FEDWAM: FEDERATING PREDICTIVE WORLD KNOWLEDGE FOR PERSONALIZED ROBOT CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

World-action models (WAMs) can benefit from experience collected across sites, but data ownership and heterogeneous model architectures complicate joint training. We introduce FedWAM, a framework that shares predictive world knowledge while retaining local demonstrations and private action experts. Clients upload conditional chunk-end predictions in a common visual feature space. A client-balanced server teacher integrates these contributions and is broadcast to all participants. Each recipient calibrates teacher supervision against local future observations, distills it into its native world pathway, and adapts its action expert to the updated context. Repeated exchange combines shared predictive learning with personalized control. The interface accommodates both LaWAM's future features and ImageWAM's internal control representations. In the reported comparisons, FedWAM reaches 78.4% LIBERO-Plus success versus 76.3% for local training, with most gains outside the recipient's training domain, while reaching 97.9% on original LIBERO. Mixed-architecture comparisons improve both recipient families, with direction-dependent donor transfer; three Piper tasks provide a complementary physical comparison. These findings motivate assessing shared predictors through both feature accuracy and the behavior of locally adapted controllers. Project page: `https://submit-anonymous.github.io/FedWAM`.

1 INTRODUCTION

World-action models (WAMs) bring future prediction and action modeling together for embodied control (Wu et al., 2024; Lyu et al., 2026). By learning relationships among observations, actions, and future outcomes, they aim to use knowledge of how environments evolve to support robot behavior. The interactions and visual conditions represented in training help determine where this predictive knowledge is useful. Experience collected across sites could broaden that coverage: robots encounter related tasks with different objects, viewpoints, and scene appearances. Recent work has shown that heterogeneous embodied experience can benefit joint dynamics and action learning (Lyu et al., 2026). These developments motivate learning WAMs from multiple sources to support control beyond the conditions represented in any one local dataset.

Access to these sources cannot always be taken for granted. A robot demonstration records both an interaction and the setting in which it occurs. In manufacturing, data ownership and intellectual-property concerns can impede data sharing (Hegiste et al., 2022). Robot recordings make these concerns concrete: they can capture workstation layouts and production sequences. The learning opportunity therefore comes with a practical constraint: WAMs should be able to benefit from experience collected elsewhere while the contributing sites retain their raw records.

Federated learning coordinates training while retaining local data. Parameter-space approaches exchange model parameters or updates; FedAvg is a representative algorithm that combines local training with weighted parameter averaging (McMahan et al., 2017). Sharing selected world components could preserve local action experts, following the broader principle of selective sharing (Miao et al., 2025). Direct aggregation, however, requires corresponding parameter coordinates in those components, which different WAM architectures may not provide. Since future modeling already contributes to control, predictions offer a candidate interface for collaboration. Distillation supports exchange through predictions across architectures (Li & Wang, 2019; Lin et al., 2020); applying it

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.

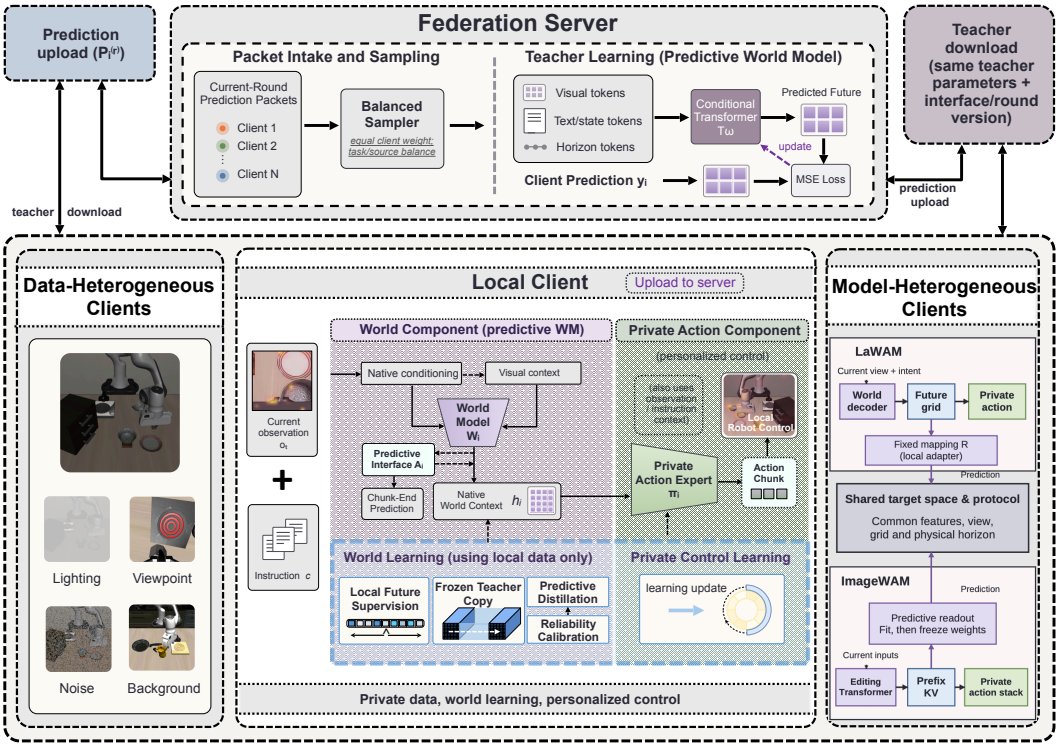


Figure 1: FedWAM overview. Clients upload conditional future predictions, receive a shared teacher, and adapt their world pathways and private controllers locally.

to WAMs requires shared predictive semantics and a path for received supervision into each client's native control representation.

We introduce **FedWAM**, a framework for *federating predictive world knowledge with personalized local control* (Figure 1). The shared knowledge is a task-conditioned relation between current inputs and visual features at the end of an action chunk. Clients express local predictions in a shared frozen encoder's coordinates and upload them with observation features, task conditions, and horizons. A client-balanced server teacher integrates these packets and is broadcast to all participants. Raw demonstrations and native WAM parameters remain local.

Each recipient calibrates distillation by comparing teacher and local prediction errors against locally observed futures. Teacher supervision updates the native world pathway, while the private action expert adapts through local demonstrations using the evolving context. Repeated exchange lets clients contribute their updated predictions. We instantiate this procedure with LaWAM and ImageWAM, using predictive accuracy to guide reception and closed-loop evaluation to assess control utility.

The reported evaluation uses LIBERO and four LIBERO-Plus visual domains, mixed LaWAM/ImageWAM populations, and three Piper tasks. It examines where recipients gain useful coverage and how predictive improvements enter control. Our contributions are:

1. **A predictive interface for heterogeneous WAMs.** Conditional chunk-end features provide common sharing semantics while preserving native world representations and private action experts.
2. **A complete predictive federation procedure.** Client-balanced teacher learning, locally calibrated absorption, and post-reception action adaptation connect decentralized predictions to personalized control.

3. **An analysis of transfer and control utility.** The comparisons separate own- and other-environment gains, mixed-pool and directional architecture transfer, and predictive accuracy from action performance.

2 Related Work

Predictive representations for control. UniPi converts language-conditioned video plans into actions (Du et al., 2023); GR-1 jointly predicts future images and actions (Wu et al., 2024). Compact alternatives include LDA-1B's future DINO features (Lyu et al., 2026) and LaWAM's latent visual subgoals (Chen et al., 2026). ImageWAM conditions actions on image-editing key–value features (Zhang et al., 2026). FedWAM connects these different control pathways through a common future-feature target.

Federated robot learning. FedVLA shares a trunk with private stems and heads (Miao et al., 2025); federated VLA fine-tuning studies sharing scopes under shift (Duan et al., 2026). F-DRL aggregates dynamics encoders for local controllers (Upadhyay et al., 2026), while FedWorld exchanges transition rules for language-model agents (Hou, 2026). FedWAM exchanges conditional visual predictions for absorption into native WAMs.

Personalization and heterogeneous distillation. FedRep shares representations with private heads (Collins et al., 2021); FedProto exchanges class prototypes across heterogeneous models (Tan et al., 2022). FedMD uses predictive consensus on public inputs (Li & Wang, 2019), FedDF distills client ensembles (Lin et al., 2020), and FedGen learns a latent generator from client predictors (Zhu et al., 2021). FedWAM applies predictive exchange to temporal visual features and couples locally calibrated reception with private control adaptation.

3 Problem Formulation

We study N clients with private demonstration datasets $\mathcal{D}_i$ drawn from local distributions P_i. Clients share an embodiment but may differ in visual conditions, task coverage, and WAM architecture. Each model conditions a private action expert on native world features and induces a policy $\pi_i \in \Pi_i$. Raw demonstrations and action experts remain local; trained clients execute independently.

A demonstration window contains current inputs $x_t = (o_t, \ell, s_t)$, an action chunk $a_{t:t+H_i-1}$, and endpoint observation o_{t+H_i}. Observations may include multiple views; ℓ is an instruction and s_t optional robot state. The physical horizon is $\tau = H_i\Delta_i$, with action period Δ_i. It describes an action-chunk endpoint within an ongoing task, including when the controller replans after a shorter prefix. *Predictive world knowledge* is the task-conditioned relation between current inputs and observations at this horizon under local execution.

For an evaluation distribution ν over tasks, environments, and initial states, we seek to improve personalized closed-loop success relative to matched local training:

$$\mathcal{J}(\pi_1, \ldots, \pi_N) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{\mathcal{E}\sim\nu}[\mathrm{Success}(\pi_i, \mathcal{E})]. \tag{1}$$

4 FedWAM

FedWAM alternates local WAM learning, conditional teacher fitting, and calibrated reception (Figure 1). Predictions provide the common interface; each client retains its native control architecture.

4.1 Conditional predictive interface

A frozen visual encoder E and spatial transform R define current and future features at the shared horizon:

$$c_t = R(E(o_t^{\mathrm{ref}})), \qquad y_t^* = R(E(o_{t+H_i}^{\mathrm{ref}})) \in \mathbb{R}^{P\times d}. \tag{2}$$

Here P tokens have dimension d, with a fixed reference-view convention and preprocessing. Client i maps its control-connected world features h_i through a native adapter A_i:

$$F_i(x) = A_i(h_i(x)), \qquad q(x) = (c_t, e(\ell), \widetilde{s}_t, m_t, \tau). \tag{3}$$

The conditions contain a fixed instruction encoding, normalized state and availability mask, and the horizon. Both F_i and q use deployment-available inputs; observed futures supervise local learning.

After local updates in round r, client i freezes $F_i^{r,-}$ and uploads training-derived condition–prediction pairs:

$$\mathcal{P}_i^r = \{(q(x_{im}), \mathrm{sg}(F_i^{r,-}(x_{im})))\}_{m=1}^{B_i^r}, \quad x_{im} \in \mathcal{D}_i^{\mathrm{tr}}. \tag{4}$$

Here sg stops gradients. Sampling balances tasks and sources under a common payload budget. Raw images, action labels, observed future targets, and native parameters remain local. The primary target is the future feature; the alternative $F_i(x) - c_t$ is equivalent in information given c_t.

4.2 Client-balanced conditional teacher

The server fits a conditional Transformer T_ω to uploaded predictions $\widehat{y}$, with equal client mass and task/source balancing within clients:

$$\mathcal{L}_{\mathrm{srv}}^r = \frac{1}{N} \sum_i \mathbb{E}_{\mathcal{P}_i^{r,\mathrm{fit}},\mathrm{bal}}\, d(T_\omega(q), \widehat{y}), \qquad d(U,V) = \frac{\|U - V\|_F^2}{Pd}. \tag{5}$$

Spatial queries produce future tokens from the shared conditions. Each round continues the preceding teacher for a fixed update budget and broadcasts the same teacher to all clients. The squared-error optimum is the conditional mean of client predictions under this balanced mixture (Appendix I.3). Its local utility depends on source support and prediction quality, motivating recipient-specific calibration.

4.3 Calibrated absorption and private control adaptation

On a source-disjoint calibration set, each client compares its packet-generating snapshot with the teacher:

$$\begin{aligned} e_i^r &= \mathbb{E}_{\mathcal{D}_i^{\mathrm{cal}},\mathrm{bal}}\, d(F_i^{r,-}(x), y^*), \quad e_{G,i}^r = \mathbb{E}_{\mathcal{D}_i^{\mathrm{cal}},\mathrm{bal}}\, d(T_{\omega^r}(q(x)), y^*), \\ g_i^r &= \mathrm{clip}_{[0,1]}\big((e_i^r - e_{G,i}^r)/(e_i^r + \epsilon)\big), \qquad \epsilon > 0. \end{aligned} \tag{6}$$

A teacher with lower error receives greater weight; otherwise reception is suppressed. Holding the teacher and gate fixed, the client optimizes

$$\mathcal{L}_{\mathrm{fed},i}^r = \mathbb{E}_{\mathcal{D}_i^{\mathrm{tr}},\mathrm{bal}}\, d(F_i(x), \mathrm{sg}(T_{\omega^r}(q(x)))), \tag{7}$$

$$\mathcal{L}_i^r = \mathcal{L}_{\mathrm{native},i} + \lambda_{\mathrm{align}} \mathcal{L}_{\mathrm{align},i} + \lambda_{\mathrm{fed}} g_i^r \mathcal{L}_{\mathrm{fed},i}^r. \tag{8}$$

Alignment supervises F_i against y^* when native learning does not already do so. Distillation updates the world pathway with private conditioning detached and action parameters excluded. Native losses preserve the architecture's gradient boundaries. Post-reception action learning uses the updated world context, including after the final exchange, so the private controller adapts to the representation it will use at deployment.

4.4 Native interfaces and federation cycle

We use shared DINOv3 features (Siméoni et al., 2025). Figure 2 summarizes the native paths. **LaWAM** decodes predicted intent into future features that condition its action expert (Chen et al., 2026); A_i selects and transforms the reference-view tokens. Native future supervision already supplies alignment. Federated gradients update the decoder with intent detached; native action learning preserves its detached world-input boundary.

ImageWAM-FLUX.2-4B conditions actions on image-editing prefix features (Zhang et al., 2026). A predictive readout is first fitted to detached features, then fixed while retaining its input derivative. Alignment and distillation thus update the native world parameters upstream of the readout. A differentiable prefix pass matches native action-inference conditions, and local action learning uses the updated context.

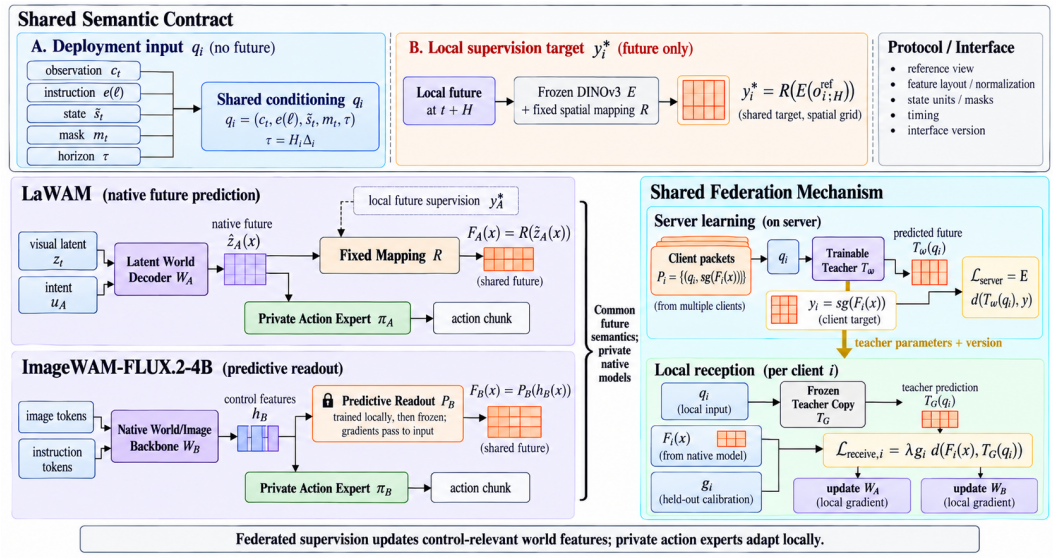


Figure 2: Native interfaces. LaWAM future tokens and ImageWAM control features map to a common target; action experts remain private.

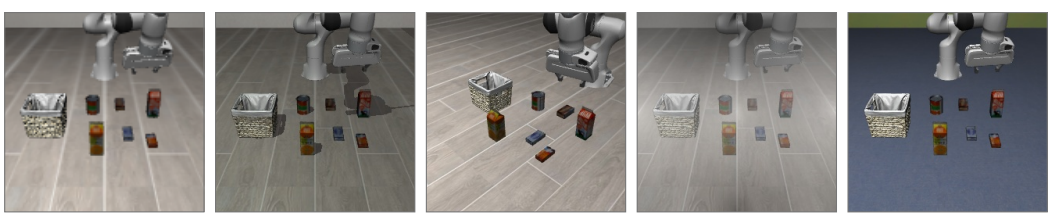
After K_0 initialization updates, each of R_{fed} rounds performs K_{pre} local updates, uploads packets, fits and broadcasts the teacher, then performs K_{post} calibrated local updates. Updated clients supply the next round's predictions. Each receives $K_0 + R_{\text{fed}}(K_{\text{pre}} + K_{\text{post}})$ native updates and deploys independently. Algorithm 1 and Appendix I give the schedule and gradient details.

5 Experiments

We examine control performance, cross-client transfer, and the connection between predictive learning and action. The comparisons use author-supplied point estimates; the arrays' source-provenance conflict and run-record coverage are documented in Appendix H.

5.1 Setting and comparisons

Four LaWAM clients train on lighting, viewpoint, noise, and background data. Each recipient is evaluated on all four LIBERO-Plus domains (Fei et al., 2026) and the 40 original LIBERO tasks across Spatial, Object, Goal, and Long (Liu et al., 2023). Figure 3 shows the environments. Source groups retain consistent train/calibration/development/test roles across variants. The reference schedule uses seed 43 and 25,000 native updates per client, including final action adaptation; executed configuration records are listed separately in Appendix B.



(a) Clean (b) Lighting (c) Viewpoint (d) Noise (e) Background

Figure 3: Simulation environments: original LIBERO and four LIBERO-Plus visual domains.

Local follows the same native objectives and interface without federation. FedAvg-World averages world parameters with private action experts; federated distillation uses public-query predictions. Self-distillation fits a teacher from the recipient's own packets. Centralized pools permitted training

data. Comparisons match local initialization, data, and native-update budgets within each architecture; server updates are accounted for separately. Appendix C defines the controls.

5.2 Control performance and cross-client transfer

Table 1 reports suite-macro LIBERO and domain-macro LIBERO-Plus success, with equal task and recipient weights. These benchmarks have separate evaluation manifests. The pretrained reference uses separately sourced original-LIBERO scores and audited frozen-model LIBERO-Plus runs (Appendix K).

Table 1: Success (%) on LIBERO and LIBERO-Plus, with separate benchmark averages. †: pretrained reference; provenance in Appendix K.

	LIBERO					LIBERO-Plus				
Method	Spatial	Object	Goal	Long	Avg.	Light	View	Noise	Bkg.	Avg.
Pretrained†	98.6	99.2	97.8	96.2	98.0	94.8	39.5	71.6	95.4	75.3
Local	98.1	98.7	96.8	95.2	97.2	94.1	44.1	73.0	93.9	76.3
FedAvg-World	98.4	98.7	97.0	95.3	97.3	94.4	45.7	73.8	94.1	77.0
Federated distillation	97.8	98.5	96.6	94.7	96.9	93.8	44.3	73.5	93.6	76.3
Self-distillation	98.4	98.7	96.8	95.4	97.3	94.8	45.8	74.7	94.5	77.5
FedWAM	98.9	99.1	97.5	95.9	97.9	94.5	48.3	76.3	94.4	78.4
Centralized	99.6	99.7	98.9	97.7	99.0	96.5	60.1	83.0	96.5	84.0

Gains concentrate on challenging visual conditions. FedWAM reaches 78.4% LIBERO-Plus success versus 76.3% for Local and 77.0% for FedAvg-World. Viewpoint and noise contribute the largest gains, at 4.1 and 3.3 pp; lighting and background already start near 94%. Original-LIBERO performance rises from 97.2% to 97.9% relative to Local. Self-distillation reaches 77.5%, indicating that teacher fitting and reception account for part of the improvement. FedWAM's further 0.9 pp gain is consistent with additional value from cross-client predictions. The centralized score of 84.0% provides a reference for the remaining gap.

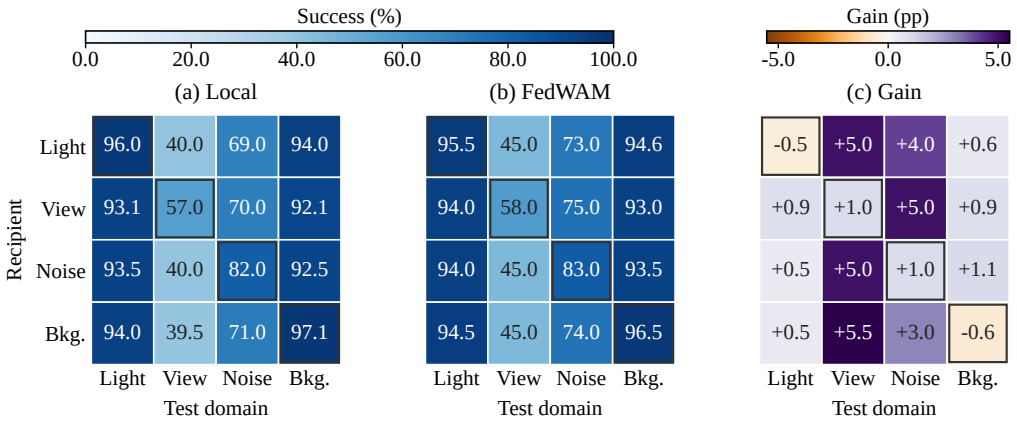


Figure 4: Simulation transfer: Local, FedWAM, and their difference (pp). Rows are clients; columns are test domains. Outlined cells mark each client's training domain.

The full matrix in Figure 4 identifies where the gain occurs. FedWAM improves all twelve other-domain cells; their mean rises from 74.1% to 76.7%, accounting for about 97% of the net gain before rounding. Own-domain success is 83.3% versus 83.0% for Local, with small decreases in two diagonal cells. This pattern associates sharing with broader coverage.

Table 2: Transfer success (%) and gain (pp).

Method	Own	Other	Δ	Pos.	Neg.
Local	83.0	74.1	N/A	N/A	N/A
FedAvg-World	82.2	75.3	0.7	4.6	3.9
Fed. distill.	81.7	74.5	0.0	2.4	2.4
Self-distill.	84.2	75.2	1.2	4.8	3.7
FedWAM	83.3	76.7	2.1	4.9	2.9

Table 2 also reports case-level positive and negative transfer, using all paired cases as the denominator. FedWAM's negative-transfer rate is 2.9%, compared with self-distillation's 3.7%, while positive-transfer rates are similar. Other-domain transfer concerns participating environments; per-recipient results appear in Appendix E.

5.3 Sharing across architectures

The mixed population assigns LaWAM and ImageWAM to disjoint source shares in each visual domain. Table 3 separates full-pool performance from donor-only comparisons that match recipient initialization, source coverage, packet format, and adaptation budget. The latter exclude each recipient's own predictions and prevent feedback to donors.

Table 3: Architecture transfer. Success (%) and gain over matched Local (pp).

		LaWAM		ImageWAM	
Study	Condition	Success	Δ	Success	Δ
Mixed pool	Local (native)	76.0	0.0	77.7	0.2
	Local	76.0	N/A	77.5	N/A
	Self-distillation	76.3	0.3	77.7	0.2
	Same-architecture sharing	77.2	1.2	78.3	0.8
	Federated distillation	75.8	-0.2	77.1	-0.4
	FedWAM w/o calibration	76.6	0.6	77.6	0.1
	FedWAM	77.5	1.5	78.2	0.7
Matched donors	Local	76.0	N/A	77.5	N/A
	Same-architecture sharing	76.9	0.9	78.1	0.6
	Cross-architecture sharing	76.5	0.5	77.3	-0.2

FedWAM improves mixed-pool LaWAM and ImageWAM by 1.5 and 0.7 pp over their respective Local baselines. Calibration adds 0.9 and 0.6 pp relative to ungated reception. Directional donor transfer is asymmetric: ImageWAM donors improve LaWAM by 0.5 pp, while LaWAM donors change ImageWAM by -0.2 pp; same-architecture donors perform better in both directions. The shared interface therefore enables exchange whose utility depends on the recipient. LaWAM's native target matches the shared features directly, whereas ImageWAM absorbs supervision through a fixed readout. Differences in readout fidelity, donor quality, and absorption offer possible explanations for the asymmetry.

5.4 Prediction and control during learning

Table 4: Prediction MSE ($\times 10^{-3}$).

Predictor	All ↓	High ↓
Copy current	52.0	116.0
Local	43.0	93.0
Self-distill.	42.5	92.0
Before reception	44.0	96.0
Teacher	41.0	90.0
After reception	40.0	87.0

Table 4 evaluates chunk-end feature MSE on prediction-test sources. Post-reception MSE is 40.0 versus Local's 43.0 in units of 10^{-3}. On high-change windows, it is 87.0 versus the copy-current baseline's 116.0, a 25% reduction. This comparison captures predictive change beyond static persistence. Local future supervision accompanies distillation, explaining how the student can improve on the teacher's 41.0 error. Matched Local and self-distillation endpoints separate these comparisons from the descriptive pre/post change, which includes further local updates. Native prediction visualizations and the future/transition equivalence appear in Appendices G.1 and F.

Figure 5 compares development trajectories at the same reported native-update checkpoints. FedAvg-World ends with lower MSE than FedWAM (0.0362 versus 0.0400), but lower success (77.3% versus 78.6%). FedWAM also has lower action loss. Feature-error rankings thus differ from control rankings: predictive changes must be useful to the private controller. FedWAM's gap over Local grows from 0.2 pp at 7,000 updates to 2.0 pp at 25,000. Complementary paired diagnostics report $+0.8$ pp from calibration and $+1.2$ pp from training actions on updated rather than cached context (Appendix F); the latter holds the world fixed and isolates how the controller uses its rep-

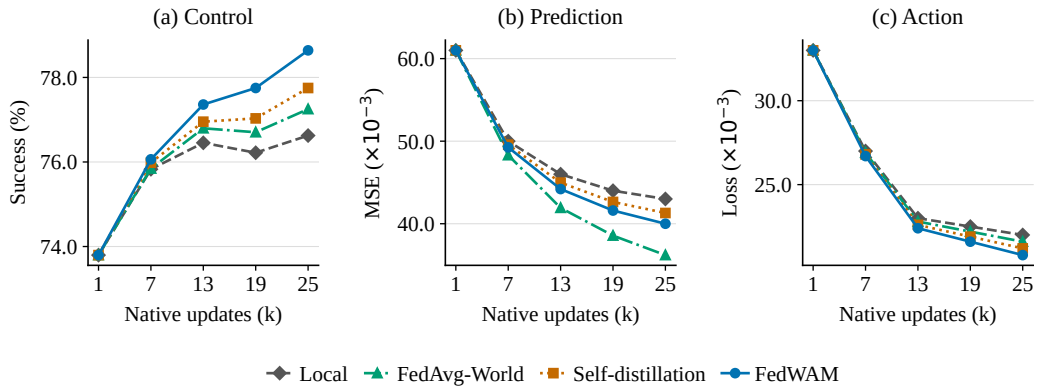


Figure 5: LaWAM development trajectories: control success, future-feature MSE, and action loss versus native updates. Lines join reported checkpoints; uncertainty intervals were not supplied.

resentation. Curves describe development, with no uncertainty intervals or test-based checkpoint selection.

5.5 Physical comparison

The Piper study uses RealSense D455 observations and four environments: clean, lighting, background, and viewpoint. Figure 6 illustrates the setup with existing platform recordings. Table 5 averages recipients for each task/environment; Figure 7 averages the same three tasks for each recipient/environment.

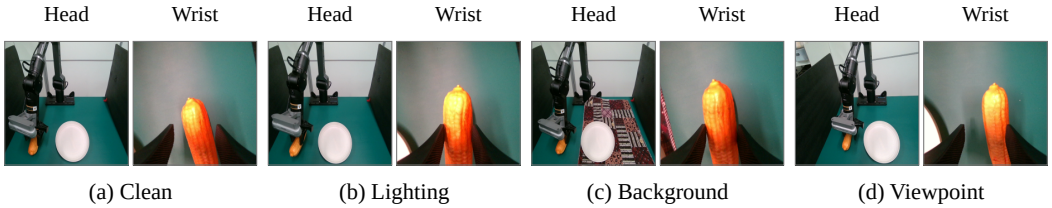


Figure 6: Piper scenes in four environments. Each pair shows head and wrist views from the same recorded observation; photographs illustrate the setup.

Table 5: Piper success (%) on three tasks and four environments, averaged over recipient clients. Avg. equally weights environments.

Task	Method	Clean	Light	Bkg.	View	Avg.
Corn to Plate	Local	81.3	72.5	72.5	60.0	71.6
	FedAvg-World	81.3	73.8	72.5	61.3	72.2
	Self-distillation	81.3	73.8	73.8	60.0	72.2
	FedWAM	85.0	77.5	77.5	66.3	76.6
Pen to Basket	Local	76.3	67.5	67.5	55.0	66.6
	FedAvg-World	76.3	68.8	67.5	56.3	67.2
	Self-distillation	76.3	68.8	68.8	55.0	67.2
	FedWAM	80.0	72.5	72.5	61.3	71.6
Cube to Bowl	Local	71.3	62.5	62.5	50.0	61.6
	FedAvg-World	71.3	63.8	62.5	51.3	62.2
	Self-distillation	71.3	63.8	63.8	50.0	62.2
	FedWAM	75.0	67.5	67.5	56.3	66.6

The reported overall success is 71.6% for FedWAM, 66.6% for Local, and 67.2% for both FedAvg-World and self-distillation. Each task improves by 5.0 pp over Local. As in simulation, the largest

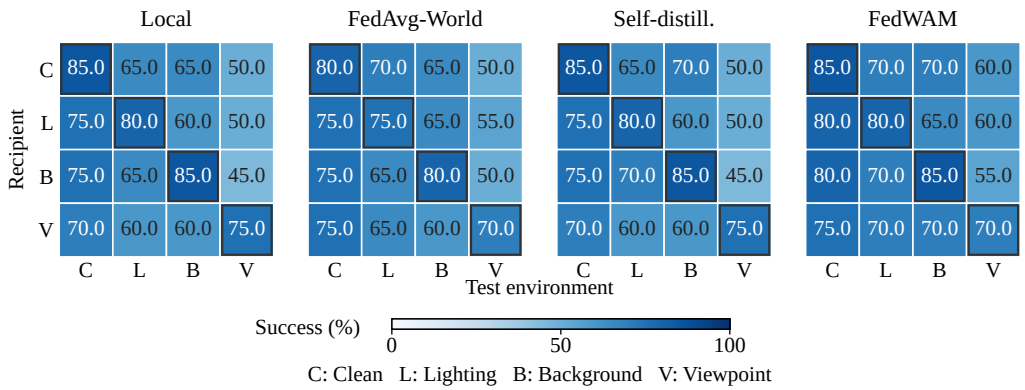


Figure 7: Physical cross-client success (%), averaged over three tasks. Rows are clients; columns are test environments. Outlined cells mark each client's training environment.

domain gain is in viewpoint (55.0% to 61.3%). The matrix shows a coverage–specialization trade-off: other-environment success rises from 61.7% to 68.8%, while own-environment success changes from 81.3% to 80.0%, including a 5 pp decrease for the viewpoint client. Trial-level records and uncertainty estimates remain to be linked to these three-task summaries (Appendix J).

6 Discussion

Predictive federation chiefly broadens visual coverage: gains concentrate outside each recipient's training domain, while average own-domain success remains close to Local. This pattern associates sharing with experience underrepresented in local demonstrations. The self-distillation comparison further separates cross-client contributions from teacher fitting. Evaluating recipients across participating environments therefore complements assessment of their local specialization. The full transfer matrix is useful here because a single aggregate score cannot show whether improved coverage is accompanied by changes in the recipient's own environment. Deployment can also weight environments by their expected frequency.

The common interface makes predictions comparable, while their usefulness depends on local absorption. LaWAM exposes future features directly; ImageWAM receives supervision through a native-feature readout. Directional transfer thus motivates recipient-specific calibration even with a common teacher. A shared target specifies what clients exchange without requiring their internal representations or optimization paths to coincide. Calibration measures predictive error, whereas deployment depends on the private controller. The MSE–success ranking mismatch reinforces this distinction. Post-reception action adaptation connects updated world features to behavior, including after the final exchange. Together, these mechanisms make knowledge reception and local control learning complementary parts of the federation cycle.

The present scope is task-conditioned visual prediction on a shared embodiment. Future features and residuals carry equivalent information given current features; conditioning and the path into control determine how this information is used. Endpoint features can omit intermediate events. Extending to new physics or embodiments requires compatible temporal and state semantics, and latent exchange alone provides no formal privacy guarantee. Appendix H records result provenance and remaining run-level evidence.

7 Conclusion

FedWAM connects heterogeneous WAMs through conditional future predictions, a shared teacher, and calibrated local absorption with private control adaptation. The reported comparisons associate this design with broader cross-client visual coverage and architecture-dependent transfer. They also motivate treating predictive accuracy and closed-loop utility as complementary criteria for learning from decentralized robot experience.

AI USE STATEMENT

Generative AI tools were used for writing and language polishing, literature retrieval and related-work discovery, research ideation and execution support, and drafting portions of the manuscript. The authors verified all cited sources, implementations, experimental protocols, numerical results, analyses, and scientific claims, and take full responsibility for the submission.

REFERENCES

Jialei Chen, Kai Wang, Kang Chen, Shuaihang Chen, Feng Gao, Wenhao Tang, Zhiyuan Li, Weilin Liu, Zhuyu Yao, Boxun Li, Yuanbo Xu, and Chao Yu. LaWAM: Latent world action models for efficient dynamics-aware robot policies. arXiv preprint arXiv:2606.15768, 2026. URL https://arxiv.org/abs/2606.15768.

Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 2089–2099. PMLR, 2021. URL https://proceedings.mlr.press/v139/collins21a.html.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh 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. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1d5b9233ad716a43be5c0d3023cb82d0-Abstract-Conference.html.

Zhekai Duan, Kevin Ziyang Xie, Xinyu Tan, Shikai Geng, Chengxu Zhou, Ramana Kompella, Gaowen Liu, and Chris Xiaoxuan Lu. An empirical study and open testbed for federated fine-tuning of vision-language-action models. arXiv preprint arXiv:2609.22973, 2026. URL https://arxiv.org/abs/2609.22973.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: A progressive robustness benchmark for visual-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 38574–38583, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Fei_LIBERO-Plus_A_Progressive_Robustness_Benchmark_for_Visual-Language-Action_Models_CVPR_2026_paper.html.

Vinit Hegiste, Tatjana Legler, and Martin Ruskowski. Application of federated machine learning in manufacturing. In *2022 International Conference on Industry 4.0 Technology (I4Tech)*, pp. 1–8. IEEE, 2022. doi: 10.1109/I4Tech55392.2022.9952385. URL https://ieeexplore.ieee.org/document/9952385.

Yuchao Hou. FedWorld: Scope-aware federation of agent world models. arXiv preprint arXiv:2608.01561, 2026. URL https://arxiv.org/abs/2608.01561.

Daliang Li and Junpu Wang. FedMD: Heterogenous federated learning via model distillation. arXiv preprint arXiv:1910.03581, 2019. URL https://arxiv.org/abs/1910.03581.

Tao Lin, Lingjing Kong, Sebastian U. Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 2351–2363. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/18df51b97ccd68128e994804f3eccc87-Abstract.html.

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. doi: 10.52202/075280-1939. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8c3c666820ea055a77726d66fc7d447f-Abstract-Datasets_and_Benchmarks.html.

Jiangran Lyu, Kai Liu, Xuheng Zhang, Haoran Liao, Yusen Feng, Wenxuan Zhu, Tingrui Shen, Jiayi Chen, Jiazhao Zhang, Yifei Dong, Wenbo Cui, Senmao Qi, Shuo Wang, Yixin Zheng, Mi Yan, Xuesong Shi, Haoran Li, Dongbin Zhao, Ming-Yu Liu, Zhizheng Zhang, Li Yi, Yizhou Wang, and He Wang. LDA-1B: Scaling latent dynamics action model via universal embodied data ingestion. In *Proceedings of Robotics: Science and Systems*, 2026. doi: 10.15607/RSS.2026.XXII.210. URL https://www.roboticsproceedings.org/rss22/p210.html.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282. PMLR, 2017. URL https://proceedings.mlr.press/v54/mcmahan17a.html.

Cui Miao, Tao Chang, Meihan Wu, Hongbin Xu, Chun Li, Ming Li, and Xiaodong Wang. FedVLA: Federated vision-language-action learning with dual gating mixture-of-experts for robotic manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6904–6913, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Miao_FedVLA_Federated_Vision-Language-Action_Learning_with_Dual_Gating_Mixture-of-Experts_for_Robotic_ICCV_2025_paper.html.

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

Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. FedProto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8432–8440, 2022. doi: 10.1609/aaai.v36i8.20819. URL https://ojs.aaai.org/index.php/AAAI/article/view/20819.

Anurag Upadhyay, Xin Lu, Yashar Baradaranshokouhi, Jun Li, and Yanguo Jing. F-DRL: Federated dynamics representation learning for robust multi-task reinforcement learning. *Information*, 17 (3):232, 2026. doi: 10.3390/info17030232. URL https://www.mdpi.com/2078-2489/17/3/232.

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. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/2c37c5bcef24b9541550261dcd63261b-Abstract-Conference.html.

Yuyang Zhang, Wenyao Zhang, Zekun Qi, He Zhang, Haitao Lin, Jingbo Zhang, Yao Mu, Xiaokang Yang, Wenjun Zeng, and Xin Jin. ImageWAM: Do world action models really need video generation, or just image editing? arXiv preprint arXiv:2606.19531, 2026. URL https://arxiv.org/abs/2606.19531.

Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12878–12889. PMLR, 2021. URL https://proceedings.mlr.press/v139/zhu21b.html.

A Data and Evaluation Protocol

The official RLDS release contains 14,347 records in 1,024 shards, covering 40 base tasks and 1,681 original demonstration groups. The four visual domains used here contain 11,475 records. The language directory retains the original 40 instructions; the study focuses on lighting, viewpoint, noise, and background variation.

Table 6 gives the audited source allocation. Each original demonstration and its rendered variants retain a common role across clients. Training supplies native updates and prediction packets, calibration sets reception coefficients, development selects settings, and prediction-test sources assess future features. The training-only public pool is excluded from every method's private training allocation.

Table 6: Source groups by split; public groups excluded from private training.

Domain	Train	Cal.	Dev.	Test	Public
Lighting	1,090	202	169	154	66
Viewpoint	1,088	202	169	154	66
Noise	1,087	200	168	154	65
Background	1,090	202	169	154	66
Global unique	1,090	202	169	154	66

The allocation contains 7,444 private training records and 1,097,083 continuous eight-action windows. Calibration, development, and prediction test contain 1,356/1,144/1,064 records and 205,323/168,446/158,483 windows. The public pool contains 467 records and 66,933 windows from 66 source groups. It covers all tasks globally; its noise subset lacks one task. Public queries provide deployment observations, instructions, and state, while future labels remain local.

For architecture comparisons, training source groups are divided into task-balanced shares S_0 and S_1, retaining source assignments across visual domains. Each domain has one client per share. Assigning LaWAM to both shares, ImageWAM to both, or one architecture per share defines the homogeneous and mixed populations. Assignment swaps control allocation effects.

Rollout cases identify the task, visual variant, initial state, environment seed, simulator version, and evaluation role. Development and test use separate case manifests, and comparisons use the same valid cases. Original LIBERO and LIBERO-Plus have distinct manifests. The frozen-checkpoint audit contains 10,030 benchmark rollouts; Appendix K reports its visual-domain subset. Replay-state overlap between demonstrations and evaluation is unresolved because complete demonstration simulator conditions are unavailable. Windows use consecutive action indices; the nominal 20 Hz convention requires a separate timing check when matching physical horizons.

B Optimization Schedule

The reference schedule uses $K_0 = 1{,}000$ initial updates and 20 rounds with $K_{\mathrm{pre}} = K_{\mathrm{post}} = 600$, totaling 25,000 native updates per client. Reference settings use effective native batch 32, 1,000 server updates per round with batch 64, and a 256 MiB packet cap with FP16 predicted features. The designated training seed is 43. These are schedule specifications; executed per-run settings are part of the record coverage described in Appendix H.

Native objectives, action normalization, and trainable modules follow the corresponding architecture. Microbatching and accumulation may differ while retaining the effective batch. Shared preprocessing defines the predictive interface; native visual inputs retain their own preprocessing. Local native updates, module-specific updates, and unique server-teacher updates are recorded separately. The final evaluation endpoint follows the last reception and action-adaptation block. The five development checkpoints correspond to 1,000/7,000/13,000/19,000/25,000 native updates.

C Baseline and Sharing Definitions

Local controls. Pretrained evaluates the released checkpoint. Local (native) uses native supervision; Local additionally includes any interface alignment required by FedWAM, with $\lambda_{\mathrm{fed}} = 0$. They coincide for LaWAM, whose native target already provides alignment. Self-distillation fits a separate teacher from each recipient's own packets and retains the same reception procedure. Teacher fitting therefore remains distinct from access to another client's predictions.

Federated controls. FedAvg-World averages compatible world parameters with equal client weights, retaining private conditioning and action experts. Federated distillation fits the common teacher from predictions on the training-only public pool. Same-architecture sharing restricts contributors to the recipient's architecture. FedWAM uses the full compatible packet pool; its ungated variant sets $g_i^r = 1$. Centralized trains on pooled permitted sources with matched aggregate sample exposure. The comparison protocol matches local initialization, native data exposure, update schedules, and post-reception action adaptation within each architecture.

Directional donor comparison. The donor-only comparison fixes recipient shares, initialization, and local updates, while matching donor sources, task/window coverage, count, and serialized payload. Teachers have the same capacity and optimization budget, and exclude the recipient's own packets. Donors receive no recipient feedback. Repeating the comparison for both recipient families measures direction-specific transfer. Donor architectures retain their native training recipes, so the contrast concerns each architecture together with its recipe. The full mixed pool additionally includes same-architecture and self-contributions.

D Metrics and Aggregation

Let $\mathcal{C}_{b,p}$ contain evaluation cases for task b and domain p, and let $Y_{i,m,c} \in \{0,1\}$ denote recipient i's success under method m. With $\mathcal{B}_p$ the represented tasks, the task-macro score is

$$S_{i,m,p} = \frac{1}{|\mathcal{B}_p|} \sum_{b \in \mathcal{B}_p} \frac{1}{|\mathcal{C}_{b,p}|} \sum_{c \in \mathcal{C}_{b,p}} Y_{i,m,c}.$$

LIBERO-Plus columns average recipients within a domain, and its overall score equally weights the four domains. Original-LIBERO columns average tasks and recipients within each suite; the overall score equally weights the four suites. Both use their respective evaluation manifests.

Own averages the four diagonal recipient/domain cells, and Other averages the twelve off-diagonal cells. Thus $\mathrm{Avg}_{\mathrm{Plus}} = (\mathrm{Own} + 3\,\mathrm{Other})/4$. Other refers to another participating client's domain. Recipient means average all four domains; their minimum is a supplementary descriptive statistic.

Positive transfer counts paired cases changing from Local failure to success; negative transfer counts the reverse. Both use all paired valid cases as the denominator within each task, followed by the same macro weights as success. Their difference equals the mean success gain before rounding. Rates conditional on Local success or failure use different denominators. Comparisons in this paper report point estimates; uncertainty and training-repeat coverage are recorded in Appendix H.

Prediction MSE follows Equation (5), averaging windows within sources, sources within tasks, tasks within clients, and clients equally. High-change windows exceed the development 75th percentile of $d(y^*, c_t)$ under the same weighting. Prediction-test labels are reserved for offline scoring. The final-round local snapshot, teacher, and post-reception model share test windows, while Local and self-distillation use matched final native-update endpoints. Action-validation losses are compared within an architecture, with fixed development batches and normalization.

E Extended Results

Table 7 reports recipient means over the four visual domains. FedWAM improves all four means by 1.9–2.3 pp over Local. These row averages complement the domain-wise scores in the main table and the case-level transfer rates. Minimum denotes the smallest observed recipient mean under the same evaluation, while the full matrix records variation across environments.

Table 7: Recipient success (%). L/V/N/B: light, view, noise, background.

Method	L	V	N	B	Min.
Local	74.8	78.0	77.0	75.4	74.8
FedAvg-World	75.6	78.6	77.4	76.3	75.6
Fed. distill.	74.9	77.9	76.9	75.5	74.9
Self-distill.	75.9	79.2	78.2	76.5	75.9
FedWAM	77.0	80.0	78.9	77.5	77.0

Table 8 gives the FedWAM matrix underlying Figure 4. The viewpoint specialist reaches 58.0% on viewpoint, compared with 45.0% for the other recipients, illustrating specialization within a shared training protocol. Original-LIBERO results remain separate suite-level aggregates in Table 1.

Table 8: FedWAM client-by-domain success (%).

Client	Light	View	Noise	Bkg.
Lighting	95.5	45.0	73.0	94.6
Viewpoint	94.0	58.0	75.0	93.0
Noise	94.0	45.0	83.0	93.5
Background	94.5	45.0	74.0	96.5

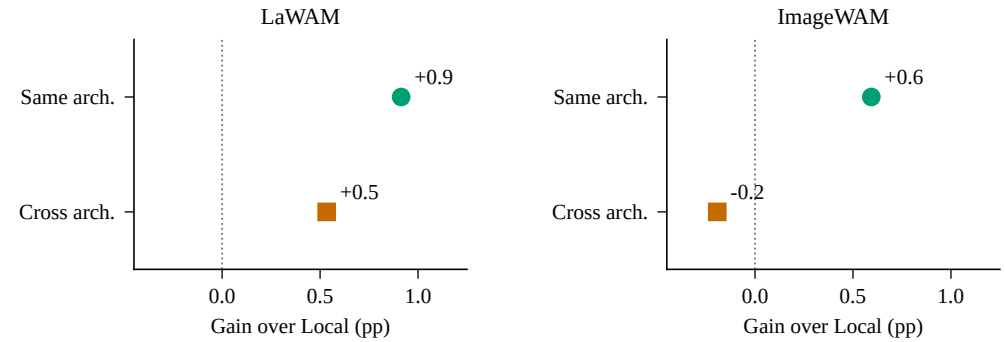


Figure 8: Matched-donor gains over Local (pp), by recipient architecture.

Figure 8 separates the two donor directions. Same-architecture donors improve both recipient families; cross-architecture donors give $+0.5$ pp for LaWAM and -0.2 pp for ImageWAM. This directional view complements the full mixed-pool gains in Table 3, where recipients also receive predictions from their own architecture.

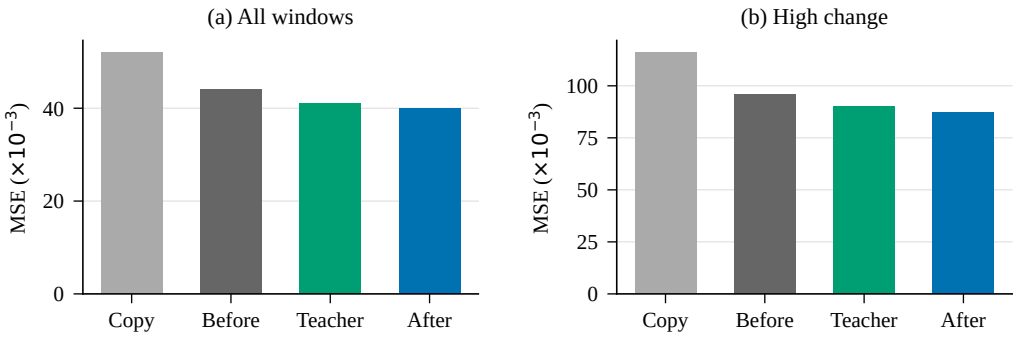


Figure 9: Chunk-end feature MSE ($\times 10^{-3}$), by baseline and reception stage.

Figure 9 compares prediction errors on the same source-balanced test windows. The copy-current baseline is higher in the high-change subset, while the post-reception predictor improves on both local baselines. The pre/post contrast includes further native learning; matched Local and self-distillation endpoints provide the complementary controls.

F Mechanism Diagnostics

Calibrated reception. The paired gate diagnostic fixes the pre-reception client, server teacher, and packets before branching into calibrated and ungated reception. The reported contrast is $+0.8$ pp. This measures one reception decision from a common state; the full-schedule ungated baseline in Table 3 measures the accumulated effect over training.

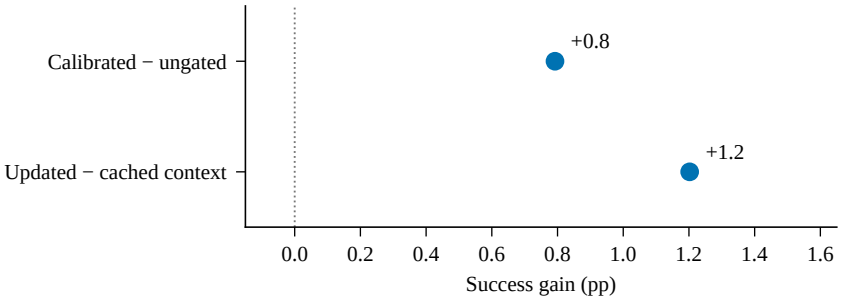


Figure 10: Control gains (pp) from calibrated reception and updated action context.

Action adaptation to updated context. Two action experts start from identical weights and receive the same action-only update budget and samples. One uses cached pre-reception world context during training; the other uses updated context. Both are evaluated with the same fixed updated world model. The $+1.2$ pp contrast therefore concerns how the private controller uses its representation, with predictive output fixed. The calibration and context contrasts are separate interventions and are not additive estimates of a combined effect.

Future and transition parameterizations. With current features c included in the condition, the paired teachers $T_{\text{future}}(q) = c + T_{\text{transition}}(q)$ satisfy

$$d(T_{\text{future}}(q), y^*) = d(T_{\text{transition}}(q), y^* - c).$$

Subtracting the known current feature preserves information and squared error. Empirical differences can instead arise from architecture, optimization, normalization, or quantization. This identity defines the representation comparison; no measured future/residual contrast is reported here.

G Native Prediction Visualizations

G.1 Native predictions: complete examples and visualization protocol

These examples document the native predictive outputs of the original LaWAM LIBERO-SFT and ImageWAM-FLUX.2-4B-LIBERO checkpoints. Neither checkpoint received federated updates for this visualization. Both recordings use seed 42 and the bowl-placement task, but their native horizons and execution settings differ; they are not a matched comparison of prediction quality or completion speed. No generated intermediate video frames, image editing, or ground-truth future images substitute for model predictions. Main-text panels use rounded uniformly spaced indices within each recorded sequence, with all recorded predictions retained below. These single-rollout illustrations do not estimate task success rates or cross-client sharing benefits.

Native latent subgoals. Figure 11 shows predictions from the original LaWAM checkpoint's control path (Chen et al., 2026). Four uniformly spaced eight-action chunks illustrate the features supplied to its action expert during a successful rollout. RGB shows the current observation; PCA colors summarize the predicted spatial features; Sim. overlays their similarity to a fixed initial arm patch on the actual chunk endpoint. The green box marks that patch. A common PCA mapping and cosine scale are used throughout, and no future observation enters the prediction. These features are neither reconstructed RGB nor attention or segmentation probabilities. This native-model example illustrates the interface, not a federation gain. Appendix G.1 retains all ten chunks and the full visualization protocol.

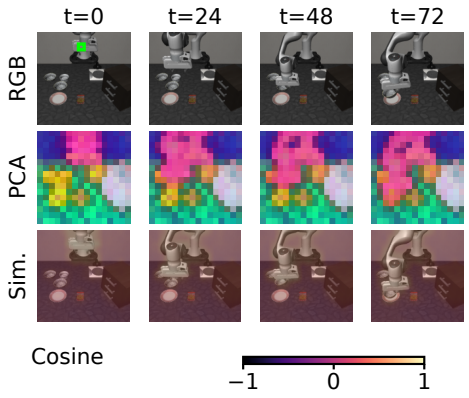


Figure 11: Native LaWAM subgoals. Rows: current RGB, predicted-feature PCA, and patch similarity at $t + 8$. Columns mark chunk starts.

LaWAM control trace and spatial features. The rollout uses `libero_spatial`, task 0, initial state 0, with 10 settling steps, a 20 Hz controller, an eight-action horizon, and replanning after all eight actions. It reaches simulator success after 80 executed actions. The ten current and predicted feature tensors are captured directly from the inference path feeding the action expert; each has 256 spatial tokens with 768 channels, displayed as a 16×16 grid without pooling. Figures 12 and 13 retain every complete chunk, in temporal order. The actual endpoint is the simulator observation after the corresponding eight actions.

Following the patch-similarity visualization idea in LaWAM (Chen et al., 2026), we select a single robot-arm patch from the initial RGB image only: zero-based grid row 2, column 7, or pixels $x = 112 \ldots 127$, $y = 32 \ldots 47$. Its current-feature vector is fixed for the whole rollout. Each heatmap gives the cosine similarity of that vector to every predicted spatial token, using a common $[-1, 1]$ scale. Display interpolation is bilinear, with opacity 0.5 over the actual endpoint; it does not change the stored 16×16 similarities. This similarity is not an attention map, segmentation label, or calibrated object-location probability.

PCA is fitted to the initial 256 current-feature vectors only. We keep three components and fix each component's sign so that its largest-magnitude loading is positive. The same mean and basis project every prediction. Each component's color range uses the common minimum and maximum over current and predicted features across the recorded sequence; no true future features enter this fit or scaling. PCA colors use nearest-neighbor display interpolation and do not represent RGB appearance. This is our implementation of a paper-style visualization, not a claim of reproducing the authors' exact plotting code, and no latent editing is performed.

ImageWAM generation and closed-loop reference. The original ImageWAM checkpoint generates each future image with its native `infer_video_flux2` branch, ten denoising steps, seed 42, and the native FLUX.2 VAE. The decoded 224×448 RGB image contains front and wrist views side by side; the figures consistently display its left 224×224 front view. Inputs contain current observations, language, and robot state, with neither true future observations nor the subsequently executed actions provided to the generator. This native RGB generation branch is distinct from the DINO readout defining our shared feature interface.

The reference observations are obtained by replaying the 83 recorded actions of a successful native rollout. Six saved observations at action indices 0, 17, 33, 50, 66, and 83 match the original lossless frames pixel for pixel, and replay preserves the success flag. The six prediction starts are fixed at 0, 12, 24, 36, 48, and 60, with reference endpoints 16 actions later. The native controller replans every 12 actions. Therefore each endpoint follows closed-loop control across a replanning event, not open-loop execution of the original 16-action chunk. This comparison does not validate causal prediction under a specified executed action sequence.

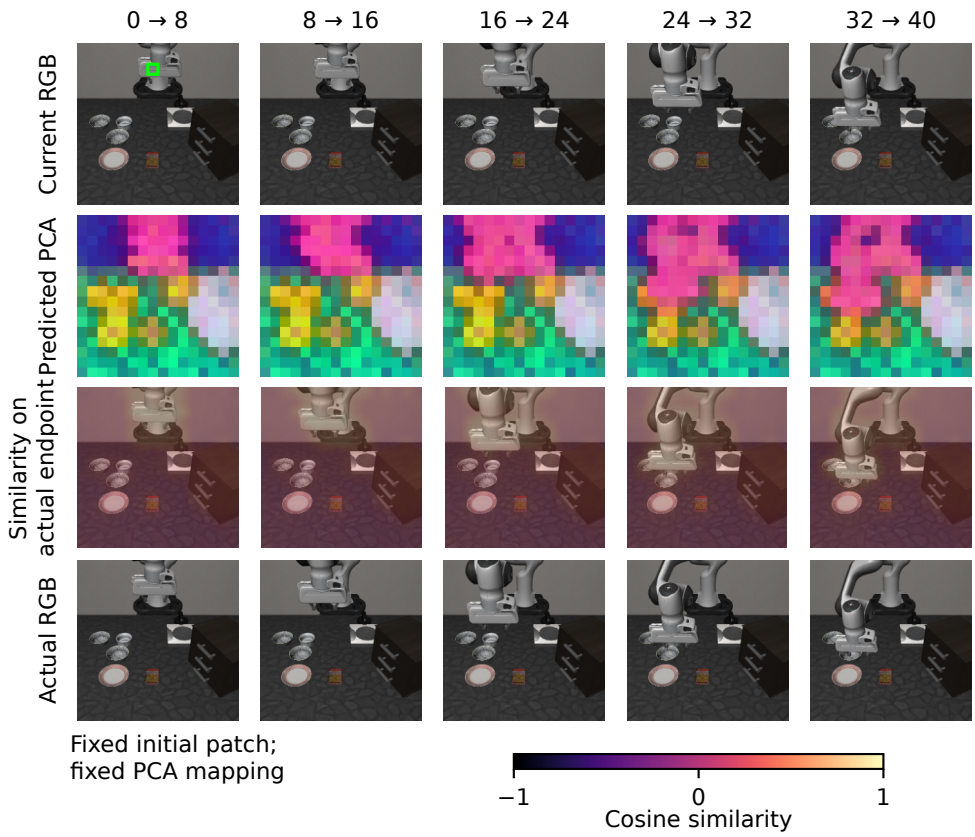


Figure 12: **Complete LaWAM native subgoal trace, chunks 1–5.** All five consecutive predictions are shown with current RGB, predicted-feature PCA, fixed-patch similarity on the actual endpoint, and unblended actual endpoint RGB. Each column spans eight executed actions; PCA and similarity conventions are shared with Figure 11.

Figure 14 retains all six queries, including $24 \rightarrow 40$, whose RGB mean squared error exceeds the copy-current baseline in the recorded diagnostic. This small example set is descriptive, not an estimate of aggregate prediction accuracy. The associated six-query comparison video is a slideshow, not a model-generated continuous future video. LaWAM's accompanying rollout and subgoal-overlay videos use actual simulator frames at 20 Hz; predictions are held fixed within their corresponding executed chunks.

H Result Provenance

The plotted comparison arrays originated in a synthetic layout-preview artifact. The authors subsequently identified the complete arrays as measured and authorized their use. The original generation history remains in the source record, and new raw runs have not been supplied to reconcile that conflict. Accordingly, these are author-supplied point estimates with unresolved provenance. The historical rollout records in Appendix K and native prediction media in Appendix G.1 have separate source records.

The data file preserves the source hash, and plotting scripts reproduce matrix averages, curves, and diagnostics without adding confidence intervals. Complete run/configuration/checkpoint mappings, executed module updates and timing, physical demonstration inventories, trial denominators, and paired uncertainty estimates remain outstanding. Source manifests identify task, domain, split, and window; rollout records additionally identify initial state, seed, outcome, and checkpoint. These records distinguish numerical figure reproduction from independent experimental verification.

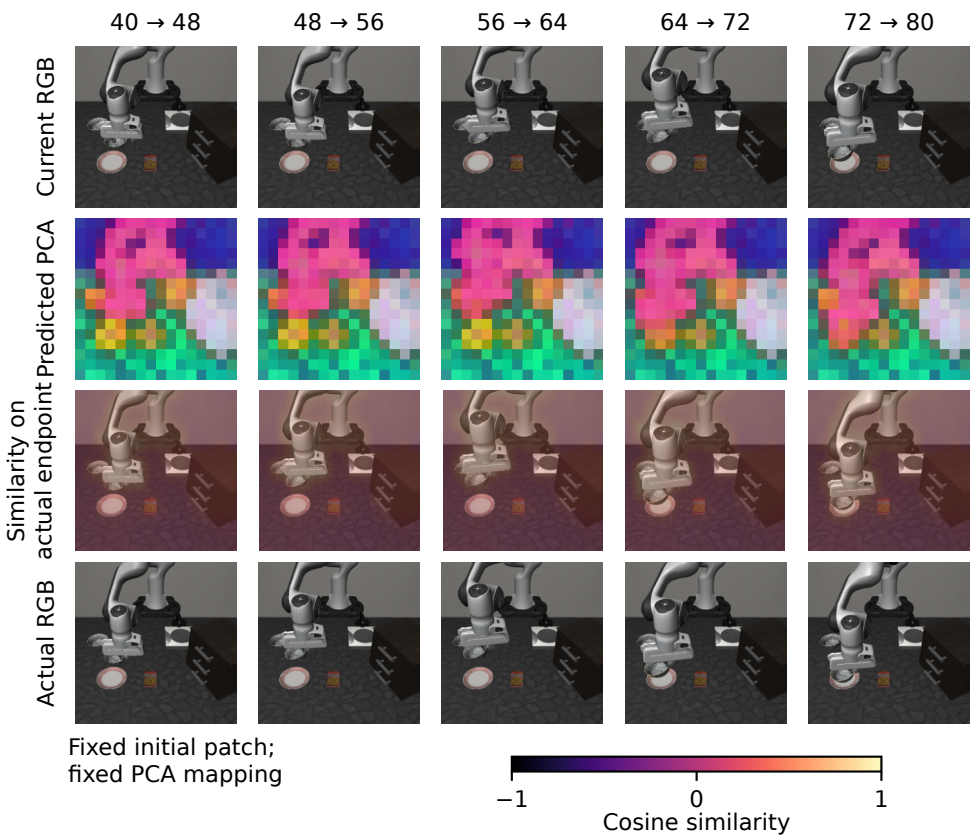


Figure 13: **Complete LaWAM native subgoal trace, chunks 6–10.** The remaining five predictions continue Figure 12, with the same initial reference patch, PCA basis, and color scales. The final observation follows action 80, when the simulator reports success. The full ten-chunk sequence is retained without filtering for visual prediction quality.

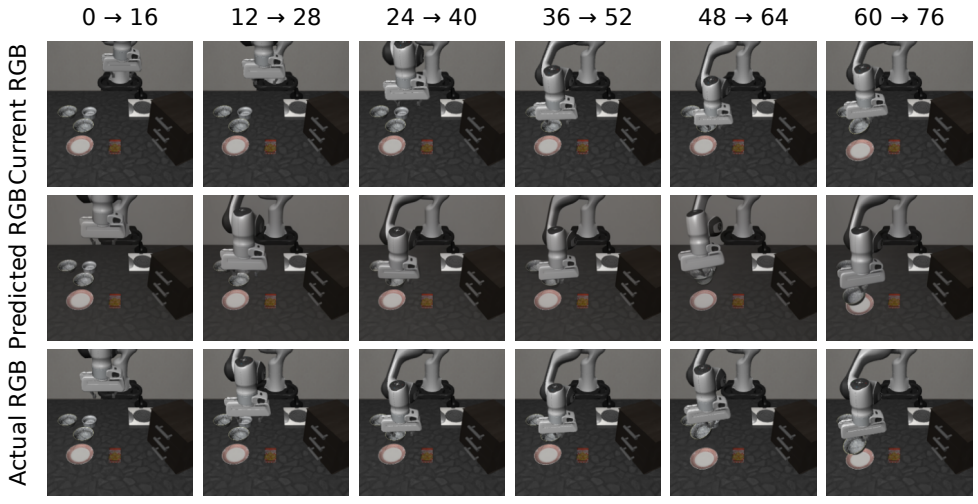


Figure 14: **All six recorded ImageWAM native future-image predictions.** Rows show current RGB, model-generated endpoint RGB, and actual closed-loop RGB at $t + 16$. Columns preserve the six fixed query starts without removing prediction errors. All images use the front-view crop; the reference crosses replanning at $t + 12$. These native-checkpoint examples do not measure federated transfer.

I Predictive Interface and Optimization Details

This section gives the interface, update paths, and conditional-regression interpretation underlying Sections 3 and 4.

I.1 Federation schedule

Let $\Theta_i^{r,+}$ denote client i's post-reception parameters and $\mathcal{L}_i^{\mathrm{base}}$ the local objective without distillation. Algorithm 1 initializes native interfaces, then alternates local prediction, server fitting, and calibrated reception. Clients run independently within local stages and synchronize at packet collection and broadcast. The final deployed controller includes post-reception adaptation.

Algorithm 1 FedWAM with private native controllers

Require: Source-partitioned local data; pretrained WAMs; shared interface; R_{fed} rounds; update budgets $K_0, K_{\mathrm{pre}}, K_{\mathrm{post}}$ and server budget

Fix the shared encoders and target protocol; initialize teacher ω^0.
for each client i independently **do**
  Take K_0 initial native/interface updates on $\mathcal{D}_i^{\mathrm{tr}}$.
  Freeze fitted readouts where applicable; retain $\Theta_i^{0,+}$.
end for
for $r = 1, \ldots, R_{\mathrm{fed}}$ **do**
  for each client i independently **do**
    From $\Theta_i^{r-1,+}$, take K_{pre} updates on $\mathcal{L}_i^{\mathrm{base}}$.
    Retain snapshot $\Theta_i^{r,-}$ and upload $\mathcal{P}_i^r$ (Equation (4)).
  end for
  Fit ω^r from ω^{r-1} on Equation (5) for the server budget.
  Broadcast the same teacher T_{ω^r} to all clients.
  for each client i independently **do**
    Compute g_i^r on $\mathcal{D}_i^{\mathrm{cal}}$ using $\Theta_i^{r,-}$ (Equation (6)).
    From $\Theta_i^{r,-}$, take K_{post} joint updates on Equation (8).
    Retain all updated local parameters as $\Theta_i^{r,+}$.
  end for
end for

Ensure: Adapted local WAMs $\{\Theta_i^{R_{\mathrm{fed}},+}\}_{i=1}^N$ for independent deployment

I.2 Native mappings and teacher optimum

For completeness, the conditional teacher optimum and the native predictive mappings are

$$T_r^*(q) = \mathbb{E}_{\overline{\mathsf{P}}^r}\left[\widehat{Y} \mid Q = q\right]. \tag{9}$$

$$u_i = C_{\psi_i}(x), \qquad \widehat{Z}_i = W_{\theta_i}(E(o_t), u_i; \tau), \qquad F_i(x) = R(\widehat{Z}_i^{\mathrm{ref}}). \tag{10}$$

$$F_i(x) = P_{\eta_i}(h_i(x)), \qquad \mathcal{L}_{\mathrm{align},i} = \mathbb{E}_{\mathcal{D}_i^{\mathrm{tr}},\mathrm{bal}}\, d(P_{\eta_i}(h_i(x)), y^*). \tag{11}$$

The conditioning, world, and readout parameters are ψ_i, θ_i, and η_i. LaWAM maps predicted intent u_i to future tokens; ImageWAM's fixed readout maps control-connected prefix features to the same targets. The distribution $\overline{\mathsf{P}}^r$ is the client-balanced packet mixture defined below.

Shared interface contract. An interface version v fixes the reference-view convention, image preprocessing, visual encoder checkpoint and layer, spatial transform R, token layout, feature normalization, text encoder, state schema, and physical horizon τ. State uses common units and normalization; unavailable entries are zero-filled and identified by m_t. Native models may retain additional views and private conditioning. Their prediction targets must refer to the agreed physical endpoint: a spatial adapter cannot correct a temporal mismatch. In Equation (10), τ identifies the decoder's training target and need not be a new runtime argument. The shared conditions describe task execution from current inputs; they do not define interventions with an externally specified action sequence.

Packet construction and server fitting. Each packet carries its round, interface version, numerical precision, and sampling metadata. Only compatible packets from the current round enter the teacher fit. The common serialized-byte cap covers both conditions and predicted features, so row counts may vary. Sampling balances tasks and source demonstrations and covers multiple temporal windows. A fixed source-group split divides training-derived packets into fitting and validation subsets. Validation monitors regression to client predictions; it does not select an early-stopping checkpoint or access clients' true future targets. Teacher architecture and capacity are fixed across rounds and model compositions, and each round retains the final state after the declared update budget.

The balanced expectation in Equation (5) expands as

$$\mathbb{E}_{\mathcal{P}_i^{r,\mathrm{fit}},\,\mathrm{bal}} f(q,\widehat{y}) = \sum_{m\in\mathcal{I}_i^{r,\mathrm{fit}}} w_{im}^r f(q_{im},\widehat{y}_{im}^r), \qquad \sum_m w_{im}^r = 1.$$

Here $\mathcal{I}_i^{r,\mathrm{fit}}$ indexes the fitting rows. We assign equal mass to represented tasks, equal mass to source groups within each task, and equal mass to sampled windows within each source. The outer average in Equation (5) gives every client equal total mass, including the recipient's own contribution in the full-pool method. Local balanced expectations use the same window–source–task hierarchy.

Calibration and representation choice. The calibration subset is reused across rounds to set one scalar g_i^r per client; it is part of adaptation and is not an independent final test. The local comparator is the pre-reception snapshot that produced the current packet. Teacher weights, reception coefficients, and shared encoders stay fixed during the post-reception block. For the transition variant, packet predictions and local targets both subtract c_t. With $T_{\mathrm{future}}(q) = c_t + T_{\mathrm{transition}}(q)$, subtracting the same c_t from prediction and target preserves the squared error pointwise. The comparison must therefore declare any differences in parameterization, normalization, or quantization.

Native objectives and added gradients. Native losses retain each architecture's specified trainable modules and gradient boundaries. In LaWAM, the added federated loss detaches predicted intent u_i and updates the world decoder. Future-derived latent-action labels are used only by native training, never as inputs to uploads or calibration. The primary native action loss receives detached future features, so changing those features affects action inputs without creating a new action-loss gradient into the decoder.

ImageWAM's readout in Equation (11) uses spatial queries with cross-attention to selected text/reference-image prefix features. Cache layers and readout capacity are selected during development and then fixed. During initial training, the readout fits future targets from detached features while the native model follows its image/action objectives. Afterward, the readout weights remain fixed but its input derivative stays active. Both added alignment and federated losses train the declared world-branch parameters upstream of those features; private conditioning is detached at that branch's input and action-expert parameters are excluded from these added gradients. Native losses retain their own permitted paths, so this isolation does not freeze conditioning during native training.

All ImageWAM interface computations use no future-image tokens and image flow time zero, matching the context used for native action inference. Image flow time is distinct from the physical prediction horizon τ. Training requires a differentiable prefix pass: an inference routine executed without gradient tracking cannot implement absorption. A gradient audit must check which world parameters are ancestors of the selected features. Freezing the readout establishes the intended update path; whether those updates help control is an empirical question.

I.3 Conditional regression interpretation

The characterization in Equation (9) follows directly from squared-error regression. Fix a round and let P_i^r be client i's balanced distribution of fitting condition–prediction pairs. For a finite packet, this is the weighted empirical distribution defined above. Sampling a client uniformly gives $\overline{\mathsf{P}}^{\,r} = N^{-1}\sum_i \mathsf{P}_i^r$. Assume the prediction $\widehat{Y}$ has finite second moment, and consider measurable teachers with finite squared error. With $T_r^*(q) = \mathbb{E}[\widehat{Y} \mid Q = q]$, the conditional residual has mean zero, giving

$$\mathbb{E}\, d(T(Q),\widehat{Y}) = \mathbb{E}\, d(T(Q),T_r^*(Q)) + \mathbb{E}\, d(T_r^*(Q),\widehat{Y}).$$

Thus T_r^* minimizes the unrestricted regression risk, up to equality almost everywhere on the mixture's support. The residual term represents disagreement that cannot be resolved from Q alone.

Let I denote the sampled client. Conditioning further on I yields

$$T_r^*(q) = \sum_{i=1}^{N} \alpha_i^r(q)\, \mathbb{E}[\widehat{Y} \mid Q = q, I = i], \qquad \alpha_i^r(q) = \Pr(I = i \mid Q = q).$$

Only clients with positive conditional mass contribute. If the condition marginals have densities $p_i^r(q)$ relative to a common dominating measure, then $\alpha_i^r(q) = p_i^r(q)/\sum_j p_j^r(q)$ wherever the denominator is positive. These weights describe the objective's conditional optimum; the algorithm does not estimate them or introduce a separate routing mechanism.

A finite Transformer is trained for a fixed number of updates to approximate this unrestricted solution. For empirical packets with distinct conditions, the unrestricted residual can vanish; averaging or transfer between nearby conditions is induced by the restricted teacher. With disjoint packet support, a condition may have contributions from only one client, and transfer to other conditions relies on the model's inductive bias and generalization. Moreover, the regression labels are client predictions, so the decomposition neither removes their systematic errors nor implies better prediction of observed futures. The local calibration and closed-loop evaluations test those separate questions.

J Physical Evaluation Protocol

Platform and tasks. The specified hardware is AgileX Piper with a RealSense D455 reference RGB view. The four environment clients are clean, lighting, background, and camera viewpoint; each contributes local training data and is evaluated in all four environments. The tasks are Corn to Plate, Pen to Basket, and Cube to Bowl. The head/wrist photographs in Figure 6 illustrate the platform using existing recordings. Each client predicts its own camera view; shared feature coordinates preserve the declared view convention rather than defining geometric correspondence across cameras.

Data and control contract. Demonstration roles are assigned to complete source episodes and their derived copies, retaining capture-session identity. Calibration targets and action supervision stay local. Native Piper adaptation specifies command coordinates, units, gripper semantics, action normalization, prediction duration, and the executed replanning prefix. Physical training has its own update schedule. The comparison protocol matches initialization, native data exposure, objectives, and post-reception adaptation across methods; trial blocks are organized by task, environment, session, and reset procedure. Device counts, executed settings, task-success criteria, and trial-level records are part of the outstanding run metadata in Appendix H.

Aggregation. Let k_{mibt}/n_{mibt} be the observed success fraction for method m, recipient i, environment b, and task t, with a nonzero trial denominator. For the three tasks $\mathcal{T}$, define

$$S_{mib} = \frac{1}{|\mathcal{T}|} \sum_t \frac{k_{mibt}}{n_{mibt}}, \qquad U_{mbt} = \frac{1}{4} \sum_i \frac{k_{mibt}}{n_{mibt}}.$$

Figure 7 reports S_{mib}; Table 5 reports U_{mbt}. Both summaries equally weight their respective tasks or recipients. The overall mean is $\frac{1}{16}\sum_{i,b} S_{mib}$, equal to the mean of the three task averages. Own and Other average the four diagonal and twelve off-diagonal cells. Clean participates in physical training and evaluation and enters these means. Percentages multiply the quantities by 100.

Figure 15 retains each task's recipient/environment differences. These complement the overall five-point task gains with the location of improvements and regressions. Paired uncertainty estimation requires reset-block and session correspondence; repeated trials from one trained model describe execution variability rather than variability across training runs.

K Available Measurements and Historical Diagnostics

This section separates measurements by their executed protocol. The frozen checkpoint's four-domain baseline is imported into Table 1. The original-LIBERO reference in that table was supplied

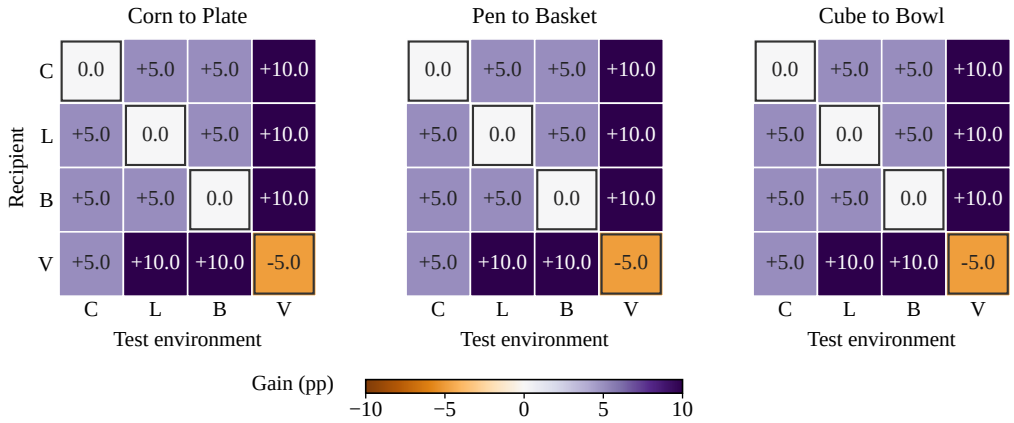


Figure 15: FedWAM gains over Local (pp) for each physical task. Rows are recipients and columns are environments; panels share one scale.

by the authors: Spatial 98.6%, Object 99.2%, Goal 97.8%, and Long 96.2%, with an equal-suite average of 98.0% after rounding (97.95% before rounding). Its raw evaluation records and checkpoint correspondence remain to be verified separately. Historical adaptation and readout-development studies do not evaluate the full four-client conditional-teacher method described in Section 4.

Frozen-checkpoint evaluation. The completed baseline contains 10,030 LIBERO-Plus rollouts, with 7,491 successes and no runtime errors. We retain only the four visual domains in this paper. The existing manifest distinguishes previously inspected or engineering cases from E_{confirm}, which excludes those known exposures. Table 1 uses all recorded cases in the four visual domains, preserving 40 base tasks per domain. The confirmation subset covers 39 viewpoint tasks, so its 40-task macro score is not reported (Table 9, dash). Replay-state overlap and evaluation-manifest correspondence remain to be verified. Each variant uses published initial-state index zero and a manifest-fixed seed; the policy executes eight actions per chunk at 20 Hz, with suite limits of 250/300/320/550 steps. One unchanged checkpoint supplies the pretrained reference for all recipients; it is evaluated once per case, not counted as four independent runs. Table 9 distinguishes pooled counts from task-macro success. The main average weights the four domain scores equally. No original, unperturbed LIBERO measurement is inferred from these perturbed trials.

Table 9: Pretrained LaWAM on four visual domains. Counts pool trials; macro success (%) equally weights tasks.

Subset	Domain	Tasks	Successes/trials	Macro (%)
All recorded	Lighting	40	1072/1142	94.8
All recorded	Viewpoint	40	575/1599	39.5
All recorded	Noise	40	1136/1601	71.6
All recorded	Background	40	1014/1076	95.4
E_{confirm}	Lighting	40	1045/1114	94.7
E_{confirm}	Viewpoint	39	564/1572	–
E_{confirm}	Noise	40	1116/1573	71.7
E_{confirm}	Background	40	1010/1072	95.4

Historical three-client adaptation. Table 10 reports an earlier LaWAM study on lighting, viewpoint, and noise: four LIBERO-Spatial base tasks and 48 fixed environment variants. Each of three client models is evaluated on all 48 cases, giving 144 model–case evaluations per method and training seed: 48 own-domain and 96 other-domain evaluations. Seeds 42, 43, and 44 reuse environment seeds and initial states; these are not 144 independent tasks. Training freezes the backbone and intent module and uses lightweight world/action adaptation (200 initial steps and four blocks of 200 steps). Prediction distillation and self-distillation here denote the historical implementations, not the conditional-teacher and client-only-teacher methods of the main study.

The "After" stage adds 200 action-only updates per client, with world and intent fixed. Both stages contain 1,728 evaluated model–case combinations. Because this diagnostic adds training on previously inspected cases, it measures the combined effect of continued adaptation and updated context. The table records successes per seed and own/other-domain means; variability is the sample standard deviation across three training seeds.

Table 10: Historical three-client success before/after action adaptation. Seed columns: successes/144; other columns: success (%). All reports mean ± seed standard deviation.

Stage	Method	Seed 42	Seed 43	Seed 44	Own	Other	All
Before	Local	102/144	88/144	94/144	70.1	63.5	65.7 ± 4.9
Before	World averaging	98/144	92/144	92/144	71.5	62.2	65.3 ± 2.4
Before	Prediction distillation	96/144	91/144	89/144	71.5	60.1	63.9 ± 2.5
Before	Self-distillation	93/144	99/144	96/144	69.4	65.3	66.7 ± 2.1
After	Local	101/144	99/144	97/144	74.3	66.0	68.8 ± 1.4
After	World averaging	102/144	101/144	97/144	72.9	67.7	69.4 ± 1.8
After	Prediction distillation	106/144	97/144	103/144	75.7	68.4	70.8 ± 3.2
After	Self-distillation	102/144	99/144	101/144	76.4	66.7	69.9 ± 1.1

Table 11 retains the corresponding historical future-feature errors, computed before additional action adaptation on an existing development cache of 80 transitions per domain. These are not independent prediction-test measurements and do not use the current cross-architecture reference contract. They therefore do not populate Table 4.

Table 11: Historical development MSE ($\times 10^{-3}$), mean ± standard deviation over seeds 42–44. These differ from the current prediction-test metrics.

Historical method	Own	Other	All
Local	156.0 ± 0.9	207.6 ± 1.1	190.4 ± 0.9
World averaging	160.3 ± 0.2	160.3 ± 0.2	160.3 ± 0.2
Prediction distillation	155.7 ± 0.7	176.6 ± 0.1	169.7 ± 0.2
Self-distillation	155.1 ± 0.7	197.0 ± 2.4	183.0 ± 1.8

ImageWAM readout development. Table 12 reports the completed initial readout-selection snapshot (seed 42), before the subsequent native-adaptation study. It compares two readout candidates with copying current features. Both candidates exceed the copy error in viewpoint and background, so neither passes the recorded requirement to improve over copying in every domain. These scores characterize readout development; control transfer is evaluated separately in the main study.

Table 12: Readout MSE ($\times 10^{-3}$), seed 42.

Domain	Copy	Query	Spatial
Lighting	35.6	32.7	30.9
Viewpoint	34.2	59.9	53.4
Noise	86.6	86.0	83.0
Background	32.6	60.2	56.9