Table 9).

Answer 4: ***Ablations support the key PSF designs, and sparsity improves inference scaling***
Ablation of progressive forcing. As shown in Table 5 (Left), we ablate the sparse-forcing schedule on the first-frame FastWAM baseline (no PSF). The 1-stage variant, where DAAR is activated once and the selected visual subset is reused for later layers, already outperforms the dense baseline on

Table 4: **Zero-shot robustness on the LIBERO-Plus spatial suite.** We report success rates (%) over 2,402 OOD tasks spanning seven perturbation axes. Green subscripts denote absolute PSF gains on key spatial axes and overall performance.

Method	Background	Camera	Language	Lighting	Layout	Init States	Sensor	Overall
π_0 (Black et al., 2024)	90.7	17.8	58.8	89.7	89.1	6.6	90.9	60.7
π_0-Fast (Pertsch et al., 2025)	97.7	87.2	84.2	37.0	95.5	26.9	93.2	74.4
Future-state conditioning (joint denoising/inverse dynamics)								
JD-WAM	77.1	43.4	97.7	99.0	92.7	80.3	61.8	78.6
JD-WAM + PSF	77.1	**55.9**$_{+12.5}$	96.7	98.6	**93.5**$_{+0.8}$	77.5	59.6	**79.8**$_{+1.2}$
IDM-WAM	70.9	50.5	94.4	97.6	89.9	74.9	67.2	77.9
IDM-WAM + PSF	79.8	**61.7**$_{+11.2}$	94.4	100.0	**96.4**$_{+6.5}$	**78.6**$_{+3.5}$	70.7	**82.9**$_{+5.0}$
First-Frame Conditioning Baseline								
FastWAM	76.0	23.9	62.1	96.2	63.4	41.1	51.0	57.3
FastWAM + PSF	73.3	**47.9**$_{+24.0}$	76.2	97.3	**84.7**$_{+21.3}$	**62.9**$_{+21.8}$	60.1	**71.1**$_{+13.8}$

Table 5: **Ablation and inference efficiency. Left:** Sparse-forcing schedule ablation on LIBERO, reporting closed-loop success rates (%) on the 10-task suite and overall performance. **Right:** Inference compute scaling for action-to-world attention under increasing visual context sizes. PSF reduces GFLOPs in all settings, with wall-clock gains emerging at larger visual contexts.

Variant	Goal	Long	Overall
Baseline	97.6	92.8	96.6
1-stage DAAR	96.4	93.4	96.7
2-stage PSS	97.6	94.0	97.4
3-stage PSS	**98.0**	**95.4**	**97.6**
5-stage PSS	97.0	94.2	97.2

Resolution × Views	Forward p50 (ms) ↓		Speedup	Δ GFLOPs Reduction
	DriveWAM	+PSF		
480p × 1	96.7	106.8	0.91×	17.5
1080p × 1	135.1	119.3	1.13×	41.4
720p × 3	145.4	122.5	1.19×	55.2
1440p × 1	184.5	129.9	**1.42** ×	**73.6**

long-horizon and overall tasks, showing the benefit of active visual selection over passive dense aggregation. Adding PSS further improves performance, with the 3-stage schedule achieving the best results. Importantly, all sparse-forcing variants surpass the dense baseline, confirming PSF's effectiveness across routing schedules. Together, these results support our central premise that sparse forcing improves WAM performance by encouraging action-aligned visual focusing.

Inference efficiency. PSF improves WAM inference efficiency in two complementary ways. **First**, stronger action-conditioned visual focusing reduces the required denoising steps. As shown in Table 3, the 2-step PSF setting achieves a $2.94\times$ speedup, while improving FDE@4s by 9.7% against the baseline. **Second**, PSF offers more favorable compute scaling in deployment settings, where high-resolution or multi-view driving/robotic inputs substantially increase the visual token count. Its sparse $\mathcal{O}(N_a K)$ action-to-world attention scales better than dense $\mathcal{O}(N_a|V_c|)$ aggregation. Table 5 (Right) shows that, across increasing resolutions and view counts, PSF reduces GFLOPs in all settings and achieves up to a $1.42\times$ speedup with 73.61 fewer GFLOPs. At small single-view inputs, sparse K/V selection can incur modest routing overhead, offsetting wall-clock gains despite lower GFLOPs. Dedicated fused sparse kernels could further improve efficiency across settings.

5 Conclusions

We introduced Progressive Sparse Forcing (PSF), a general action-to-world conditioning mechanism that can be integrated into action DiTs across different WAM layouts. PSF rethinks dense action-to-world attention over the full visual context by progressively constraining deep action tokens to compact, action-selected visual subsets, with routing learned only through the original denoising objective. Across robotic manipulation and autonomous driving, PSF improves closed-loop performance on LIBERO, zero-shot spatial robustness on LIBERO-Plus, and long-horizon trajectory planning on the PhysicalAI-Autonomous-Vehicles dataset. Beyond improving accuracy, PSF makes the action denoising process itself more effective, achieving strong prediction quality with substantially fewer action-denoising steps. Its sparse action-to-world pathway further reduces attention cost as visual context grows, making PSF well suited for high-resolution and multi-camera deployment settings. Overall, our results suggest that active, action-conditioned visual focusing is a promising design principle for building more accurate, efficient, and deployable World-Action Models.

AI USE STATEMENT

In this work, we used generative AI tools to assist with writing-related tasks, including grammar correction, wording refinement, LaTeX formatting, and improving the clarity and conciseness of paper text. We did not use generative AI tools to generate research ideas, design the proposed method, produce experimental results, create datasets, or make scientific claims. All AI-assisted text was reviewed and edited by the authors. All technical claims, mathematical formulations, experimental numbers, tables, and figures were produced and verified by the authors. We take full responsibility for the final content of this work, including any text revised with the aid of generative AI.

ETHICS STATEMENT

This work studies Progressive Sparse Forcing (PSF), an action-to-world conditioning mechanism for World-Action Models in robotic manipulation and autonomous driving. The goal of PSF is to improve action prediction, visual grounding, and inference efficiency by making visual conditioning more action-aligned. Our experiments are conducted on standard simulated manipulation benchmarks and offline autonomous-driving evaluation settings, and do not involve human subjects or the collection of private personal data. Nevertheless, embodied control and autonomous-driving systems can have safety-critical consequences if deployed in the real world. The results in this paper should therefore not be interpreted as sufficient validation for direct deployment. Any real-world use of WAMs or PSF-enhanced policies should require additional safety validation, monitoring, fail-safe mechanisms, and compliance with relevant domain regulations. PSF does not introduce new sensing modalities, privileged annotations, or auxiliary supervision, but more efficient embodied models should still be deployed responsibly by practitioners and organizations.

REPRODUCIBILITY STATEMENT

We provide detailed information to support reproducibility. The PSF algorithm, including dynamic action-aware routing, Top-K visual-token selection, progressive sparse forcing, and sparse action-to-world attention, is described in the main method section and further specified in Appendix C. Appendix A summarizes the relevant DiT, flow-matching, and WAM formulations. Appendix D and Appendix D.1 document the experimental settings, including model layouts, optimizer settings, denoising schedules, inference budgets, and PSF hyperparameters. Appendix D.2 reports the full video/action denoising-step ablation for autonomous driving, and Appendix D.3 provides additional qualitative visual-grounding results. Code will be publicly released.

REFERENCES

Niket Agarwal, Arslan Ali, Jon Allen, Martin Antolini, Adeline Aubame, Alisson Azzolini, Junjie Bai, Maciej Bala, Yogesh Balaji, Josh Bapst, et al. Cosmos 3: Omnimodal World Models for Physical AI. *arXiv preprint arXiv:2606.02800*, 2026.

Florent Bartoccioni, Elias Ramzi, Victor Besnier, Shashanka Venkataramanan, Tuan-Hung Vu, Yihong Xu, Loick Chambon, Spyros Gidaris, Serkan Odabas, David Hurych, et al. VaViM and VaVAM: Autonomous Driving through Video Generative Modeling. *arXiv preprint arXiv:2502.15672*, 2025.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 35101–35113, June 2026.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A Vision-Language-Action Flow Model for General Robot Control. *arXiv preprint arXiv:2410.24164*, 2024.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. SAM 3: Segment Anything with Concepts. In *International conference on learning representations*, volume 2026, pp. 138846–138923, 2026.

Pengtao Chen, Xianfang Zeng, Maosen Zhao, Mingzhu Shen, Wei Cheng, Gang Yu, and Tao Chen. Sparse-vDiT: Unleashing the Power of Sparse Attention to Accelerate Video Diffusion Transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 2957–2965, 2026.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, et al. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

Xingtai Gui, Meijie Zhang, Tianyi Yan, Wencheng Han, Jiahao Gong, Feiyang Tan, Cheng-zhong Xu, and Jianbing Shen. Bridging Scene Generation and Planning: Driving with World Model via Unifying Vision and Motion Representation. *arXiv preprint arXiv:2603.14948*, 2026.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: A Vision-Language-Action Model with Open-World Generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, et al. Causal World Modeling for Robot Control. *arXiv preprint arXiv:2601.21998*, 2026.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for Generative Modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. DeepSeek-V3.2: Pushing the Frontier of Open Large Language Models. *arXiv preprint arXiv:2512.02556*, 2025.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

Mengmeng Liu, Diankun Zhang, Jiuming Liu, Jianfeng Cui, Hongwei Xie, Guang Chen, Hangjun Ye, Michael Ying Yang, Francesco Nex, and Hao Cheng. DriveVA: Video Action Models are Zero-Shot Drivers. *arXiv preprint arXiv:2604.04198*, 2026.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow. *arXiv preprint arXiv:2209.03003*, 2022.

Jonas Pai, Liam Achenbach, Victoriano Montesinos, Benedek Forrai, Oier Mees, and Elvis Nava. mimic-video: Video-Action Models for Generalizable Robot Control Beyond VLAs. *arXiv preprint arXiv:2512.15692*, 2025.

William Peebles and Saining Xie. Scalable Diffusion Models with Transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182, 2023.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual Reasoning with a General Conditioning Layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient Action Tokenization for Vision-Language-Action Models. *arXiv preprint arXiv:2501.09747*, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning Transferable Visual Models From Natural Language Supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

Qiuhong Shen, Shihua Zhang, Yue Liao, Qi Li, Zhenxiong Tan, Shizun Wang, Shuicheng Yan, and Xinchao Wang. World Action Models: A Survey. *arXiv preprint arXiv:2606.20781*, 2026.

Chen Shi, Jinrui Xu, Shaoshuai Shi, Kehua Sheng, Bo Zhang, and Li Jiang. DriveWAM: Video Generative Priors Enable Scalable World-Action Modeling for Autonomous Driving. *arXiv preprint arXiv:2605.28544*, 2026.

Wenxuan Song, Ziyang Zhou, Han Zhao, Jiayi Chen, Pengxiang Ding, Haodong Yan, Yuxin Huang, Feilong Tang, Donglin Wang, and Haoang Li. ReconVLA: Reconstructive Vision-Language-Action Model as Effective Robot Perceiver. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 18549–18557, 2026.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *Advances in neural information processing systems*, 30, 2017.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and Advanced Large-Scale Video Generative Models. *arXiv preprint arXiv:2503.20314*, 2025.

Siyin Wang, Junhao Shi, Zhaoyang Fu, Xinzhe He, Feihong Liu, Chenchen Yang, Yikang Zhou, Zhaoye Fei, Jingjing Gong, Jinlan Fu, et al. World Action Models: The Next Frontier in Embodied AI. *arXiv preprint arXiv:2605.12090*, 2026.

Yan Wang, Wenjie Luo, Junjie Bai, Yulong Cao, Tong Che, Ke Chen, Yuxiao Chen, Jenna Diamond, Yifan Ding, Wenhao Ding, et al. Alpamayo-R1: Bridging Reasoning and Action Prediction for Generalizable Autonomous Driving in the Long Tail. *arXiv preprint arXiv:2511.00088*, 2025.

Tianyi Yan, Tao Tang, Xingtai Gui, Yongkang Li, Jiasen Zheng, Weiyao Huang, Lingdong Kong, Wencheng Han, Xia Zhou, Xueyang Zhang, Yifei Zhan, Kun Zhan, Cheng-zhong Xu, and Jianbing Shen. AD-R1: Closed-Loop Reinforcement Learning for End-to-End Autonomous Driving with Impartial World Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 1085–1095, June 2026.

Yandan Yang, Shuang Zeng, Tong Lin, Xinyuan Chang, Dekang Qi, Junjin Xiao, Haoyun Liu, Ronghan Chen, Yuzhi Chen, Dongjie Huo, et al. ABot-M0: VLA Foundation Model for Robotic Manipulation with Action Manifold Learning. *arXiv preprint arXiv:2602.11236*, 2026.

Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Hao Li, Hengtao Li, Jie Li, Jindi Lv, Jingyu Liu, et al. GigaWorld-Policy: An Efficient Action-Centered World–Action Model. *arXiv preprint arXiv:2603.17240*, 2026a.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World Action Models are Zero-shot Policies. *arXiv preprint arXiv:2602.15922*, 2026b.

Hanyang Yu, Haitao Lin, Jingbo Zhang, Wenyao Zhang, Chenghao Gu, Heng Li, and Ping Tan. MaskWAM: Unifying Mask Prompting and Prediction for World-Action Models. *arXiv preprint arXiv:2606.13515*, 2026.

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.

Zhihang Yuan, Hanling Zhang, Pu Lu, Xuefei Ning, Linfeng Zhang, Tianchen Zhao, Shengen Yan, Guohao Dai, and Yu Wang. DiTFastAttn: Attention Compression for Diffusion Transformer Models. *Advances in Neural Information Processing Systems*, 37:1196–1219, 2024.

A PRELIMINARIES

A.1 DIFFUSION TRANSFORMERS

Diffusion Transformers (DiTs) (Peebles & Xie, 2023) form the backbone of modern generative models, replacing U-Net architectures (Ho et al., 2020) with scalable Transformer blocks (Vaswani et al., 2017). At layer $l \in \{1, \dots, L\}$, input noised latent $X^{(l)}$ are modulated by the diffusion timestep and condition embeddings via Adaptive Layer Normalization (AdaLN) (Perez et al., 2018), yielding $\tilde{X}^{(l)}$. Self-attention then projects $\tilde{X}^{(l)}$ into query, key, and value matrices $Q^{(l)} = \tilde{X}^{(l)} W_Q$, $K^{(l)} = \tilde{X}^{(l)} W_K$, and $V^{(l)} = \tilde{X}^{(l)} W_V$, computing $\mathrm{Attn}^{(l)} = \mathrm{Softmax}(Q^{(l)}(K^{(l)})^\top / \sqrt{d_k}) V^{(l)}$, where W_Q, W_K, W_V are learnable weights and d_k is the scaling factor.

Beyond intra-sequence self-attention, generating targeted outputs requires injecting external conditions. These conditions are typically integrated via AdaLN, dedicated cross-attention layers, or by concatenating context tokens for in-context conditioning (Rombach et al., 2022; Peebles & Xie, 2023). In recent WAMs, the primary condition comprises visual observations, such as initial frames or predicted future states. Regardless of the specific implementation, the injection of this visual context fundamentally relies on an *action-to-world* attention pathway where action queries attend to visual keys and values. Crucially, standard WAMs maintain this dense cross-modal coupling uniformly across all layers (Li et al., 2026; Ye et al., 2026b). While effective for initial global grounding, naively stacking dense attention may provide limited pressure to progressively distill the fine-grained, object-level visual details required for precise physical actions, as suggested by Fig. 1. This motivates an explicit reconfiguration of the action-to-world attention pathway.

A.2 FLOW MATCHING FOR ACTION GENERATION

To generate action trajectories, modern policies often employ Flow Matching (Lipman et al., 2022; Liu et al., 2022), learning a vector field to map a simple prior $x_0 \sim \mathcal{N}(0, I)$ to the data distribution $x_1 \sim p_{\text{data}}$. Specifically, it constructs a linear path $x_t = (1-t)x_0 + tx_1$ for $t \in [0, 1]$ and trains a network (e.g., DiT) to regress the target velocity $u_t = dx_t/dt = x_1 - x_0$ conditioned on context c (e.g., visual observations, task prompt). The model is optimized by minimizing:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{t, x_0, x_1, c}\left[\left\| v_\theta(x_t, t, c) - (x_1 - x_0) \right\|_2^2\right]. \tag{8}$$

During inference, a clean action trajectory x_1 is generated by sampling x_0 and solving the ordinary differential equation (ODE) $dx_t/dt = v_\theta(x_t, t, c)$. Within a single forward pass at a fixed flow time t, action representations are updated across DiT depth before the final velocity readout. The following analysis formalizes why later action states provide a principled basis for depth-adaptive visual routing, without treating network depth as flow time.

A.3 FLOW-MATCHING VIEW OF DEPTH-ADAPTIVE ROUTING

We provide a conditional analysis that motivates applying stronger visual routing at deeper action layers. The analysis uses the flow-matching path defined in Appendix A and the objective in Eq. 8.

Endpoint interpretation of velocity prediction. For the linear path $x_t = (1-t)x_0 + tx_1$, the target velocity is $u = x_1 - x_0$. At a fixed flow time t and context c, the squared-error optimal velocity field is

$$v^\star(x_t, t, c) = \mathbb{E}[u \mid x_t, t, c]. \tag{9}$$

Since $x_1 = x_t + (1-t)u$, any velocity estimate $\hat{u}$ induces a clean-action endpoint estimate

$$\hat{x}_1 = x_t + (1-t)\hat{u}. \tag{10}$$

Thus, for any intermediate action state h_l, if we diagnostically apply the final velocity readout P and define

$$\hat{u}_l = P(h_l), \qquad \hat{x}_{1,l} = x_t + (1-t)\hat{u}_l, \tag{11}$$

then the endpoint error and velocity error satisfy the exact identity

$$\|\hat{x}_{1,l} - x_1\| = (1-t)\|\hat{u}_l - u\|. \tag{12}$$

This identity is used only for analysis. PSF does not apply intermediate readouts during training or inference.

Proposition 1: Conditional depth bound. Let h_l denote the action state at layer l, and write the residual updates as

$$h_{k+1} = h_k + \Delta_k, \qquad k = l, \dots, L-1. \tag{13}$$

Assume the final velocity readout P is L_P-Lipschitz on the action representations visited by the forward pass. At a fixed flow time t, define

$$\epsilon_t = \Big(\mathbb{E}[\|P(h_L) - u\|^2 \mid t]\Big)^{1/2}, \qquad d_{k,t} = \Big(\mathbb{E}[\|\Delta_k\|^2 \mid t]\Big)^{1/2}. \tag{14}$$

Then the intermediate endpoint estimate satisfies

$$\Big(\mathbb{E}[\|\hat{x}_{1,l} - x_1\|^2 \mid t]\Big)^{1/2} \le (1-t)\left(\epsilon_t + L_P \sum_{k=l}^{L-1} d_{k,t}\right). \tag{15}$$

Proof. The residual decomposition gives

$$h_L - h_l = \sum_{k=l}^{L-1} \Delta_k. \tag{16}$$

By the triangle inequality and the Lipschitz property of P,

$$\|P(h_l) - u\| \le \|P(h_L) - u\| + L_P \sum_{k=l}^{L-1} \|\Delta_k\|. \tag{17}$$

Applying the conditional L_2 triangle inequality and then using Eq. 12 yields Eq. 15. □

This bound provides a residual-path justification for depth-adaptive routing: deeper action states are closer to the supervised velocity readout in terms of remaining DiT updates, so the bound contains fewer residual terms. The result does not require interpreting depth as flow time and does not claim monotonic improvement of every realized intermediate representation.

Gradient path for selected routing scores. At an activated sparse layer, DAAR first selects a Top-K set $\mathcal{K}_i$ for action token a_i. We analyze one use of a selected routing score in a sparse-forcing attention operation, conditioned on the fixed Top-K set $\mathcal{K}_i$. Let

$$p_j = \frac{\exp(a_j + \lambda s_j)}{\sum_{q \in \mathcal{K}_i} \exp(a_q + \lambda s_q)}, \qquad o = \sum_{j \in \mathcal{K}_i} p_j w_j, \tag{18}$$

where a_j is the backbone attention logit, s_j is the DAAR routing score, and w_j is the visual value vector. Differentiating this attention use gives the local gradient contribution

$$\frac{\partial \mathcal{L}_{\mathrm{FM}}}{\partial s_j} = \lambda p_j \left\langle \nabla_o \mathcal{L}_{\mathrm{FM}},\, w_j - o \right\rangle, \qquad j \in \mathcal{K}_i. \tag{19}$$

Thus, selected DAAR scores receive gradients from the original denoising objective through their effect on sparse-forcing attention. When PSS reuses the same routing scores across multiple layers, the total gradient is the sum of such local contributions over all uses. The Top-K indices are held fixed in this derivative, so Eq. 19 should not be interpreted as a differentiable relaxation of Top-K selection.

Sparse-context preservation. We also state a conditional perturbation bound for sparse visual selection. Consider a full-score reference that uses the same DiT backbone logits and DAAR score bias as PSF, but exposes all valid visual tokens. This is a mathematical comparator, not the original dense WAM baseline.

For action token a_i, let $\mathcal{V}_i$ be the valid visual set and $\mathcal{K}_i \subseteq \mathcal{V}_i$ be the selected set. Define the full-score attention probability

$$\pi_{ij} = \frac{\exp(a_{ij} + \lambda s_{ij})}{\sum_{q \in \mathcal{V}_i} \exp(a_{iq} + \lambda s_{iq})}, \qquad \mu_i = \sum_{j \in \mathcal{V}_i \setminus \mathcal{K}_i} \pi_{ij}, \tag{20}$$

where μ_i is the omitted attention mass. Let o_i^{full} and o_i^{PSF} denote the full-score and sparse attention outputs, respectively.

Proposition 2: Sparse-context preservation. Assume $\|w_j\| \le B$ for every valid visual value vector. Let O concatenate the attention outputs over action queries, and suppose the remaining mapping G from O to the predicted velocity is L_G-Lipschitz when other layer inputs are fixed. Then

$$\|O^{\mathrm{PSF}} - O^{\mathrm{full}}\|_F \le 2B\|\boldsymbol{\mu}\|_2, \qquad \|\hat{u}^{\mathrm{PSF}} - \hat{u}^{\mathrm{full}}\| \le 2L_G B\|\boldsymbol{\mu}\|_2. \tag{21}$$

Consequently,

$$\|\hat{x}_1^{\mathrm{PSF}} - \hat{x}_1^{\mathrm{full}}\| \le (1-t)\, 2L_G B\|\boldsymbol{\mu}\|_2. \tag{22}$$

Proof. The sparse attention output is the full-score distribution renormalized on $\mathcal{K}_i$. Let o_i^{drop} be the normalized value average over omitted tokens when $\mu_i > 0$. Then

$$o_i^{\mathrm{full}} = (1-\mu_i)o_i^{\mathrm{PSF}} + \mu_i o_i^{\mathrm{drop}}. \tag{23}$$

Therefore,

$$\|o_i^{\mathrm{PSF}} - o_i^{\mathrm{full}}\| \le \mu_i \left(\|o_i^{\mathrm{PSF}}\| + \|o_i^{\mathrm{drop}}\|\right) \le 2B\mu_i. \tag{24}$$

Summing over action queries gives the Frobenius bound. Applying the Lipschitz property of G gives the velocity perturbation bound, and Eq. 12 gives the endpoint bound. □

Proposition 2 provides a sufficient, not necessary, condition relative to the full-score reference. It is a local approximation bound for a sparse attention operation, rather than a grounding guarantee for the trained PSF policy. Separately, the quantitative and qualitative results in Sec. 4 evaluate whether the learned sparse routing exhibits action-aligned visual focusing in practice.

A.4 World-Action Models

World-action models (WAMs) enable generalizable control by co-training action generation with video prediction, grounding decision-making in physical scene dynamics. To formalize these architectures, we first introduce the unified observation encoding process, followed by the dual-stream generation mechanism, which naturally diverges into two primary conditioning layouts based on the action-to-world attention masking.

Observation and Context Encoding. To model the continuous physical world within a Transformer, raw sensory inputs must be mapped to a unified token sequence. During training, both the current visual observation and the subsequent future frames are processed through a causal Variational Autoencoder (VAE) (Wan et al., 2025) and patchified into a sequence of video tokens $V \in \mathbb{R}^{N_v \times D}$. We partition this sequence as $V = [V_o, V_f]$, where V_o encapsulates the currently observed state (e.g., the initial frame) and V_f represents the future visual rollout. Simultaneously, the conditioning context $C \in \mathbb{R}^{N_c \times D}$ is constructed to guide the generation. It comprises the task instruction, encoded via a pre-trained language model (e.g., T5 (Raffel et al., 2020)), and the agent's proprioceptive state (e.g., robotic joint angles or ego-vehicle states), embedded via a Multi-Layer Perceptron (MLP). Both modalities are projected into the shared D-dimensional hidden space.

Dual-Stream Action and World Generation. The core generative process is driven by a dual-stream DiT comprising L layers, which jointly process the video tokens V and action tokens as defined in Sec. 2. At each layer, the architecture typically executes a two-step update: first, the visual and action streams actively exchange information to ground action in the physical environment; subsequently, the representations flow into their respective dedicated DiTs to independently perform cross-attention over the context C (e.g., task prompt, robot proprios). These operations repeat across all L layers, after which the final representations are projected to predict their respective target velocities, guided by the flow matching objective (Eq. 8).

To provide a unified formulation for diverse world-action frameworks, we focus on the cross-modal exchange stage, specifically the *action-to-world* attention pathway. Here, action queries extract physical evidence from visual tokens. The cross-modal visibility in this pathway is strictly governed by an effective attention mask, denoted as $M_{A\to V} \in \{0,1\}^{N_a \times N_v}$. Based on how this mask constructs the visual context, mainstream models diverge into two primary paradigms:

(I) First-Frame Conditioning: In this paradigm, action generation relies exclusively on the initial environmental state (Yuan et al., 2026; Ye et al., 2026a). The future visual context is absent ($V_f = \emptyset$), and $M_{A\to V}$ restricts action tokens to attend solely to V_o. This layout requires the policy to infer future dynamics and formulate actions relying entirely on the initial visual observation.

(II) Future-State Conditioning: This paradigm leverages V_f to provide an explicit, rich temporal context, typically instantiated in two ways. In the Inverse Dynamics Model Denoising framework (IDM-WAM) (Pai et al., 2025; Li et al., 2026), V_f comprises *ground-truth future* frames during training, and $M_{A\to V}$ permits full access to $V_o \cup V_f$ to learn precise state transitions. In *Joint Denoising* models (JD-WAM) (Ye et al., 2026b), V_f consists of noisy future tokens generated concurrently with the actions, and $M_{A\to V}$ enables the co-evolution of noisy visual and action representations.

Despite these distinct layouts, the visual context exposed to action tokens is ultimately fixed by the dense attention mask $M_{A\to V}$, forcing action queries to uniformly attend to observed first-frame or predicted future visual sequences across every layer. This rigid, layer-invariant coupling can lead to the attention homogenization and structural redundancies discussed in Fig. 1, motivating our proposed progressive sparse reconfiguration.

B Additional Related Work

World-Action Models. Recent WAMs extend diffusion or flow-matching Transformers from visual generation to embodied action prediction, often by modeling future visual states and action trajectories jointly (Li et al., 2026; Ye et al., 2026b; Yuan et al., 2026; Ye et al., 2026a; Bi et al., 2026; Yan et al., 2026; Pai et al., 2025). These models differ in their conditioning layouts, including first-frame conditioning, inverse-dynamics-style future-state conditioning, and joint video-action denoising; Appendix A.4 provides a unified formulation. PSF is complementary to these architectures: it preserves the backbone, visual tokenizer, and training objectives, while reconfiguring the shared action-to-world attention pathway in deep action DiT layers.

Attention redundancy in video-generation DiTs. A growing line of work studies efficiency and redundancy in DiTs for image and video generation. Prior analyses show that dense attention in video DiTs can become structurally redundant, with attention maps dominated by repeated local patterns or slowly changing correlations across layers and timesteps (Yuan et al., 2024; Chen et al., 2026). Existing methods often exploit this redundancy through sparse attention, token pruning, or approximate attention computation to reduce generation cost. While these methods improve efficiency for visual generation, they primarily target visual-token interactions and do not address which visual tokens should condition action prediction in WAMs. In contrast, PSF targets the action-to-world pathway and uses evolving action states to select visual context for denoising actions.

Visual grounding in Vision-Language-Action models. Vision-Language-Action (VLA) models have also raised concerns about weak or dispersed visual grounding in action prediction. Recent studies find that action representations may attend broadly or inconsistently to task-relevant regions, motivating additional grounding mechanisms (Song et al., 2026). Some remedies introduce external spatial guidance, such as segmentation or object masks from foundation models (Carion et al., 2026; Yu et al., 2026), while others add auxiliary reconstruction or grounding losses (Song et al., 2026). These approaches can improve grounding but require additional supervision, preprocessing, or training objectives. PSF instead asks whether action-aligned visual focusing can be induced directly from the original denoising objective and aligned with the denoising process. By enforcing depth-decaying sparsity and injecting routing scores into the action-to-world attention logits, PSF learns sparse visual routing end-to-end without external grounding labels or auxiliary reconstruction losses.

C Additional PSF Details

C.1 Forward Algorithm

Algorithm 1 summarizes the forward pass of Progressive Sparse Forcing (PSF). The algorithm starts from the base dense mask M_{dense} defined by the underlying WAM layout. Before the first sparsification boundary, PSF is identical to the dense baseline. At each sparsification boundary, DAAR computes action-world relevance scores, selects compact visual subsets, and stores the selected indices and routing scores as the active sparse visibility pattern $\mathcal{M}_l$. This sparse configuration is then reused until the next boundary.

Algorithm 1 Forward Pass of Progressive Sparse Forcing

Require: Action tokens A, candidate visual tokens V_c, denoising/flow time t
Require: Dense base mask M_{dense}, sparsification boundaries $\mathcal{B} = \{b_1, \dots, b_m\}$
Require: Retention ratios $\{r_1, \dots, r_m\}$ with $1 > r_1 > \cdots > r_m > 0$

Initialize active sparse indices $\mathcal{K}_i \leftarrow \emptyset$ and routing scores $\hat{S}_i \leftarrow \emptyset$
Initialize sparse flag $\text{Sparse} \leftarrow \text{False}$
for DiT layer $l = 1$ to L **do**
  Compute current-layer, timestep-modulated action and visual features $\tilde{h}_A^l, \tilde{h}_V^l$
  if $l = b_k \in \mathcal{B}$ **then**
    Compute DAAR relevance scores S_{ij}^l over visual tokens valid under M_{dense}
    For each action token a_i, define $\mathcal{V}_i = \{j \mid M_{\text{dense},ij} \neq -\infty\}$ and select Top-K visual indices:

$$\mathcal{K}_i \leftarrow \text{TopK}_{j \in \mathcal{V}_i}(S_{ij}^l, K_i), \qquad K_i = \lfloor r_k |\mathcal{V}_i| \rfloor.$$

    Gather selected routing scores $\hat{S}_i \leftarrow S_{i,\mathcal{K}_i}^l$
    Set $\text{Sparse} \leftarrow \text{True}$
  end if
  if $\text{Sparse} = \text{False}$ **then**
    Compute dense action-to-world attention using M_{dense}
  else
    Gather current-layer visual keys and values using $\mathcal{K}_i$
    Compute sparse-forcing attention with injected scores $\hat{S}_i$
  end if
  Update action and visual tokens through the standard DiT block (e.g., aggregate task prompt)
end for
return Denoised action prediction

Between two adjacent sparsification boundaries, the selected index set $\mathcal{K}_i$ is kept fixed. In each subsequent sparse layer, the same indices gather the current-layer visual keys and values, while the routing scores computed at the boundary are reused as the additive sparse bias. This persistence avoids unstable layer-by-layer re-selection and gives the action stream sufficient depth to integrate the selected visual evidence.

C.2 Training Objective and Stabilization

PSF uses the original WAM training objective defined in Eq. 7 and introduces no auxiliary grounding or routing loss. This appendix details how the sparse routing path receives gradients and how we stabilize early training. A key design of PSF is that DAAR is not used merely as a hard token-pruning module. At an activated layer l, after DAAR selects the compact index set $\mathcal{K}_{l,i}$, the corresponding routing scores $\hat{S}_{l,i}$ are injected into the main action-to-world attention logits, as in Eq. 5. Gradients from the original flow-matching objective, in particular $\mathcal{L}_a$, therefore propagate to DAAR through these selected scores. The Top-K indices are treated as deterministic routing decisions, while the selected score values remain differentiable. This allows the sparse visibility pattern $\mathcal{M}_l$ to be learned end-to-end from the original action prediction objective, without object masks, language grounding labels, or auxiliary reconstruction losses.

To stabilize early training, we apply a sparsity warmup schedule. Let r_k denote the target retention ratio at the sparsification boundary b_k. At training step τ, the effective ratio is

$$r_k(\tau) = (1 - \alpha_\tau) \cdot 1 + \alpha_\tau \cdot r_k, \qquad \alpha_\tau \in [0, 1], \tag{25}$$

where α_τ increases linearly from 0 to 1 during warmup. Thus, PSF starts from dense action-to-world conditioning and gradually enforces the target sparsity schedule, preventing premature removal of visual evidence before the router has learned meaningful action-world relevance.

C.3 LAYOUT-SPECIFIC INSTANTIATIONS

PSF applies to different WAM layouts by changing only the candidate visual set V_c exposed to DAAR. The routing mechanism, sparsity schedule, and sparse-forcing attention layer remain unchanged.

First-frame conditioning. In first-frame-conditioned WAMs, action generation relies only on the observed visual state. The candidate visual set is

$$V_c = V_o.$$

DAAR routes only over the observed visual tokens. This forces deep action layers to extract compact spatial and geometric anchors from the current visual features, without access to future video tokens.

Future-state conditioning: inverse dynamics. In inverse-dynamics-style WAMs, the action branch is conditioned on both observed and future visual states:

$$V_c = V_o \cup V_f.$$

During training, V_f corresponds to teacher-forced future visual tokens, which can be either clean or probabilistically noised (Li et al., 2026). DAAR therefore routes over the full visual transition, allowing action tokens to focus on the parts of the trajectory most informative for action prediction. The original temporal validity constraints of the baseline are preserved through M_{dense}.

Future-state conditioning: joint denoising. In joint video-action denoising WAMs, future visual tokens are noisy and co-evolve with action tokens during denoising. The candidate set remains

$$V_c = V_o \cup V_f,$$

but V_f follows the video diffusion schedule. Since DAAR uses the current-layer, timestep-modulated hidden states, the sparse visibility pattern adapts to the evolving action and visual representations at each activated boundary.

Across all layouts, PSF preserves the original backbone architecture, candidate visual set, and training objective. It only replaces dense action-to-world attention in selected deep layers with the sparse-forcing attention defined in the main text.

D EXPERIMENTAL CONFIGURATIONS

This section provides the implementation details omitted from the main text, including baseline adaptations, model architectures, denoising/noising schedules, training hyperparameters, and inference protocols.

D.1 MODEL AND TRAINING CONFIGURATIONS

We provide the main implementation configurations used in our experiments. Across all comparisons, PSF keeps the backbone architecture, tokenizer, training objective, optimizer, and inference protocol of the corresponding dense WAM baseline unchanged. The only modification is the replacement of selected deep action-to-world attention layers with the PSF sparse-forcing pathway as described in Sec. 3.

Robotic manipulation. For LIBERO and LIBERO-Plus, we evaluate PSF on three WAM layouts: FastWAM, IDM-WAM, and JD-WAM. All three use the same Mixture-of-Transformers backbone with separate video and action DiT experts. The video/action streams have 30 layers, hidden dimensions 3072/1024, 24 attention heads, and FFN dimensions 14336/4096, respectively. Visual observations are encoded into 48-channel WanVideoVAE latents (Wan et al., 2025), and language instructions are represented by T5 features (Raffel et al., 2020). Each observation contains two 224×224 camera views, i.e., a third-person camera and a wrist camera. In all evaluations, the action horizon is 1.6 s, and the replanning interval is 1 s.

The three manipulation baselines differ only in their conditioning layout. FastWAM uses first-frame conditioning, where action tokens attend to the observed visual tokens V_o. IDM-WAM follows

Table 6: Training and evaluation hyperparameters. For robotic manipulation, values are shared by FastWAM, IDM-WAM, and JD-WAM unless explicitly listed separately. PSF uses the same settings as its corresponding dense baseline.

Configuration	Robotic manipulation	Autonomous driving
Benchmarks	LIBERO, LIBERO-Plus	PhysicalAI-AV
Training objective	Flow matching	Flow matching
Optimizer	AdamW	AdamW
AdamW betas	$(0.9, 0.95)$	$(0.9, 0.95)$
Learning rate	1×10^{-4}	1×10^{-5}
Weight decay	1×10^{-2}	0.1
LR schedule	Cosine	Constant
Warmup steps	2000	1000
Batch size	256	48
Training budget	10 epochs	50,000 steps
Precision	bf16	bf16
Default inference steps	10 action denoising steps	10 action denoising steps
Evaluation protocol	50 trials per task, success rate	ADE/FDE at 3s and 4s
Evaluation subset	LIBERO and LIBERO-Plus	1000 test samples (Shi et al., 2026)
Initialization	Wan2.2-TI2V-5B + ActionDiT	Pretrained Lingbot-VA (Li et al., 2026)

the inverse-dynamics setting, where action prediction is conditioned on teacher-forced future visual tokens V_f; following Lingbot-VA (Li et al., 2026), noise is applied to the future visual condition during training. JD-WAM follows the joint video-action denoising setting of DreamZero, where video and action tokens are denoised jointly with asynchronous video/action noise sampling (Ye et al., 2026b). To accommodate this stochastic multi-modal denoising process, we also inject Gaussian noise into the PSF relevance scores during training, which encourages robust routing under varying video-action noise levels.

Autonomous driving. For PhysicalAI-Autonomous-Vehicles, the baseline DriveWAM is initialized from a pretrained Lingbot-VA backbone (Li et al., 2026). It uses a 30-layer DiT with hidden dimension 3072, 24 attention heads, FFN dimension 14336, 48-channel visual latents, and T5 text features. The model jointly predicts future visual latents and ego trajectories under a flow-matching objective in an inverse dynamics fashion. The model predicts 120 future action steps, corresponding to 12 seconds at 10Hz, organized into three 40-step action chunks. During evaluation, after each 4s chunk, the visual K/V cache is refreshed using the ground-truth observation before rolling out the next chunk. Following the baseline (Shi et al., 2026), we therefore report ADE/FDE at 3s and 4s within each chunk, using the corresponding first 30 and 40 future steps.

PSF hyperparameters. For all robotic manipulation baselines and autonomous driving experiments, PSF uses the same three-stage sparsification schedule with boundary layers $\mathcal{B} = \{10, 20, 25\}$ and retention ratios $\{0.8, 0.6, 0.4\}$. The DAAR router uses routing dimension $d_r = 128$ and $H_r = 4$ routing heads, as defined in Eq. 3. The selected routing scores are injected into the action-to-world attention logits through the learnable forcing strength λ in Eq. 5, initialized to $\{0.02, 0.05, 0.10\}$ across the three stages. The sparsity ratio is linearly warmed up during the first 2000 training steps. We do not introduce object masks, grounding annotations, auxiliary reconstruction losses, or extra task-specific supervision.

For DriveWAM, we use the same PSF schedules. The only differences from the manipulation setting are the initial forcing strengths $\lambda = \{0.10, 0.15, 0.20\}$ and the sparsity warmup. All other training and evaluation settings are kept identical to dense DriveWAM. In the denoising-step ablation, DriveWAM and DriveWAM+PSF use the same video-step sweep $v \in \{1, 2, 3\}$, action-step sweep $a \in \{1, 2, 3, 5, 8, 10\}$, and stop video-denoising fraction $f = 0.6$ (Li et al., 2026).

Fair comparison. PSF does not change the DiT depth, hidden dimension, attention heads, visual tokenizer, language encoder, action horizon, optimizer, or training objective of any baseline. It is trained under the same data, optimization, and inference budgets as the corresponding dense WAM.

Table 7: 3-second prediction-horizon results on the PhysicalAI test set (1000 samples, stop denoising fraction f=0.6 (Li et al., 2026; Shi et al., 2026)). We sweep the number of video-denoising steps $v \in \{1, 2, 3\}$ and action-denoising steps $a \in \{1, 2, 3, 5, 8, 10\}$. Both v and a denote denoising-step counts. Each +PSF value carries a teal subscript showing the relative reduction compared with DriveWAM under the same denoising-step setting. Bold indicates the best value.

		ADE@3s (m) ↓		FDE@3s (m) ↓	
v	a	DriveWAM	+PSF	DriveWAM	+PSF
v=1 s	1	0.853	0.683$_{-19.9\%}$	1.886	1.617$_{-14.3\%}$
	2	0.597	0.466$_{-21.9\%}$	1.448	1.258$_{-13.1\%}$
	3	0.535	0.435$_{-18.7\%}$	1.363	1.218$_{-10.7\%}$
	5	0.484	**0.424**$_{-12.4\%}$	1.312	**1.214**$_{-7.4\%}$
	8	**0.467**	0.423$_{-9.3\%}$	**1.309**	1.228$_{-6.1\%}$
	10	0.467	0.426$_{-8.8\%}$	1.319	1.240$_{-6.0\%}$
v=2 s	1	0.866	0.692$_{-20.1\%}$	1.920	1.649$_{-14.1\%}$
	2	0.610	0.477$_{-21.8\%}$	1.483	1.295$_{-12.7\%}$
	3	0.546	0.446$_{-18.4\%}$	1.393	1.253$_{-10.1\%}$
	5	0.493	0.435$_{-11.9\%}$	1.335	1.247$_{-6.5\%}$
	8	0.474	0.433$_{-8.6\%}$	1.327	1.260$_{-5.1\%}$
	10	0.474	0.436$_{-8.0\%}$	1.336	1.270$_{-5.0\%}$
v=3 s	1	0.871	0.697$_{-19.9\%}$	1.935	1.665$_{-13.9\%}$
	2	0.616	0.483$_{-21.6\%}$	1.499	1.312$_{-12.5\%}$
	3	0.551	0.451$_{-18.1\%}$	1.407	1.270$_{-9.7\%}$
	5	0.498	0.440$_{-11.6\%}$	1.347	1.263$_{-6.2\%}$
	8	0.478	0.438$_{-8.2\%}$	1.338	1.275$_{-4.7\%}$
	10	0.478	0.441$_{-7.7\%}$	1.347	1.285$_{-4.6\%}$

Thus, the reported gains isolate the effect of replacing passive dense action-to-world attention with progressive, action-conditioned sparse visual routing.

D.2 Additional ablation and evaluations.

Video and action denoising step ablation. We conduct a full denoising-step ablation over both the video and action branches. Tables 7 and 8 report ADE/FDE at the 3s and 4s prediction horizons on the PhysicalAI test set, sweeping the number of video-denoising steps $v \in \{1, 2, 3\}$ and action-denoising steps $a \in \{1, 2, 3, 5, 8, 10\}$. Here, both v and a denote denoising-step counts rather than time durations. Across all evaluated denoising budgets, DriveWAM+PSF consistently improves over the corresponding DriveWAM baseline. Notably, PSF achieves better trajectory prediction than the default dense DriveWAM setting with fewer action-denoising steps, showing that action-aligned visual focusing reduces the need for repeated DiT refinement. This further supports our interpretation that PSF makes each action-denoising step more effective, improving prediction accuracy while enabling faster inference.

Table 8: 4-second prediction-horizon results on the PhysicalAI test set (1000 samples, stop denoising fraction f=0.6 (Li et al., 2026; Shi et al., 2026)). We sweep the number of video-denoising steps $v \in \{1, 2, 3\}$ and action-denoising steps $a \in \{1, 2, 3, 5, 8, 10\}$. Both v and a denote denoising-step counts. Each +PSF value carries a teal subscript showing the relative reduction compared with DriveWAM under the same denoising-step setting. Bold indicates the best value.

		ADE@4s (m) ↓		FDE@4s (m) ↓	
v	a	DriveWAM	+PSF	DriveWAM	+PSF
v=1 s	1	1.258	1.050$_{-16.6\%}$	2.967	2.629$_{-11.4\%}$
	2	0.945	0.793$_{-16.0\%}$	2.465	2.239$_{-9.2\%}$
	3	0.878	0.761$_{-13.4\%}$	2.396	**2.211**$_{-7.8\%}$
	5	0.832	**0.755**$_{-9.3\%}$	**2.382**	2.238$_{-6.0\%}$
	8	**0.822**	0.762$_{-7.3\%}$	2.416	2.290$_{-5.2\%}$
	10	0.828	0.770$_{-7.0\%}$	2.446	2.319$_{-5.2\%}$
v=2 s	1	1.279	1.068$_{-16.5\%}$	3.022	2.689$_{-11.0\%}$
	2	0.966	0.814$_{-15.7\%}$	2.516	2.303$_{-8.5\%}$
	3	0.896	0.781$_{-12.8\%}$	2.437	2.271$_{-6.8\%}$
	5	0.845	0.774$_{-8.4\%}$	2.411	2.293$_{-4.9\%}$
	8	0.833	0.780$_{-6.3\%}$	2.438	2.339$_{-4.1\%}$
	10	0.838	0.787$_{-6.1\%}$	2.466	2.367$_{-4.0\%}$
v=3 s	1	1.288	1.078$_{-16.3\%}$	3.050	2.719$_{-10.8\%}$
	2	0.976	0.825$_{-15.5\%}$	2.544	2.335$_{-8.2\%}$
	3	0.905	0.791$_{-12.5\%}$	2.462	2.302$_{-6.5\%}$
	5	0.853	0.784$_{-8.1\%}$	2.432	2.322$_{-4.5\%}$
	8	0.839	0.790$_{-5.9\%}$	2.456	2.367$_{-3.7\%}$
	10	0.844	0.794$_{-5.9\%}$	2.483	2.394$_{-3.6\%}$

Table 9: Full 10,030-task LIBERO-Plus results. We report success rates (%) across four task groups and the overall average. Subscripts on +PSF rows denote changes relative to the corresponding dense WAM baseline. Bold indicates the higher score within each dense/PSF pair.

Method	**Spatial**	**Object**	**Goal**	**Long**	**Overall**
JD-WAM	78.6	**67.8**	**56.0**	60.4	**65.7**
JD-WAM + PSF	**79.8**$_{+1.2}$	66.9$_{-0.9}$	55.2$_{-0.8}$	**60.7**$_{+0.3}$	65.6$_{-0.1}$
IDM-WAM	77.9	68.2	52.8	**65.6**	66.1
IDM-WAM + PSF	**83.0**$_{+5.1}$	**68.9**$_{+0.7}$	**56.1**$_{+3.3}$	65.2$_{-0.4}$	**68.3**$_{+2.2}$
FastWAM	57.3	67.4	46.2	42.0	53.2
FastWAM + PSF	**71.1**$_{+13.8}$	**81.0**$_{+13.6}$	**51.8**$_{+5.6}$	**45.7**$_{+3.7}$	**62.4**$_{+9.2}$

Full LIBERO-Plus evaluation. Table 9 reports the full 10,030-task LIBERO-Plus results. The full-suite evaluation is consistent with the main-text OOD analysis: PSF provides the largest robustness gains for FastWAM, improving the overall success rate by +9.2 points and substantially boosting spatial and object generalization. On IDM-WAM, PSF also improves the overall score by +2.2 points, with clear gains on spatial and goal tasks. For JD-WAM, PSF preserves the overall zero-shot robustness at near parity.

D.3 Additional Qualitative Visual Grounding Analysis

This appendix presents per-task attention evolution for the PSF model under each out-of-distribution (OOD) disturbance type of the LIBERO-Plus benchmark. For every task, we visualize the model's action-token-to-visual-token attention at four DiT layers and at eight evenly spaced time points (in seconds) of a successful closed-loop rollout. The heatmap is min-max normalized to $[0, 1]$ within each task, and the visual tokens masked by PSF are grayed out. Across all disturbance types and scene geometries, the attended region tracks the manipulated object/contact zone and shifts with the trajectory, indicating that the sparse router's selectivity is task-driven rather than a fixed dataset bias.

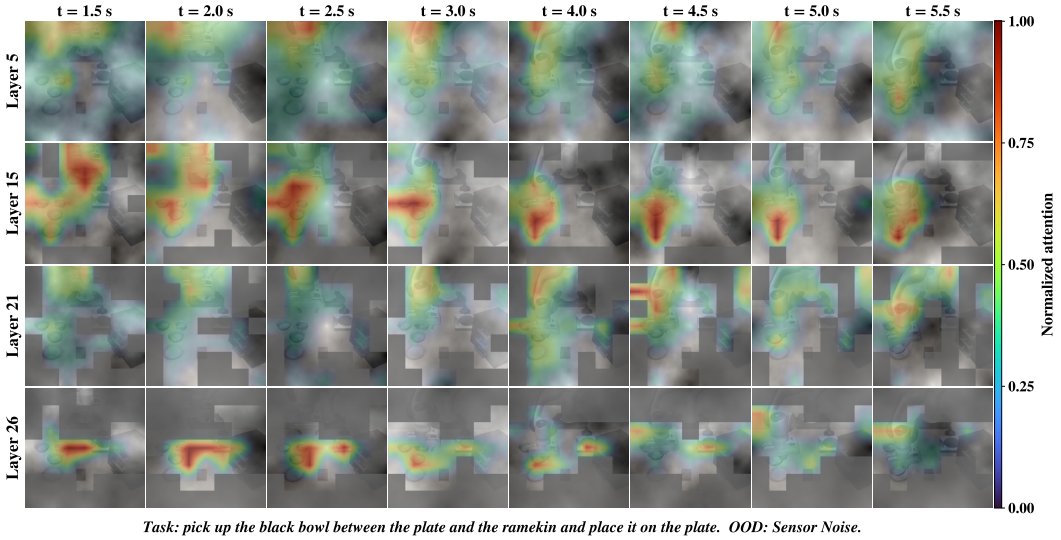


Figure 5: Attention evolution under **sensor-noise** OOD perturbation for the task *pick up the black bowl between the plate and the ramekin and place it on the plate*. Despite heavy visual sensor noise, attention stays anchored to the bowl and the plate across the whole trajectory.

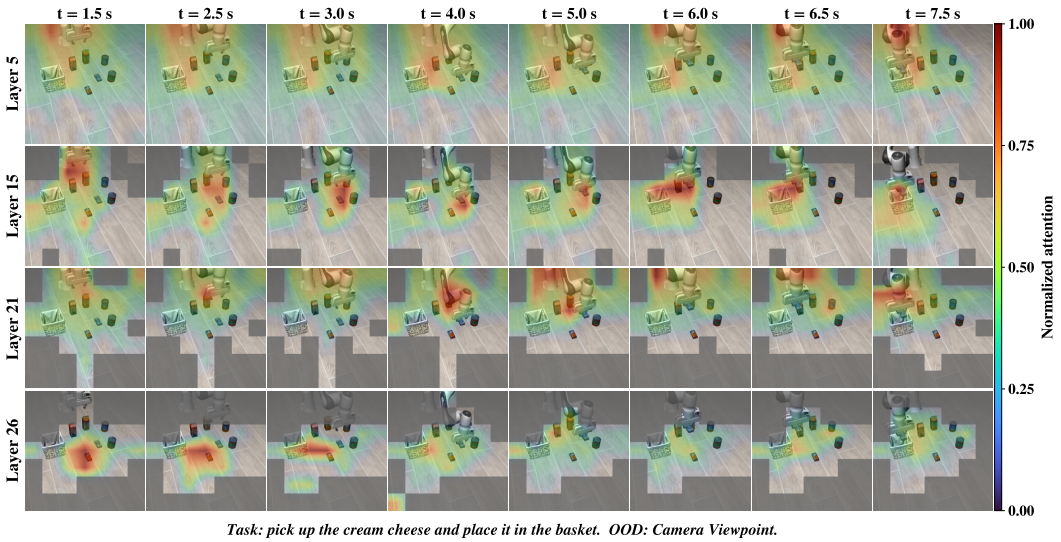


Figure 6: **Camera-viewpoint** OOD perturbation (move x=20, y=15) for the task *pick up the cream cheese and place it in the basket*. The attended region follows the cream-cheese and the basket, confirming that the sparse router tracks the manipulated object rather than a fixed camera-pose.

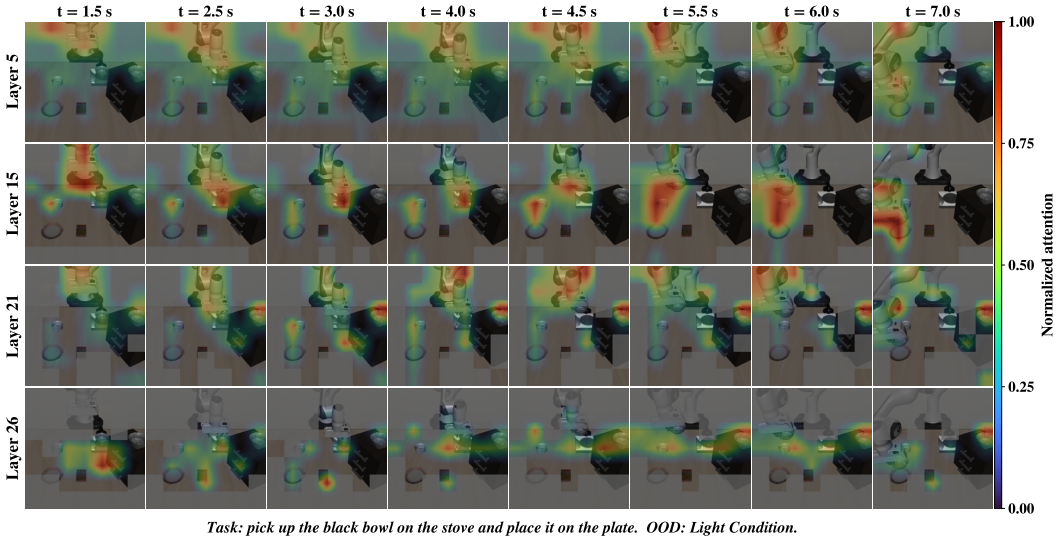


Figure 7: Attention under **lighting** OOD perturbation (light condition 2) for the task *pick up the black bowl on the stove and place it on the plate*, where the manipulated object lies on the *left* side of the scene. Attention concentrates on the bowl/stove region and shifts toward the plate during placement, showing temporally evolving, manipulation-aligned selectivity under illumination change.

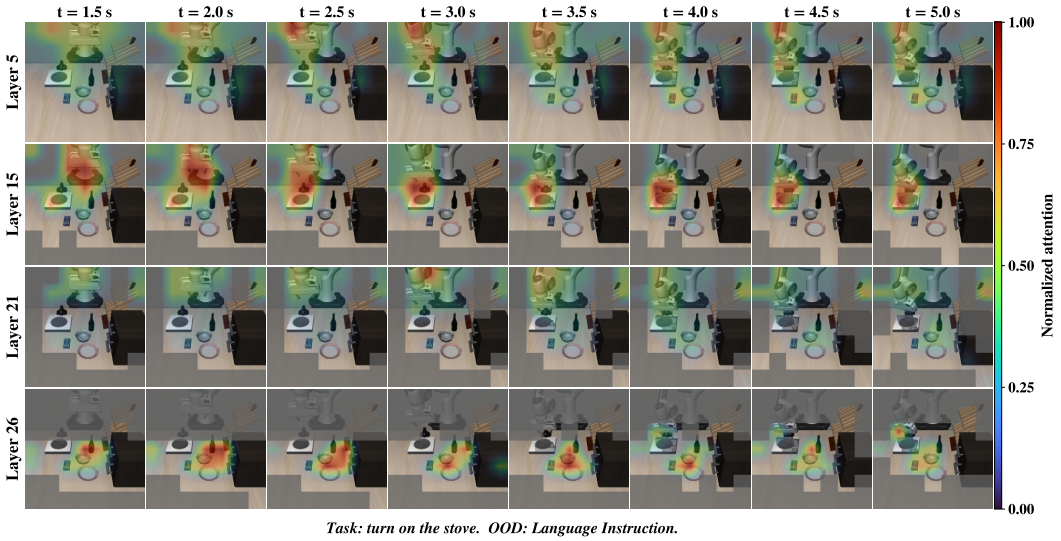


Figure 8: Attention under **language-instruction** OOD perturbation (paraphrase variant 5) for the task *turn on the stove*. Although the instruction wording differs from training, the attended region consistently covers the stove knob throughout the brief interaction, demonstrating that the router's selectivity is grounded in the scene/task and is invariant to the linguistic surface form.

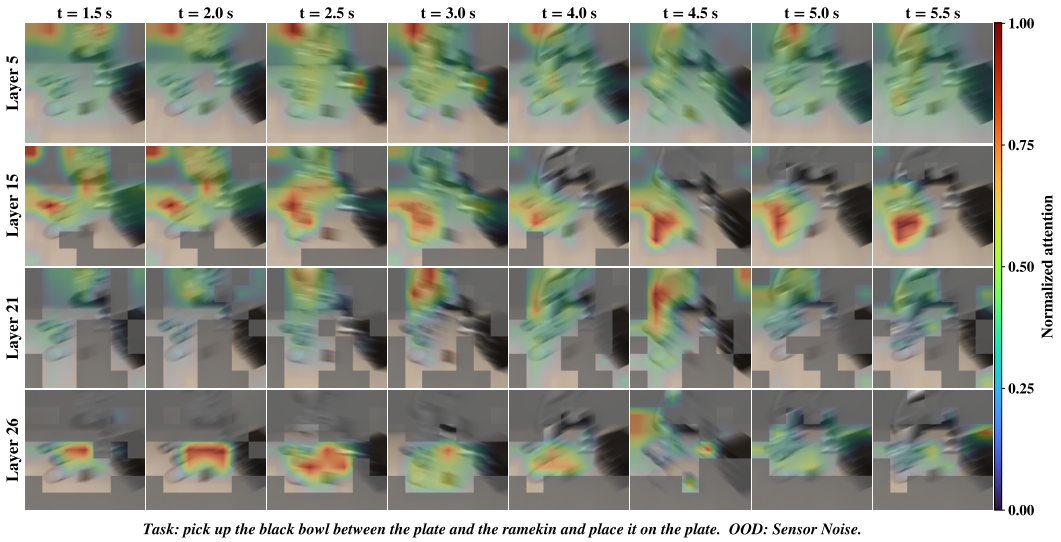


Figure 9: Attention under **sensor-noise** OOD perturbation (noise level 10, milder than Fig. 5) for the same task *pick up the black bowl between the plate and the ramekin and place it on the plate*. Compared with the heavier-noise counterpart, attention is more sharply localized on the bowl while the contact-anchored pattern is preserved, indicating graceful degradation under increasing noise.

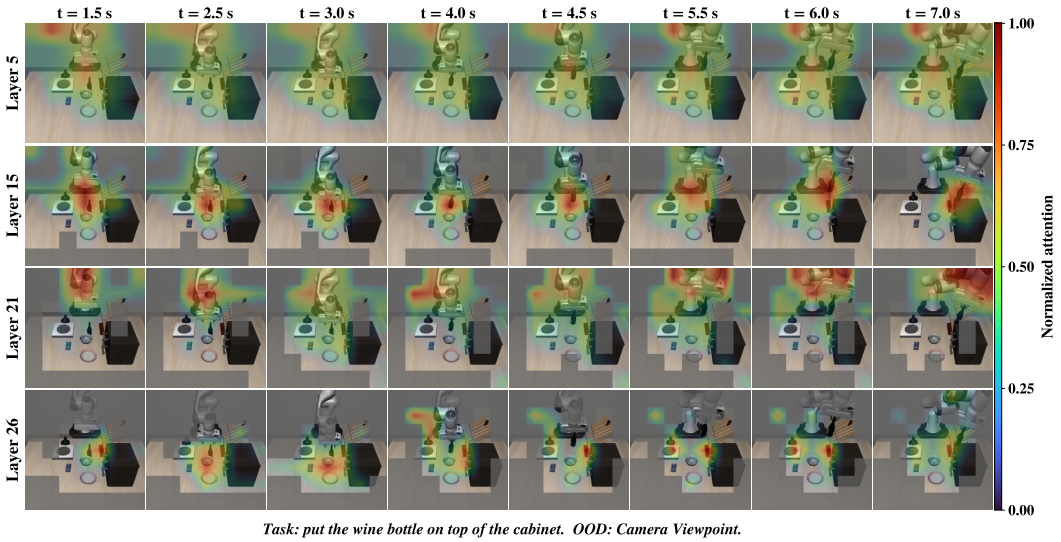


Figure 10: Attention under **camera-viewpoint** OOD perturbation (depth-only shift $z{=}123$) for the task *put the wine bottle on top of the cabinet*, where the manipulated object is on the *right* side of the scene. The attended region follows the wine bottle and the cabinet top throughout, providing a counter-example to a hypothesized left-positional dataset bias: when the manipulation is on the right, the sparse router attends to the right.