CONGR-WAM: DIAGNOSING LOCAL CONFLICTS FOR SPARSE GRADIENT ROUTING IN WORLD ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Joint World Action Models learn visual dynamics and robot control together, yet global gradient alignment can conceal local conflicts between the two objectives. An analysis of video K/V parameters reveals a stable spatial pattern of conflict across training checkpoints and task suites, while conflict within a given subspace varies across optimizer steps. ConGR-WAM uses this structure to guide sparse gradient routing, separating where to intervene from when to intervene. Offline diagnosis constructs a conflict topology over block-pair subspaces with equal parameter budgets, yielding a sparse intervention subspace selected from gradient evidence through a tail-robust maximin criterion. During training, symmetric projection is applied only when the video and action gradients conflict within the selected subspace. Routing preserves the original gradient sum elsewhere and on nonconflicting steps. Controlled ablations show that routing in the diagnosed subspace outperforms random sparse and dense conditional alternatives, and that conditional activation improves over unconditional projection. On LIBERO and RoboTwin 2.0, ConGR-WAM achieves state-of-the-art performance among comparable World Action Models across single-arm and bimanual manipulation settings. Our project page is available at `https://congr-wam-review.pages.dev/`.

1 INTRODUCTION

World Action Models (WAMs) couple visual prediction with action learning, using predictive supervision to inform robot control (Kim et al., 2026; Ma et al., 2026a; Yuan et al., 2026). This joint formulation is attractive because predictive learning can expose future dynamics that are useful for control. For manipulation, anticipating how a scene may evolve and deciding how to act are closely related, but they are not the same learning problem. A shared representation must accommodate both objectives, and information that helps reconstruct a future observation need not contribute equally to the next control decision. The benefit of joint learning therefore depends not only on access to predictive information, but also on how the two training signals interact.

However, sharing predictive information does not guarantee compatible optimization. The video and action objectives meet inside shared parameters, and different parameter regions need not serve them in the same way. A global gradient inner product sums contributions across those regions, so positive contributions in one region can outweigh negative ones elsewhere. Global compatibility can therefore conceal local interference, while broad intervention can alter gradients that are already aligned. An aggregate decision alone does not establish which parameter regions warrant correction. This motivates examining the spatial distribution of conflict rather than treating the shared video expert as a single intervention unit. The resulting question is where intervention is justified and when it is needed.

Gradient analysis of a joint WAM on robotic manipulation tasks reveals this mismatch within video key and value (K/V) parameters. Among diagnostic records with nonnegative cosine after aggregation over all diagnostic blocks, 15.4% remain conflicting within the selected block pair subspace. Equal budget comparisons reveal a nonuniform conflict topology with strong ranking consistency across checkpoints and task suites. This suggests that interference has a repeatable spatial structure

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.

rather than being uniformly distributed across the shared video expert. Yet subspaces with strong conflict evidence are not conflicting at every optimizer step, so a location that is important in aggregate may still be aligned at a particular update. In Figure 1, spatial structure therefore guides where to intervene, while current gradients determine when. The two sources of variation suggest that location and activation should be inferred from different evidence.

These observations motivate ConGR-WAM, which connects the observed conflict structure to a sparse intervention subspace selected from gradient evidence. Evidence across checkpoints and task suites identifies where correction is warranted, while current local gradients determine when it is activated. Selection seeks recurring conflict evidence across diagnostic conditions rather than a location favored by one checkpoint or task suite. It does not use rollout success to choose the intervention location. Once selected, the subspace retains its membership during training, but its gradients are checked anew at each optimizer step. This distinguishes a persistent intervention location from a permanently active correction. The goal is not to replace ordinary joint learning with continual conflict correction. Instead, routing targets a recurring source of interference and leaves the original joint gradient unchanged whenever the objectives agree. This separation also makes intervention location, scope, and timing directly testable through controlled comparisons.

With diagnosis and selection performed independently on LIBERO (Liu et al., 2023) and RoboTwin 2.0 (Chen et al., 2026), ConGR-WAM achieves state-of-the-art performance among comparable World Action Models. Controlled ablations test intervention location, scope, and timing, while selector analyses examine the roles of checkpoint, suite, and tail robustness. Together, these experiments test whether useful gradient intervention should be localized in parameter space and selective over training time. The main contributions are as follows.

- We identify structured local gradient conflicts hidden by global aggregation in joint WAM training and establish their consistency across checkpoints and task suites.
- We introduce ConGR-WAM, which combines a sparse intervention subspace selected from gradient evidence with conflict-triggered symmetric projection.
- We demonstrate through benchmark evaluations and controlled analyses that diagnostic selection, intervention scope, and stepwise activation improve joint WAM optimization without changing the deployment model.

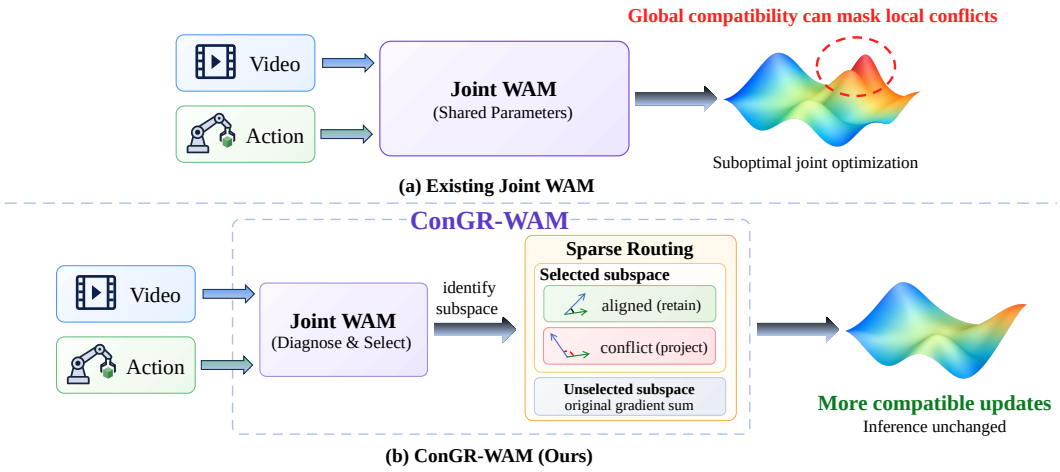

Figure 1: Global compatibility can conceal local conflict. (a) Joint video and action optimization may appear globally compatible despite local interference. (b) ConGR-WAM diagnoses a sparse subspace and applies symmetric projection only on conflicting steps, preserving other gradients before clipping and optimizer updates. Inference remains unchanged.

2 Related Work

World Action Models. Predictive control builds on visual foresight (Finn & Levine, 2017; Ebert et al., 2018) and latent imagination (Ha & Schmidhuber, 2018; Hafner et al., 2020), extending to

physical robot learning, scalable control, and predictive representations (Wu et al., 2023; Hansen et al., 2024; Assran et al., 2025). Video grounding, compositional prediction, and predictive features further support action learning (Du et al., 2023; Ko et al., 2024; Luo & Du, 2025; Zhou et al., 2024; Hu et al., 2025). Action diffusion and diverse robot data support generalist policies (Chi et al., 2025; Open X-Embodiment Collaboration, 2024; Khazatsky et al., 2024; Octo Model Team et al., 2024). RT-2 and OpenVLA transfer vision and language knowledge to control (Brohan et al., 2023; Kim et al., 2025b), while π_0 and $\pi_{0.5}$ use flow matching for generalist action modeling (Black et al., 2025b;a). WAMs combine future and action prediction (Wu et al., 2024; Cheang et al., 2024; Li et al., 2025; Zhu et al., 2025), drawing on pretrained video models and specialized action decoders (Shen et al., 2025; Ye et al., 2026; Feng et al., 2025; Pai et al., 2026). Motus and LingBot-VA couple modality-specific experts (Bi et al., 2026; Li et al., 2026). Cosmos Policy integrates actions into video diffusion (Kim et al., 2026), whereas DiT4DiT conditions action generation on intermediate video representations (Ma et al., 2026a). Fast-WAM retains video supervision without iterative future generation at inference (Yuan et al., 2026). Sparse forward interactions (Zhao et al., 2026) and RIFT's anticipation tokens (Zhang et al., 2026a) offer alternative conditioning mechanisms. These models differ in how they use predictive information. ConGR-WAM focuses on how video and action objectives interact during training.

Joint Optimization For World Action Models. AttenA+ reweights action supervision using velocity information (Peng et al., 2026), and PFD distills future-conditioned corrections into an action adapter (Fang et al., 2026). Related training refinements improve action adaptation (Kim et al., 2025a) or separate forward and inverse dynamics pretraining (Zhang et al., 2026b). More generally, joint learning regulates objective interactions through loss weighting and gradient normalization (Kendall et al., 2018; Chen et al., 2018), multiobjective optimization (Sener & Koltun, 2018; Navon et al., 2022), and gradient modification (Yu et al., 2020; Liu et al., 2021). Conflict evidence also guides architectural sharing (Shi et al., 2023), parameter freezing (Zhang et al., 2024), and gradient combination within parameter buckets (Hoang et al., 2026). Our work complements these approaches by studying where gradient intervention should be applied in joint WAM training, using local conflict evidence to select a sparse parameter subspace.

3 Problem Formulation

A World Action Model jointly learns future visual prediction and action generation from multiview observations, proprioception, and language. Following Fast-WAM (Yuan et al., 2026), video and action experts use flow matching losses over future visual latents and actions, respectively. The action expert attends to first frame video features, allowing both objectives to affect video parameters. For trainable parameters θ, fixed objective weights $\lambda_v, \lambda_a > 0$, and a parameter coordinate subset S, the joint objective and local task gradients are

$$\mathcal{L}(\theta) = \lambda_v \mathcal{L}_v(\theta) + \lambda_a \mathcal{L}_a(\theta), \qquad g^S_{m,t} = \lambda_m \nabla_{\theta_S} \mathcal{L}_{m,t}, \quad m \in \{v, a\}. \tag{1}$$

Here, $\mathcal{L}_{m,t}$ denotes the loss averaged across ranks and accumulation microbatches at optimizer step t. Local gradient interaction is measured by

$$\delta_t(S) = \langle g^S_{v,t}, g^S_{a,t} \rangle. \tag{2}$$

The objectives conflict within S when $\delta_t(S) < 0$. The problem is to identify a sparse intervention subspace $S^\star$ from diagnostic gradients alone and modify its joint gradient only on conflicting steps, preserving the original gradient sum elsewhere and on nonconflicting steps.

4 Conflict Diagnosis and Sparse Gradient Routing

Overview. In Figure 2, ConGR-WAM separates where to intervene from when to intervene through three stages. Offline diagnosis measures local video–action gradient conflict across equal-budget block-pair K/V subspaces. Robust selection aggregates evidence across checkpoints and task suites using a tail-robust maximin criterion to select one sparse subspace. During training, symmetric projection is applied only when the current gradients conflict within the selected subspace, preserving the original gradient sum elsewhere and on nonconflicting steps. The architecture, parameter count, and inference procedure remain unchanged.

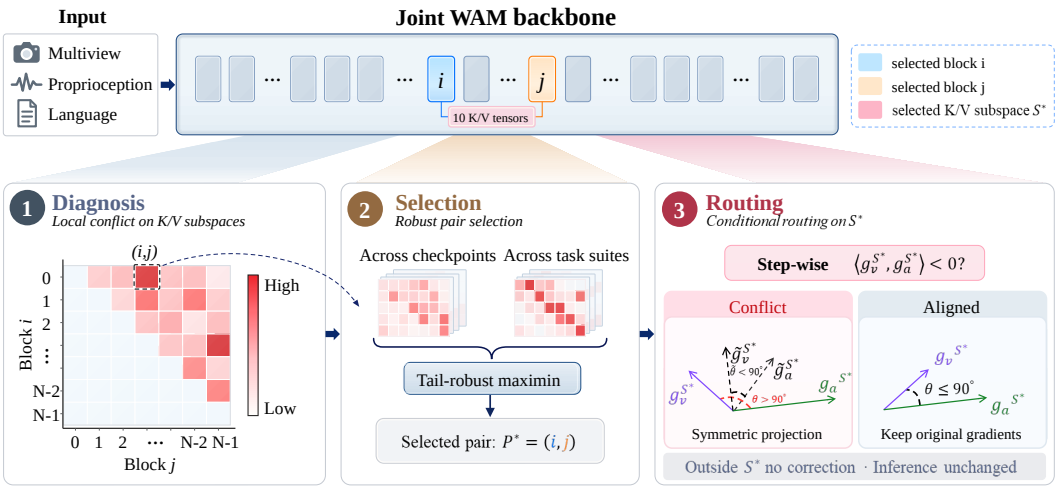

Figure 2: ConGR-WAM decouples where to intervene from when to intervene. (1) Diagnosis measures local video–action gradient conflict across equal-budget block-pair K/V subspaces. (2) Selection aggregates checkpoint and task-suite evidence with a tail-robust maximin rule to select one sparse subspace. (3) During training, its ten tensors form one routing group. Conflicting task gradients are symmetrically projected, while the original joint gradient is retained elsewhere and on nonconflicting steps before global clipping and AdamW. The trainable parameter set and inference procedure remain unchanged.

Offline Conflict Diagnosis. For video Transformer block i, let Θ_i^{KV} collect the parameter coordinates of its key and value projection weights and biases, together with the key normalization scale. Each unordered pair $p = (i, j)$ in $\mathcal{P} = \{(i, j) : 0 \le i < j < N\}$ defines a candidate subspace $S_p = \Theta_i^{KV} \cup \Theta_j^{KV}$. The $N = 30$ blocks yield 435 candidates with identical parameter budgets, each covering ten tensors, five per block. At each diagnostic cell $c = (k, u)$, comprising checkpoint k and data group u, the two weighted losses are differentiated using the same batch and stochastic forward pass, without updating parameters. Let $g_{m,c,b}^p$ denote the resulting gradient restricted to S_p on batch b. For valid diagnostic batches, the conflict score is

$$r_{p,c,b} = \left[-\frac{\langle g_{v,c,b}^p, g_{a,c,b}^p \rangle}{\|g_{v,c,b}^p\|_2 \|g_{a,c,b}^p\|_2} \right]_+, \qquad [x]_+ = \max(x, 0). \tag{3}$$

Both task gradients are evaluated on the same pair subspace. Pair inner products and squared norms are reconstructed by summing blockwise statistics. These scores define a conflict topology over video blocks, with candidate pairs as edges.

Robust Subspace Selection. Let $\mathcal{B}_c$ denote the common valid batch set in cell c, and $q_{p,c}^{0.95}$ the 95th percentile of its scores for candidate p. To limit the influence of isolated batches, scores are capped before averaging.

$$s_{p,c} = \frac{1}{|\mathcal{B}_c|} \sum_{b \in \mathcal{B}_c} \min\{r_{p,c,b}, q_{p,c}^{0.95}\}. \tag{4}$$

Zero-conflict batches remain in the average. Eligible candidates require sufficient valid batches and positive finite cell scores. The largest batch's share of total raw conflict must not exceed 0.15 in any cell or 0.10 in median across cells. These conditions define $\mathcal{P}_{\text{eligible}}$. Tail-robust maximin selection then gives

$$p^\star = \arg\max_{p \in \mathcal{P}_{\text{eligible}}} \min_u \operatorname{median}_k s_{p,(k,u)}, \qquad S^\star = S_{p^\star}. \tag{5}$$

Checkpoint medians capture persistent evidence, while the group minimum favors subspaces whose conflict remains pronounced even in the least conflicting group. Ties favor the higher median group score, then the lexicographically smaller pair. Selection uses only gradient evidence, without rollout success or downstream validation performance, and is performed independently for each benchmark. On LIBERO, four checkpoints, four task suites, and 122 common valid batches per suite select the

zero-indexed pair $(0, 8)$. The selected subspace remains unchanged during training, while projection is activated according to the current local gradient conflict.

Conditional Sparse Routing. At each optimizer step, task gradients are averaged across ranks, accumulated in FP32, and averaged across accumulation microbatches before routing. All ten selected tensors form one group for computing inner products and norms. Let g_t denote the original joint gradient over all trainable parameters, abbreviate $g_m = g_{m,t}^{S^\star}$, and set $\delta_t = \delta_t(S^\star)$. Let $\iota_{S^\star}$ embed a local vector into the full parameter space, assigning zeros elsewhere. Conditional symmetric projection (Yu et al., 2020) can then be expressed as a sparse correction.

$$\widetilde{g}_t = g_t + [-\delta_t]_+ \, \iota_{S^\star} \left(\frac{g_v}{\|g_v\|_2^2 + \varepsilon} + \frac{g_a}{\|g_a\|_2^2 + \varepsilon} \right), \qquad \varepsilon = 10^{-12}. \tag{6}$$

On conflicting steps, the selected entries equal the sum of the two symmetric projections, both computed from the original task gradients. The correction is supported only on $S^\star$ and vanishes when $\delta_t \geq 0$, preserving the original gradient sum elsewhere and on nonconflicting steps. The full routed gradient is then globally clipped and passed to AdamW. Gradient preservation applies before these transformations. ConGR-WAM retains the original trainable parameter set, model architecture, and inference procedure.

5 Experiments

5.1 Experimental Setup

Evaluation. ConGR-WAM is evaluated on LIBERO (Liu et al., 2023) and RoboTwin 2.0 (Chen et al., 2026) using task success rate. LIBERO covers Spatial, Object, Goal, and Long, with 10 tasks per suite and 50 evaluation trials per task, yielding 2,000 evaluation outcomes. RoboTwin covers 50 bimanual tasks under clean and domain-randomized settings with unseen language instructions, using 100 episodes per task and setting, for 10,000 episodes in total.

Implementation. ConGR-WAM retains the Fast-WAM architecture and inference procedure. Diagnosis and subspace selection are performed independently for each benchmark. LIBERO routing variants use 10 training epochs and AdamW. Both benchmarks use 10 inference steps. LIBERO disables action ensembling, and RoboTwin uses a replanning interval of 24 steps.

5.2 Main Results on LIBERO and RoboTwin

In Tables 1 and 2, ConGR-WAM is compared with representative VLA and World Action Model methods. In Table 4, routing location, scope, and timing are isolated under a common training and evaluation protocol described in Sections 5.5 and 5.6. Reported gains denote absolute differences in success rates.

Table 1: Results on LIBERO. Success rates are reported in percent. PT denotes robot pretraining before benchmark adaptation. Shading identifies ConGR-WAM, and bold metric values indicate the best result in each metric column.

Method	PT	Spatial ↑	Object ↑	Goal ↑	Long ↑	Average ↑
OpenVLA (Kim et al., 2025b)	Yes	84.7	88.4	79.2	53.7	76.5
π_0 (Black et al., 2025b)	Yes	96.8	98.8	95.8	85.2	94.1
$\pi_{0.5}$ (Black et al., 2025a)	Yes	98.8	98.2	98.0	92.4	96.9
LingBot-VA (Li et al., 2026)	Yes	98.5	99.6	97.2	**98.5**	98.5
Motus (Bi et al., 2026)	Yes	96.8	99.8	96.6	97.6	97.7
VITA (Ma et al., 2026b)	Yes	95.9	98.9	95.1	96.8	96.7
PFD (Fang et al., 2026)	No	98.6	99.2	99.2	95.4	98.1
Fast-WAM (baseline) (Yuan et al., 2026)	No	98.2	**100.0**	97.0	95.2	97.6
ConGR-WAM (Ours)	No	**100.0**	99.4	**100.0**	95.8	**98.8**

LIBERO. ConGR-WAM achieves 98.8% average success, the highest among the compared methods, exceeding the baseline by 1.2%. It improves over the baseline in three of the four suites, reaching 100.0% success on Spatial and Goal. Architecture, parameter count, and inference remain unchanged.

Table 2: Results on RoboTwin. Success rates are reported in percent. PT denotes robot pretraining before benchmark adaptation. Average is the mean of unrounded Clean and Rand. rates, reported to one decimal place. Shading identifies ConGR-WAM, and bold metric values indicate the best result in each metric column.

Method	PT	Clean ↑	Rand. ↑	Average ↑
π_0 (Black et al., 2025b)	Yes	65.9	58.4	62.2
$\pi_{0.5}$ (Black et al., 2025a)	Yes	82.7	76.8	79.8
Motus (Bi et al., 2026)	Yes	88.7	87.0	87.8
Motus from WAN2.2 (Bi et al., 2026)	No	77.6	77.0	77.3
LingBot-VA (Li et al., 2026)	Yes	92.9	91.5	92.2
X-WAM (Guo et al., 2026)	Yes	89.8	90.7	90.3
AttenA+WAM (Peng et al., 2026)	No	93.1	91.9	92.5
Fast-WAM (baseline) (Yuan et al., 2026)	No	91.9	91.8	91.8
ConGR-WAM (Ours)	No	**93.2**	**92.7**	**93.0**

RoboTwin 2.0. With independently diagnosed subspace selection, ConGR-WAM achieves 93.2% clean and 92.7% randomized success, exceeding the baseline by 1.3% and 0.9%, respectively. It leads both settings among the compared methods, with 93.0% average success and a gain of 1.2% over the baseline, extending the evidence to bimanual manipulation.

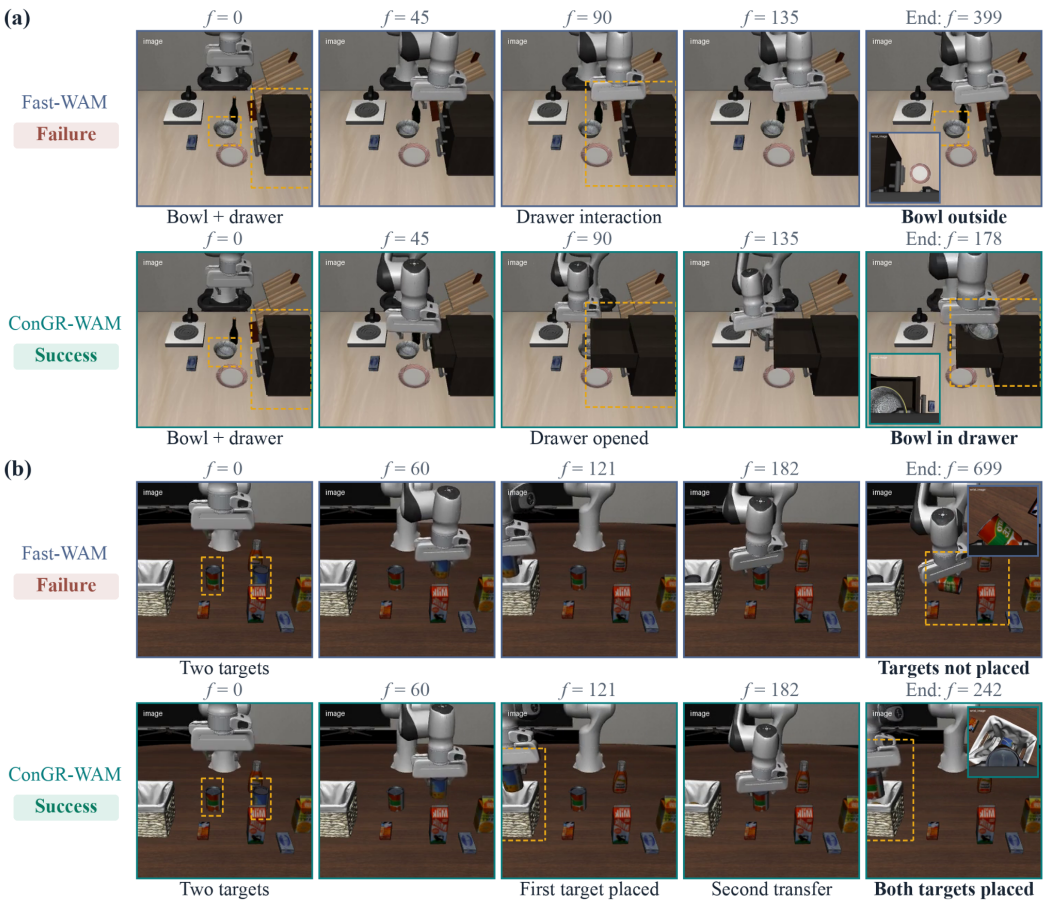

Figure 3: Paired LIBERO executions. (a) ConGR-WAM opens the drawer and places the target bowl inside, whereas the baseline leaves it on the table in Goal task 3, trial 20. (b) ConGR-WAM transfers both target objects to the basket, whereas the baseline fails to place them in Long task 0, trial 26. Dashed boxes highlight targets and interactions, and wrist-view insets show the terminal states. Columns share frame indices f, except End, which shows each execution's own terminal frame.

Qualitative Results. In Figures 3 and 4, paired executions show how ConGR-WAM and the baseline perform on LIBERO and RoboTwin 2.0. ConGR-WAM completes drawer placement and two-object transfer on LIBERO, and microphone handover and three-block stacking on RoboTwin, whereas the displayed baseline executions fail to complete the instructions.

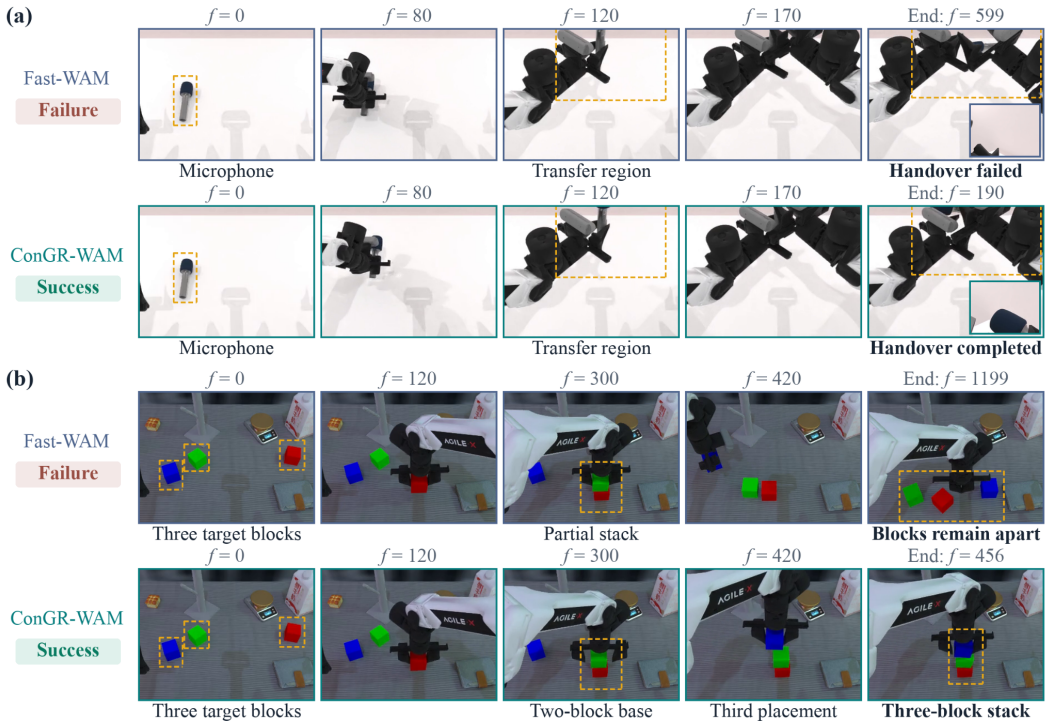

Figure 4: Paired RoboTwin 2.0 executions with matched scene seeds. (a) ConGR-WAM completes clean microphone handover between grippers, whereas the baseline fails. (b) Under domain randomization, ConGR-WAM completes the three-block stack, whereas the baseline ends with separated blocks after a partial stack. Dashed boxes mark targets and interactions, and insets show terminal wrist views. Columns share frame indices f, except for each execution's End frame.

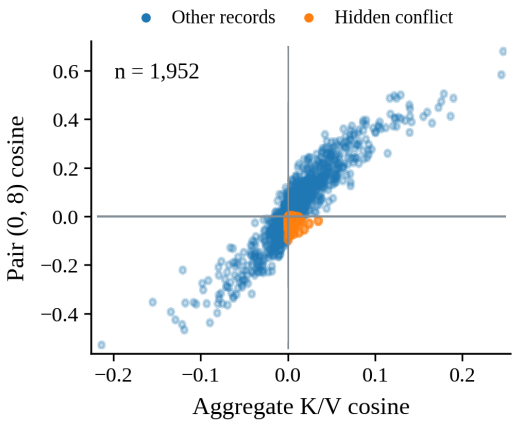

Figure 5: Global alignment can hide local conflict. Each point compares video–action gradient cosine over all diagnostic K/V parameters and within pair (0, 8), using 1,952 LIBERO records. Orange points mark local conflict in 181 of 1,178 globally aligned records (15.4%).

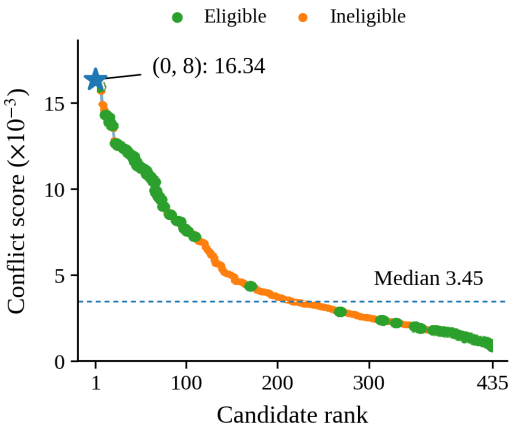

Figure 6: Conflict varies across equal-budget locations. The 435 pairs are ranked by worst-suite score after checkpoint-median aggregation of P95-capped conflict. Colors show eligibility, the star marks pair (0, 8), and the dashed line marks the candidate median.

5.3 Localized Conflict Topology Discovery

In Figure 5, 15.4% of globally aligned records still conflict within pair (0, 8), showing that global aggregation can miss local interference. In Figure 6, the 435 block pairs have a nonuniform score spectrum despite identical tensor types and parameter budgets. Pair (0, 8) scores 16.3×10^{-3}, approximately 4.7 times the candidate median. These equal-budget differences isolate intervention location from routed capacity.

In Figure 7, pair rankings are consistent across checkpoint-by-suite conditions, with a mean Spearman correlation of 0.916. This repeatable structure supports cross-condition selection without implying conflict at every step. Section 5.4 tests the selection components, while Section 5.6 examines stepwise activation.

5.4 Robustness of Conflict Subspace Selection

In Table 3, selector variants are evaluated on the same diagnostic gradients without rollout evaluation. LOCO/LOSO reselection and bootstrap ranks characterize selection stability under each rule. Appendix A.2 defines the variants and resampling protocol.

Table 3: Selector robustness on LIBERO diagnostic gradients. Score is the full-rule aggregation score of the selected pair ($\times 10^{-3}$). Tail indicates eligibility under the full rule. LOCO and LOSO count recovery of each variant's full-data selection across four leave-one-checkpoint-out and leave-one-suite-out refits, respectively. Median rank summarizes 2,000 batch-bootstrap resamples under the corresponding variant.

Selector variant	Pair	Score ↑	Tail	LOCO ↑	LOSO ↑	Median rank ↓
Tail-robust maximin	(0, 8)	**16.3**	✓	**4/4**	**4/4**	**1.0**
Pooled checkpoints	(8, 9)	14.2	✓	2/4	3/4	4.0
w/o suite maximin	(8, 9)	14.2	✓	2/4	2/4	3.0
w/o tail robustness	(7, 8)	16.1	✗	2/4	**4/4**	**1.0**

The full rule selects (0, 8) and recovers it in all LOCO and LOSO refits, with a bootstrap median rank of 1.0. Pooling checkpoints or removing suite maximin selects the weaker-scoring pair (8, 9). Without tail robustness, pair (7, 8) also has median rank 1.0 but fails the complete tail gate. The same pair reaches 96.7% as R1 in Table 4, compared with 98.8% for (0, 8). Thus, ranking stability alone does not ensure admissible tail behavior. The full rule combines both with strong worst-suite evidence.

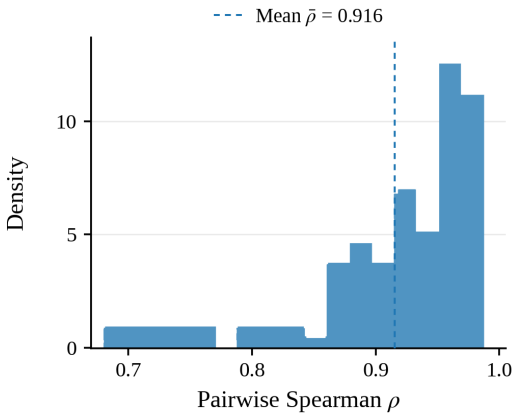

Figure 7: Conflict-ranking consistency across diagnostic conditions. The histogram gives 120 pairwise Spearman correlations between the 435-pair rankings across 16 LIBERO checkpoint-by-suite conditions. The dashed line marks the mean, $\bar{\rho} = 0.916$.

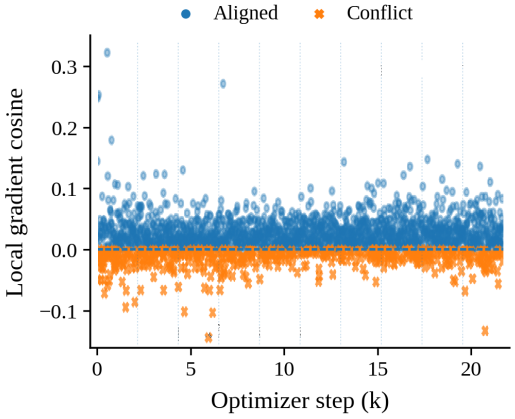

Figure 8: Local conflict is intermittent during training. Pair (0, 8) has negative video–action cosine at 498 of 2,170 logged steps (22.95%), triggering routing. Values are logged every ten updates before projection, and vertical guides mark epoch boundaries.

Table 4: Spatial and temporal routing controls on LIBERO. Success is the mean task success rate (%), and Params counts routed parameters only. Sparse variants route one block pair, whereas dense routing groups all diagnostic K/V parameters. † denotes a point estimate without an estimated interval.

Routing variant	Scope	Activation	Params (M)	Success ↑	Gain [95% interval]
Random sparse R1	(7, 8)	On conflict	37.8	96.7	+2.1 [1.4, 3.1]
Random sparse R2	(9, 19)	On conflict	37.8	97.4	+1.4 [0.8, 2.2]
Dense conditional	30 blocks	On conflict	566.5	97.6	+1.2 [0.4, 1.9]
Unconditional projection	(0, 8)	Every step	37.8	96.9	+1.9†
ConGR-WAM (Ours)	(0, 8)	On conflict	37.8	**98.8**	Reference

5.5 Spatial Selectivity of Gradient Routing

In Table 4, all variants retain unit loss coefficients and the original diffusion timestep weights. ConGR-WAM exceeds random pairs (7, 8) and (9, 19) by 2.1% and 1.4%, with both paired intervals above zero. Equal parameter types and budgets support diagnostic location selection rather than sparsity alone. Routing-group definitions, gradient handling, and the paired-interval protocol are in Appendices B.2 and C.1.

Dense conditional routing uses a 15-fold larger routed parameter budget, yet ConGR-WAM improves success by 1.2%, with a paired interval of [0.4%, 1.9%]. This favors the tested localized intervention over a single group spanning all diagnostic K/V parameters. In Figure 9, the spatial controls share the same task initialization for a qualitative comparison, as described in Appendix C.2.

5.6 Temporal Selectivity and Computational Cost

In Figure 8, most logged steps retain the original local gradient sum before clipping and AdamW. The activation rates describe only the sampled subset, not the full 21,700-step trajectory.

Why Conditional Activation Matters. The temporal control retains pair (0, 8), its ten tensors, and symmetric projection but activates at every step, as shown in Table 4. Conditional activation yields a point-estimate gain of 1.9%, raising success from 96.9% to 98.8% and supporting stepwise gating within the selected subspace.

Computational Cost. In Table 5, model loading and diagnostic gradient collection on existing checkpoints incur a one-time wall-clock cost of 27.7 minutes on four GPUs. Selection then runs on the CPU. Under matched timing conditions, the baseline and ConGR-WAM take 7.3s and 8.3s per optimizer step, corresponding to 13.7% overhead. Routing adds only training cost. Both methods retain 6.0B policy-core parameters and identical inference operators, with batch runtimes of 863.3ms and 865.6ms, respectively.

Table 5: Computational cost. Inference runtime is measured per batch. Parameter counts exclude the frozen VAE. Timing details are in Appendix B.3.

Metric	Baseline	ConGR-WAM (Ours)
Diagnosis cost (min)	Not required	27.7
Training step time (s) ↓	7.3	8.3
Policy-core parameters (B)	6.0	6.0
Inference batch runtime (ms) ↓	863.3	865.6

6 Conclusion

In this work, we identify structured local gradient conflicts that global aggregation can conceal in joint World Action Model training. We introduce ConGR-WAM to separate where to intervene from when to intervene. Offline diagnosis and tail-robust maximin selection identify a sparse subspace, while symmetric projection is activated only by current local conflict. Our method preserves the original joint gradient elsewhere and on nonconflicting steps. On LIBERO and RoboTwin 2.0, we achieve state-of-the-art performance among comparable World Action Models without changing the architecture, parameter count, or inference procedure. Controlled ablations show that both diagnosed location and conditional activation matter.

Reproducibility Statement

Sections 3 and 4 specify the diagnostic score, subspace selection, and gradient routing rule. Section 5 details benchmark evaluation, ablations, and computational cost. Appendices A–C provide algorithmic details, diagnostic summaries, evaluation protocols, and implementation settings. The project page is available at `https://congr-wam-review.pages.dev/`.

AI Use Statement

No generative AI tools were used in conducting this research or preparing this manuscript.

References

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 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, 2026.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: A vision-language-action model with open-world generalization. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40, 2025a.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. In *Proceedings of Robotics: Science and Systems*, 2025b.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183, 2023.

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.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan-ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 794–803, 2018.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion Policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10–11):1684–1704, 2025.

Yilun Du, Mengjiao Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B. Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 9156–9172, 2023.

Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv preprint arXiv:1812.00568*, 2018.

Pengcheng Fang, Hongli Chen, and Xiaohao Cai. Privileged Foresight Distillation: Zero-cost future correction for World Action Models. *arXiv preprint arXiv:2604.25859v1*, 2026.

Yao Feng, Hengkai Tan, Xinyi Mao, Chendong Xiang, Guodong Liu, Shuhe Huang, Hang Su, and Jun Zhu. Vidar: Embodied video diffusion model for generalist manipulation. *arXiv preprint arXiv:2507.12898*, 2025.

Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2786–2793, 2017.

Jun Guo, Qiwei Li, Peiyan Li, Zilong Chen, Nan Sun, Yifei Su, Heyun Wang, Yuan Zhang, Xinghang Li, and Huaping Liu. Unified 4D World Action Modeling from Video Priors with Asynchronous Denoising. *arXiv preprint arXiv:2604.26694*, 2026.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, 2024.

Long P. Hoang, Yiran Zhao, Wei Lu, and Wenxuan Zhang. Multilingual fine-tuning via localized gradient conflict resolution. *arXiv preprint arXiv:2606.05613*, 2026.

Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video Prediction Policy: A generalist robot policy with predictive visual representations. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 24328–24346, 2025.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7482–7491, 2018.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Abraham Lee, Youngwoon Lee, Arhan Jain, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Hatch, Shan Lin, Jingpei Lu, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Walke, Blake Wulfe, Ted Xiao, Jonathan Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Donovon Jackson, Charlotte Le, Yunshuang Li, Kevin Lin, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abby O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Andrew Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J. Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. DROID: A large-scale in-the-wild robot manipulation dataset. In *Proceedings of Robotics: Science and Systems*, 2024.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. In *Proceedings of Robotics: Science and Systems*, 2025a.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713, 2025b.

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. In *International Conference on Learning Representations*, 2026.

Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B. Tenenbaum. Learning to act from actionless videos through dense correspondences. In *International Conference on Learning Representations*, 2024.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Luyao Zhang, Mingrui Yu, Zelin Gao, Nan Xue, Boyu Zhou, Xing Zhu, Mingyu Ding, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. In *Proceedings of Robotics: Science and Systems*, 2026.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified Video Action Model. In *Proceedings of Robotics: Science and Systems*, 2025.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 18878–18890, 2021.

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*, volume 36, pp. 44776–44791, 2023.

Yunhao Luo and Yilun Du. Grounding video models to actions through goal conditioned exploration. In *International Conference on Learning Representations*, 2025.

Teli Ma, Jia Zheng, Zifan Wang, Chunli Jiang, Andy Cui, Junwei Liang, and Shuo Yang. DiT4DiT: Jointly modeling video dynamics and actions for generalizable robot control. *arXiv preprint arXiv:2603.10448*, 2026a.

Xiangkai Ma, Lekai Xing, Han Zhang, Wenzhong Li, and Sanglu Lu. Unifying Perception and Action: A Hybrid-Modality Pipeline with Implicit Visual Chain-of-Thought for Robotic Action Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22380–22390, 2026b.

Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. Multi-task learning as a bargaining game. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16428–16446, 2022.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, 2024.

Open X-Embodiment Collaboration. Open X-Embodiment: Robotic learning datasets and RT-X models. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 6892–6903, 2024.

Jonas Pai, Liam Achenbach, Oliver Sanchez, Stefanos Charalambous, Victoriano Montesinos, Benedek Forrai, Oier Mees, and Elvis Nava. mimic-video: Video-action models for generalizable robot control beyond VLAs. In *Proceedings of Robotics: Science and Systems*, 2026.

Daojie Peng, Fulong Ma, Jiahang Cao, Qiang Zhang, Xupeng Xie, Jian Guo, Ping Luo, Andrew F. Luo, Boyu Zhou, and Jun Ma. AttenA+: Rectifying action inequality in robotic foundation models. *arXiv preprint arXiv:2605.13548v1*, 2026.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

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. In *Advances in Neural Information Processing Systems*, volume 38, pp. 95597–95621, 2025.

Guangyuan Shi, Qimai Li, Wenlong Zhang, Jiaxin Chen, and Xiao-Ming Wu. Recon: Reducing conflicting gradients from the root for multi-task learning. In *International Conference on Learning Representations*, 2023.

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. In *International Conference on Learning Representations*, 2024.

Philipp Wu, Alejandro Escontrela, Danijar Hafner, Ken Goldberg, and Pieter Abbeel. DayDreamer: World models for physical robot learning. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 2226–2240, 2023.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi Jim Fan, and Joel Jang. World Action Models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5824–5836, 2020.

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.

Chushan Zhang, Jinguang Tong, Xuesong Li, Yikai Wang, and Hongdong Li. Keep the future, drop the rollout: RIFT for World Action Models. *arXiv preprint arXiv:2608.11521v1*, 2026a.

Wenyao Zhang, Bozhou Zhang, Zekun Qi, Wenjun Zeng, Xin Jin, and Li Zhang. Disentangled robot learning via separate forward and inverse dynamics pretraining. In *International Conference on Learning Representations*, 2026b.

Zhi Zhang, Jiayi Shen, Congfeng Cao, Gaole Dai, Shiji Zhou, Qizhe Zhang, Shanghang Zhang, and Ekaterina Shutova. Proactive gradient conflict mitigation in multi-task learning: A sparse training perspective. *arXiv preprint arXiv:2411.18615*, 2024.

Weiheng Zhao, Haoyi Jiang, Xin Shi, Liu Liu, Fan Huang, Zhizhong Su, Wei Sui, and Xinggang Wang. Faster-WAM: Efficient inference-time future conditioning for robust World Action Models. *arXiv preprint arXiv:2608.04404*, 2026.

Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. RoboDreamer: Learning compositional world models for robot imagination. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 61885–61896, 2024.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. In *Proceedings of Robotics: Science and Systems*, 2025.

A Diagnostic and Selection Details

Diagnostic workflow. LIBERO diagnosis uses the released checkpoint and the epoch-1, epoch-5, and epoch-10 checkpoints of a reference joint-training trajectory. After offline selection, LIBERO routing variants are trained from a common initialization under the settings in Table 9. The selected subspace remains unchanged during routed training. In Table 7, the benchmark-specific diagnostic configurations are summarized.

A.1 Pair Reconstruction and Robust Selection

For block i, the scan stores the inner product $d_i = \langle g_v^i, g_a^i \rangle$ and squared norms $n_{m,i} = \|g_m^i\|_2^2$, where each gradient is restricted to that block's K/V coordinates. For disjoint blocks i, j, the pair statistics are

$$\delta(i,j) = d_i + d_j, \qquad \|g_m^{(i,j)}\|_2^2 = n_{m,i} + n_{m,j}, \quad m \in \{v, a\}. \tag{A1}$$

Thus, all 435 equal-budget candidates are evaluated from the same blockwise records, without a separate backward pass for each pair.

A valid pair record has finite statistics and both squared norms at least 10^{-12}. Every candidate–cell combination in the LIBERO diagnostic set contains 122 valid batches. The P95 cap in Equation (4) uses linear interpolation between sorted scores. Zero-conflict batches remain in the mean. Tail eligibility requires the largest batch's share of uncapped total conflict to be at most 0.15 in every cell and at most 0.10 in median across cells. Cell scores must also be positive and finite.

Table 6: LIBERO diagnostic scores for the selected pair $(0, 8)$, in units of 10^{-3}. Each cell is the P95-capped mean over 122 common valid batches. The final column takes the checkpoint median, and the bold minimum is the score reported in Table 3. Aggregation uses unrounded scores.

Suite	Release	Epoch 1	Epoch 5	Epoch 10	Median
Spatial	35.3	5.4	22.8	46.6	29.1
Object	33.9	12.6	24.7	33.7	29.2
Goal	18.4	12.0	14.3	25.1	**16.3**
Long	36.0	12.0	22.9	27.8	25.4

In Table 6, taking checkpoint medians and then the minimum over suites for $(0, 8)$ gives 16.3×10^{-3}. Eligible candidates are ordered by this maximin score, then by the median suite score, and finally lexicographically. Rollout outcomes do not enter subspace selection.

A.2 Sensitivity Analysis Protocol

The pooled-checkpoint variant pools batch scores across checkpoints before applying the P95 cap. Removing suite maximin replaces the suite minimum with a mean. Removing tail robustness removes both P95 capping and the tail eligibility gates. In Table 3, each selected pair is rescored

using the full rule. Leave-one-checkpoint-out and leave-one-suite-out analyses each make four refits, measuring recovery of that variant's full-data selection. Bootstrap median rank tracks the same fixed selection over 2,000 batch resamples under the corresponding rule, with within-suite batch indices shared across checkpoints.

Table 7: Diagnostic configurations and selected subspaces. LIBERO cells are checkpoint–suite combinations. For RoboTwin, shuffled training episodes are assigned round-robin to four disjoint partitions. Selection uses the first 122 valid batches per partition. Pair indices are zero-based.

Benchmark	**Diagnostic checkpoints**	**Data groups**	**Valid batches**	**Pair**
LIBERO	Release + epochs 1, 5, 10	Spatial, Object, Goal, Long	122/cell (1,952 total)	$(0, 8)$
RoboTwin 2.0	Official Fast-WAM release	4 disjoint partitions	122/partition (488 total)	$(9, 11)$

B Implementation and Computational Cost

B.1 Training and Inference Settings

The backbone has 30 video Transformer layers of width 3072 and 30 action Transformer layers of width 1024. Video frames are encoded by the VAE, and current proprioception is encoded into the conditioning context. In Table 8, inputs and evaluation settings are compared across benchmarks. In Table 9, the LIBERO optimization settings are specified.

Table 8: Benchmark inputs and evaluation settings. Image sizes are height $\times$ width after view composition. LIBERO per-task evaluation results are reported in Appendix C.1.

Setting	**LIBERO**	**RoboTwin 2.0**
Camera views	External and wrist	Head and two wrists
Model image size	224×448	384×320
View composition	Two views side by side	Head above wrist views
Action / proprioception dimensions	7 / 8	14 / 14
Tasks	40 across four suites	50 in each setting
Evaluation settings	Spatial, Object, Goal, Long	Clean and randomized
Trials per task and setting	50	100
Inference steps / replanning interval	10 / 10	10 / 24

Table 9: LIBERO training settings. The effective batch size is $4 \times 16 \times 2 = 128$. Optimizer ϵ is distinct from the routing stabilizer in Equation (6).

Setting	**Value**
Optimizer	AdamW with $\beta_1 = 0.9$, $\beta_2 = 9.5 \times 10^{-1}$, $\epsilon = 10^{-8}$
Learning rate / weight decay	1.0×10^{-4} / 1.0×10^{-2}
Schedule / warmup	Cosine / 1,085 updates (5.0%)
Training length	10 epochs (21,700 optimizer updates)
Batching	4 GPUs, 16 samples per GPU, 2 accumulation steps
Video / action loss coefficients	1.0 / 1.0
Global gradient clipping	Maximum norm 1.0
Precision / distributed optimizer	BF16 / ZeRO stage 1
Activation checkpointing	Enabled
Checkpoint interval	2,170 optimizer updates

The reported LIBERO ConGR-WAM result uses the epoch-10 checkpoint. LIBERO training samples span 33 observation steps, with video sampled every four steps. LIBERO disables action ensembling. RoboTwin evaluates clean and randomized scenes with unseen language instructions.

B.2 Routed Parameters and Gradient Handling

Each selected video block contributes $W_K, W_V \in \mathbb{R}^{3072\times3072}$ and $b_K, b_V, \gamma_K \in \mathbb{R}^{3072}$, where γ_K is the key normalization scale. This gives $2(3072)^2 + 3(3072) = 18{,}883{,}584$ parameters per block. Two blocks form one routing group of ten tensors and 37,767,168 parameters. The dense control groups 150 tensors and 566,507,520 parameters. Neither variant projects each tensor independently.

Algorithm 1 ConGR-WAM diagnostic selection and conditional sparse routing

Offline diagnosis and selection

1 Differentiate the weighted video and action losses on the same diagnostic batch and stochastic forward pass without updating parameters.

2 Reconstruct all equal-budget block-pair statistics using Equation (A1) and compute conflict scores with Equation (3).

3 Apply P95 capping and the eligibility gates, then select $S^\star$ with Equation (5).

At each optimizer step t

4 Compute joint gradients, average task gradients across ranks in FP32, then accumulate and average across microbatches.

5 Form $g_v = g_{v,t}^{S^\star}$, $g_a = g_{a,t}^{S^\star}$, and $\delta = \langle g_v, g_a \rangle$ over the entire selected group.

6 **If** $\delta < 0$, compute both projections from the original g_v, g_a and sum them, as in Equation (6).

7 **Otherwise**, use the unprojected sum $g_v + g_a$ on $S^\star$.

8 Keep the ordinary joint gradient on all remaining coordinates.

9 Globally clip the resulting gradient and perform the AdamW update.

Algorithm 1 summarizes the complete update order. The subspace is selected independently for each benchmark and retained throughout training. Projection uses the original task gradients and $\varepsilon = 10^{-12}$. Gradient preservation applies before global clipping and AdamW. Inference uses no routing controller.

B.3 Timing Protocol

Timing measurements use NVIDIA A800 80GB PCIe GPUs. Training-step timing follows the batching and distributed configuration in Table 9, with the same precision and activation-checkpointing settings for both methods. The reported 7.3s and 8.3s are means over repeated measurements. Inference batch runtimes are measured under matched model and input configurations, with no routing operations at inference.

The one-time diagnostic cost of 27.7 minutes measures wall-clock time on four GPUs for model loading and gradient collection on existing checkpoints. Checkpoint training is excluded. Selection is subsequently performed on the CPU and is not included in the reported diagnostic time.

C Supplementary Evaluation Details

C.1 LIBERO Task Results and Paired Uncertainty

In Table 10, all 40 ConGR-WAM task results are reported using 50 evaluation trials per task. Each suite averages its ten task rates, and the mean across four suites is 98.8%. The three spatial intervals in Table 4 use 10,000 suite-stratified paired task bootstrap resamples, each drawing ten tasks with replacement per suite and sharing the task indices across methods. Interval endpoints are the 2.5th and 97.5th percentiles of ConGR-WAM minus control. The unconditional-projection comparison is a point estimate.

Table 10: Per-task success rates of ConGR-WAM (Ours) on LIBERO at epoch 10, with 50 evaluation trials per task. Success (%) equals successful trials divided by 50 and multiplied by 100. Task IDs are zero-indexed within each suite, and LIBERO-Long corresponds to LIBERO-10. Evaluation covers 40 tasks and 2,000 trials with action ensembling disabled.

ID	Task	Success (%)	Successes
LIBERO-Spatial			
0	Pick up the black bowl between the plate and the ramekin and place it on the plate	100	50/50
1	Pick up the black bowl next to the ramekin and place it on the plate	100	50/50
2	Pick up the black bowl from table center and place it on the plate	100	50/50
3	Pick up the black bowl on the cookie box and place it on the plate	100	50/50
4	Pick up the black bowl in the top drawer of the wooden cabinet and place it on the plate	100	50/50
5	Pick up the black bowl on the ramekin and place it on the plate	100	50/50
6	Pick up the black bowl next to the cookie box and place it on the plate	100	50/50
7	Pick up the black bowl on the stove and place it on the plate	100	50/50
8	Pick up the black bowl next to the plate and place it on the plate	100	50/50
9	Pick up the black bowl on the wooden cabinet and place it on the plate	100	50/50
	Suite average	**100.0**	500/500
LIBERO-Object			
0	Pick up the alphabet soup and place it in the basket	100	50/50
1	Pick up the cream cheese and place it in the basket	98	49/50
2	Pick up the salad dressing and place it in the basket	100	50/50
3	Pick up the bbq sauce and place it in the basket	98	49/50
4	Pick up the ketchup and place it in the basket	100	50/50
5	Pick up the tomato sauce and place it in the basket	100	50/50
6	Pick up the butter and place it in the basket	100	50/50
7	Pick up the milk and place it in the basket	98	49/50
8	Pick up the chocolate pudding and place it in the basket	100	50/50
9	Pick up the orange juice and place it in the basket	100	50/50
	Suite average	**99.4**	497/500
LIBERO-Goal			
0	Open the middle drawer of the cabinet	100	50/50
1	Put the bowl on the stove	100	50/50
2	Put the wine bottle on top of the cabinet	100	50/50
3	Open the top drawer and put the bowl inside	100	50/50
4	Put the bowl on top of the cabinet	100	50/50
5	Push the plate to the front of the stove	100	50/50
6	Put the cream cheese in the bowl	100	50/50
7	Turn on the stove	100	50/50
8	Put the bowl on the plate	100	50/50
9	Put the wine bottle on the rack	100	50/50
	Suite average	**100.0**	500/500
LIBERO-Long			
0	Put both the alphabet soup and the tomato sauce in the basket	98	49/50
1	Put both the cream cheese box and the butter in the basket	100	50/50
2	Turn on the stove and put the moka pot on it	98	49/50
3	Put the black bowl in the bottom drawer of the cabinet and close it	100	50/50
4	Put the white mug on the left plate and put the yellow and white mug on the right plate	86	43/50
5	Pick up the book and place it in the back compartment of the caddy	100	50/50
6	Put the white mug on the plate and put the chocolate pudding to the right of the plate	86	43/50
7	Put both the alphabet soup and the cream cheese box in the basket	96	48/50
8	Put both moka pots on the stove	96	48/50
9	Put the yellow and white mug in the microwave and close it	98	49/50
	Suite average	**95.8**	479/500
Overall average		**98.8**	**1,976/2,000**

C.2 Qualitative Spatial Routing Comparison

In Figure 9, qualitative examples complement the routing location and scope comparisons in Section 5.5. All four executions share the task initialization. ConGR-WAM retrieves the specified bowl from the top drawer and places it on the plate. In the displayed dense and random-sparse executions, the target remains in the drawer while another bowl is manipulated. This comparison illustrates the task-level behavior underlying the aggregate routing controls.

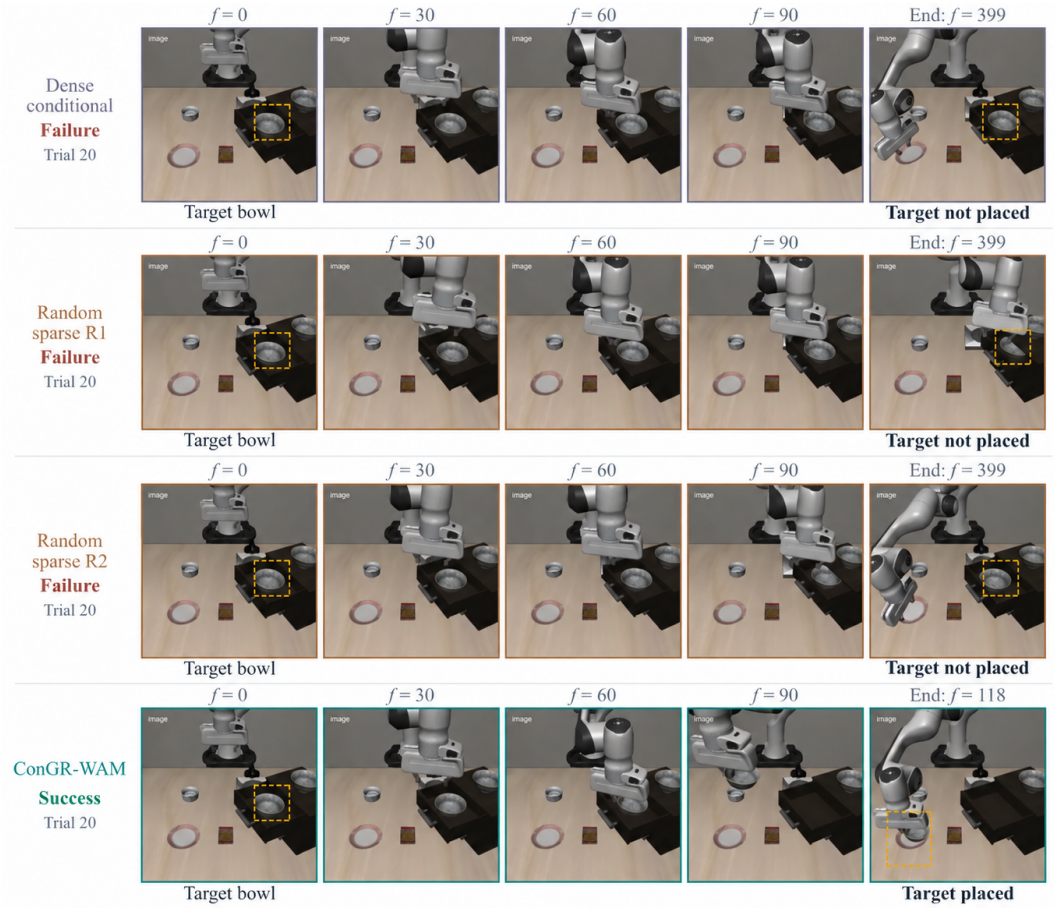

Figure 9: Spatial routing comparison on LIBERO Spatial, task 4, trial 20 at epoch 10. Dense conditional routing groups all 30 blocks. Random sparse R1 and the displayed random sparse R2 route $(7, 8)$ and $(9, 19)$, respectively, while ConGR-WAM routes $(0, 8)$. Gold boxes identify the instructed bowl. Columns share frame indices f, except End, which shows each execution's terminal frame. The displayed controls leave the target in the drawer, whereas ConGR-WAM places it on the plate.

C.3 RoboTwin Qualitative Example Metadata

The clean microphone-handover pair in Figure 4(a) shares scene seed 4300084, with baseline episode 65 and ConGR-WAM episode 61. The randomized three-block-stacking pair in Figure 4(b) shares scene seed 4300006, with episode 4 for both. Matching is by task, setting, and scene seed, so episode indices need not coincide. Terminal frames are taken at each execution's own termination.