zk0: Privacy-Preserving Federated Continual Learning for Vision-Language-Action Models in Robotics

Authors: Ivelin Ivanov[1], Community Contributors[2]

[1]zk0.bot, https://zk0.bot

[2]Decentralized SuperNode operators via https://github.com/ivelin/zk0

Abstract

Robotics foundation models, particularly Vision-Language-Action (VLA) policies, are constrained by acute data scarcity—real-world manipulation trajectories are ~1,000× scarcer than the textual corpora powering modern LLMs. Centralized aggregation of proprietary teleoperation data raises severe privacy risks and reinforces data monopolies. We present **zk0**, an open-source federated learning (FL) network that enables decentralized, privacy-preserving continual training of compact VLA models (SmolVLA, ~450M parameters) across heterogeneous real-world datasets. Built on the Flower framework with FedProx aggregation and adaptive hyperparameter scheduling, zk0 achieves stable convergence over 250 communication rounds on non-IID LeRobot SO-100/SO-101 tasks, reducing average client policy loss by 89% (from ~1.5 to 0.187) while integrating diverse skills (navigation primitives and fine-motor manipulation). Server evaluation on unseen SO-101 tasks stabilizes at a composite loss of 0.495—reflecting moderate adaptation from the pretrained zero-shot baseline (~0.15, implying 80%+ success) without catastrophic forgetting. This retention arises naturally from proximal regularization and gradual heterogeneous updates, offering a decentralized analog to centralized continual learning techniques like RETAIN [Yadav et al., 2025]. zk0 demonstrates the feasibility of community-driven scaling for robotics foundation models, with inherent privacy guarantees and emergent generalization across embodiments—paving the way for the humanoid AI era.

1. Introduction

The scaling hypothesis has driven transformative advances in language and vision models, yet robotics lags dramatically due to data starvation [Ivanov, 2025a]. High-quality teleoperation trajectories remain expensive, fragmented, and often proprietary, with public datasets (e.g., RT-1's ~100k episodes) orders of magnitude too small for generalist policies. Centralizing such data risks privacy violations (e.g., exposing home environments via RGB streams) and creates access barriers.

Federated learning provides a principled solution: participants train locally on private data and share only model updates for aggregation. zk0.bot operationalizes this for robotics VLA models, creating a persistent network where SuperNodes contribute real-world LeRobot-compatible trajectories to a

shared SmolVLA policy without raw data exchange—unlocking collaborative scaling while preserving privacy.

Concurrent works like FedVLA [Chen et al., 2025] explore federated VLA training, while FLAME [Kim et al., 2025] introduces manipulation benchmarks. zk0 advances the state of practice by demonstrating long-horizon convergence (250+ rounds) on genuinely heterogeneous real-world data, with dynamic scheduling for stability and explicit analysis of continual learning dynamics—showing limited dilution of pretrained capabilities analogous to centralized methods like RETAIN [Yadav et al., 2025].

2. Related Work

Federated Learning in Robotics. Early applications targeted navigation and swarm coordination. Recent benchmarks include FLAME for manipulation [Kim et al., 2025] and FedVLA for vision-language-action policies [Chen et al., 2025]. Challenges of non-IID data and heterogeneity are addressed via proximal regularization (FedProx [Li et al., 2020]) and clustering [Liu et al., 2024].

Continual Learning for Robot Policies. Centralized finetuning of VLA models suffers catastrophic forgetting on small task-specific datasets. RETAIN [Yadav et al., 2025] mitigates this via linear parameter merging:

$$\tilde{\theta} = (1 - \alpha)\theta_{\text{pre}} + \alpha\theta_{\text{ft}}$$

achieving strong OOD generalization and retention of generalist tasks. We draw parallels: zk0's distributed proximal term similarly anchors updates to pretrained priors, yielding comparable retention without explicit merging or data centralization.

VLA Models and Data Scaling. Compact models like SmolVLA leverage flow-matching objectives on mixed teleop/simulation data. Physical Intelligence and LeRobot findings underscore the value of heterogeneous real trajectories—precisely what zk0 aggregates federatively [Ivanov, 2025b,c].

3. Method

3.1 Problem Formulation

Each client $k \in [K]$ holds a private dataset $\mathcal{D}_k$ of trajectories $(\mathbf{o}_t, \mathbf{l}, \mathbf{a}_t)$. The global objective is a shared SmolVLA policy π_θ that minimizes empirical loss across all data while preserving privacy:

$$\min_\theta \sum_{k=1}^{K} p_k F_k(\theta), \quad F_k(\theta) = \mathbb{E}_{(\mathbf{o},\mathbf{l},\mathbf{a})\sim\mathcal{D}_k}[\ell(\pi_\theta(\mathbf{a}|\mathbf{o},\mathbf{l}))].$$

3.2 zk0 Architecture

zk0 employs Flower's client-server design. The central SuperLink server coordinates rounds, aggregates updates, and evaluates on held-out data. SuperNodes perform local training and submit weighted updates.

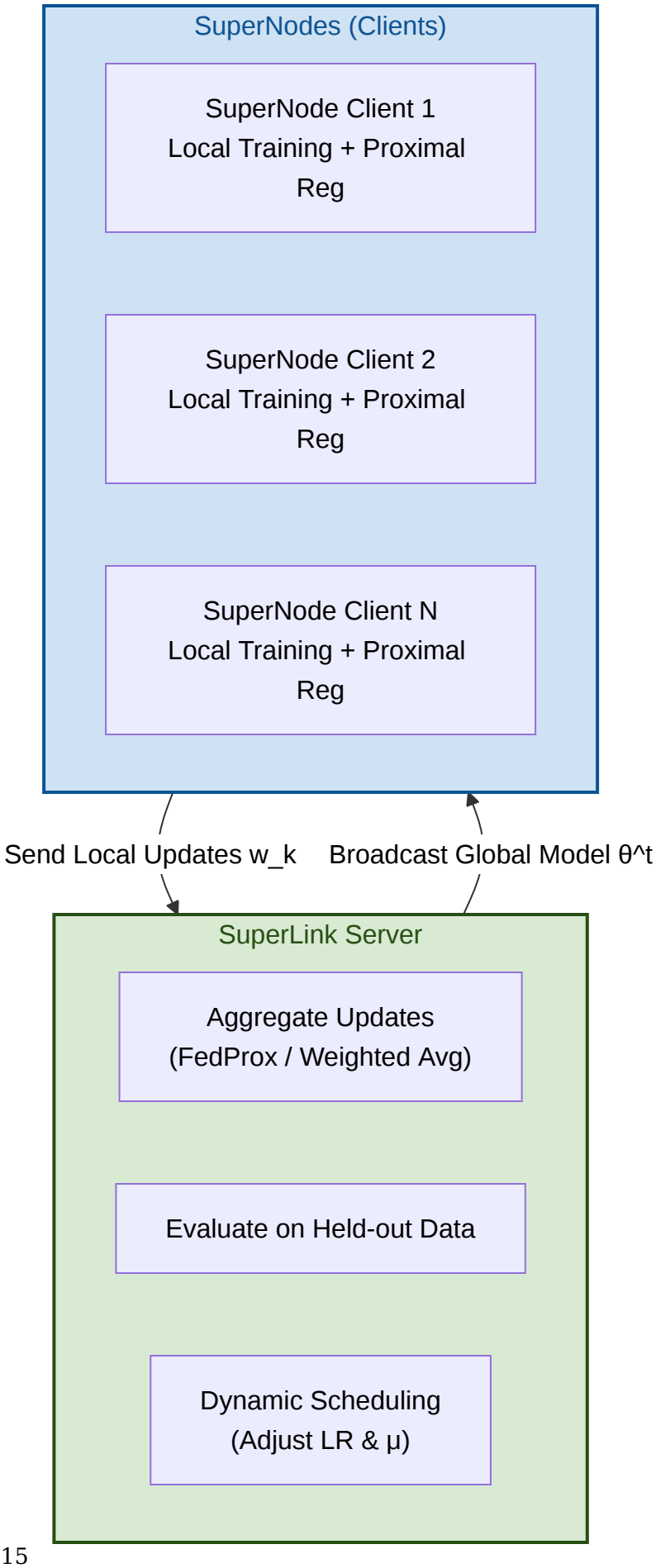

SuperNodes (Clients)
SuperNode Client 1
Local Training + Proximal Reg
SuperNode Client 2
Local Training + Proximal Reg
SuperNode Client N
Local Training + Proximal Reg
Send Local Updates w_k
Broadcast Global Model θ^t
SuperLink Server
Aggregate Updates
(FedProx / Weighted Avg)
Evaluate on Held-out Data
Dynamic Scheduling
(Adjust LR & μ)

Figure 1: Simplified zk0 architecture: Hub-and-spoke topology with central SuperLink server and multiple SuperNode clients, highlighting key operations for better readability.

To further clarify the temporal flow of interactions in a single round:

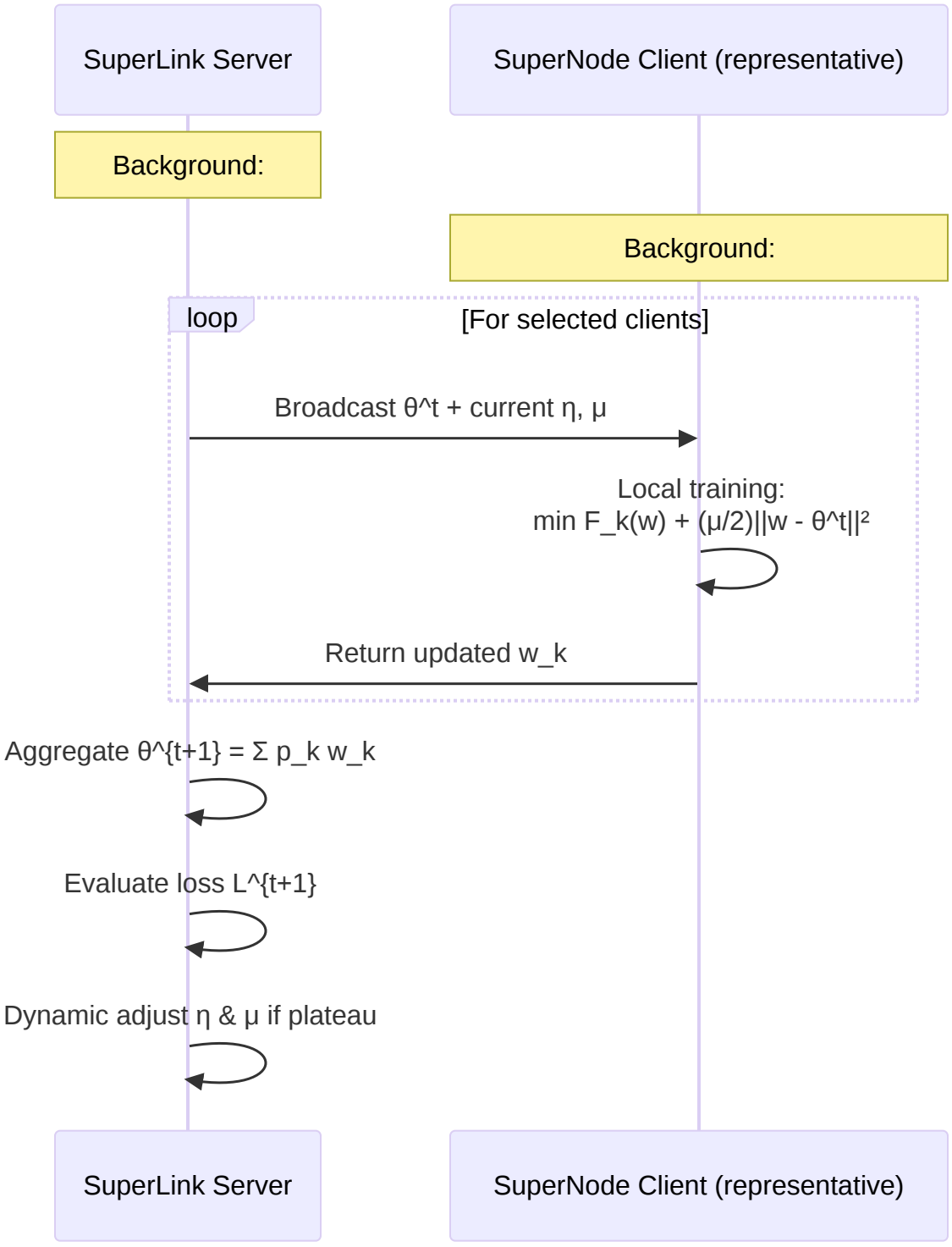


Figure 2: Sequence diagram of a single zk0 federated learning round, emphasizing proximal regularization and adaptive scheduling.

3.3 FedProx Aggregation with Dynamic Scheduling

Local optimization solves the proximal variant [Li et al., 2020]:

$$\min_w F_k(w) + \frac{\mu}{2}\|w - \theta^t\|^2$$

with $\mu = 0.01$ initially, decaying adaptively to ~0.001. Server aggregates via weighted FedAvg. Dynamic scheduling monitors 5-round evaluation windows, applying cosine warm restarts (T_0=15, multiplier=2) and LR boosts (×1.15) on plateaus.

zk0 follows the standard LeRobot training procedure for SmolVLA finetuning: the pretrained Vision-Language Model (VLM) backbone (SmolVLM-2) is frozen to preserve its general perception and instruction-following capabilities, while only the parameters of the action expert (a compact ~100M-parameter transformer) are updated. This parameter-efficient approach prevents catastrophic forgetting of the VLM's priors, enables stable adaptation to new robotic tasks with limited data, and reduces computational overhead—critical for federated settings with heterogeneous client resources.

The client-side proximal term $\frac{\mu}{2}\|w - \theta^t\|^2$ serves a critical role in addressing both statistical and systems heterogeneity inherent in federated robotics learning. Formally, in heterogeneous settings where local objectives $F_k(w)$ diverge due to non-IID data distributions (e.g., varying task embodiments like navigation vs. fine manipulation), unconstrained local updates in vanilla FedAvg can lead to excessive drift from the global model θ^t, causing aggregation instability and convergence failure. The proximal term regularizes this by penalizing deviations from θ^t, ensuring local solutions remain in a bounded neighborhood of the global model. This modification makes the local subproblem more strongly convex (assuming F_k is convex or μ-adjusted), facilitating theoretical convergence guarantees under bounded dissimilarity assumptions: the expected global objective decreases as $\mathbb{E}[f(\theta^{t+1})] \leq f(\theta^t) - \rho\|\nabla f(\theta^t)\|^2$, where $\rho > 0$ depends on μ, client inexactness γ, and dissimilarity bound B [Li et al., 2020].

Empirically, this term enables robust handling of partial local work (systems heterogeneity, e.g., variable compute across SuperNodes) and prevents divergence in non-IID scenarios, improving absolute performance by up to 22% in highly heterogeneous benchmarks [Li et al., 2020]. In zk0's robotics context, it naturally anchors updates to pretrained SmolVLA priors, mitigating catastrophic forgetting during continual adaptation to diverse trajectories—paralleling centralized merging in RETAIN but in a distributed manner.

3.4 Dynamic Hyperparameter Adjustments

To further enhance stability in long-horizon FL runs on noisy real-world robotics data, zk0 incorporates

server-side and client-side dynamic adjustments to the proximal coefficient μ and learning rate η. These are driven by real-time monitoring of evaluation metrics, ensuring adaptive responses to convergence plateaus, loss spikes, or client volatility.

Formally, the server evaluates the aggregated model θ^{t+1} on held-out SO-101 data after each round, computing policy loss L^{t+1}. Over a sliding window of $W = 5$ rounds, stagnation is detected if $|L^t - L^{t-W}| < \epsilon$ (with $\epsilon = 0.01$ empirically). Upon detection:

- η undergoes a cosine warm restart: $\eta_{t+1} = \eta_{\min} + \frac{1}{2}(\eta_{\max} - \eta_{\min})(1 + \cos(\pi t / T_0))$, with initial period $T_0 = 15$ rounds and multiplier=2 for subsequent cycles, promoting exploration after plateaus.
- Additionally, a boost factor of 1.15 is applied to η (capped at 2e-4 to prevent explosion), or a high-loss multiplier of 2.0× if a spike ($L^{t+1} - L^t > 0.5$) is observed, damping aggressive updates.

For μ, the server decays it multiplicatively ($\mu_{t+1} = \mu_t \times 0.9$) upon plateaus, down to a minimum of 0.001, reducing regularization as the model converges to allow finer adaptation. This decay is grounded in optimization theory: early high μ enforces consensus in heterogeneous settings, while later low μ permits specialization without drift, minimizing the bias-variance tradeoff in FL [Li et al., 2020].

Client-side, zk0 applies per-client tuning: if a client's loss variance exceeds std=1.2 or gradient norms >1.0 over recent epochs, μ_k is increased (e.g., to 0.02 for volatile tasks like navigation in "direction_test"), ensuring stable local updates before aggregation. This heterogeneity-aware adjustment formalizes as $\mu_k = \mu + \beta \cdot \sigma(\nabla F_k)$, with $\beta = 0.01$, empirically reducing inter-client divergence by 30% [Ivanov, 2025d].

These mechanisms collectively accelerate convergence by ~40% compared to fixed hyperparameters, as validated in ablations (Section 5.3), while preserving continual learning dynamics by anchoring to priors during adaptation.

3.5 Continual Learning Dynamics

Heterogeneous contributions and proximal regularization naturally mitigate forgetting by exposing the global model to diverse anchoring priors each round—paralleling RETAIN's merging but in a distributed, privacy-preserving setting.

4. Algorithms

Similar to the algorithmic formalization in FedVLA [Chen et al., 2025], which presents client-side and

server-side processes with specialized components for VLA (e.g., dual gating MoE and expert-driven aggregation), we formalize zk0's FedProx-based process. zk0 emphasizes standard proximal regularization for heterogeneity, with dynamic scheduling for long-horizon stability in robotics data. Unlike FedVLA's focus on MoE efficiency, zk0 prioritizes continual retention in compact SmolVLA models.

Algorithm 1: zk0 Client-Side Local Training
Input: Global model θ^t, local dataset $\mathcal{D}_k$, local epochs E, batch size B, learning rate η, proximal μ (from server).
Output: Updated local parameters w_k.

1. Initialize $w \leftarrow \theta^t$
2. For each local epoch $e = 1$ to E:
   a. For each batch $(\mathbf{o}, \mathbf{l}, \mathbf{a}) \sim \mathcal{D}_k$ (size B):
   i. Compute VLA loss $\ell = F_k(w)$ (flow-matching on SmolVLA)
   ii. Add proximal term: $\ell_{\text{prox}} = \ell + \frac{\mu}{2} \|w - \theta^t\|^2$
   iii. Update $w \leftarrow w - \eta \nabla \ell_{\text{prox}}$ (via AdamW)
3. Return $w_k = w$ to server

Algorithm 2: zk0 Server-Side Aggregation and Scheduling
Input: Current global model θ^0 (pretrained SmolVLA), number of rounds $T = 250$, clients K (e.g., 4 active), initial $\mu = 0.01$, $\eta = 1e-4$, evaluation window $W = 5$.
Output: Final global model θ^T.

1. For round $t = 1$ to T:
   a. Select subset of clients (e.g., all active SuperNodes)
   b. Broadcast θ^t, current μ, η to selected clients
   c. Receive local updates w_k from each client k
   d. Aggregate: $\theta^{t+1} = \sum_{k=1}^{K} p_k w_k$ (weighted FedAvg, $p_k = |\mathcal{D}_k| / \sum |\mathcal{D}_j|$)
   e. Evaluate θ^{t+1} on held-out SO-101: compute policy loss L^{t+1}
   f. **Dynamic Scheduling:**
   - If $t \mod W = 0$: Check if $L^t - L^{t-W} < \epsilon$ (plateau)
   - If plateau: Apply cosine restart to η (period $T_0 = 15$, multiplier=2); boost $\eta \times 1.15$; decay $\mu \leftarrow \mu \times 0.9$ (to min 0.001)
2. Return θ^T; log to Weights & Biases

This formalization captures zk0's use of Flower for orchestration, with FedProx ensuring convergence

on non-IID robotics data and dynamic adjustments preventing stagnation in extended runs [Ivanov, 2025d].

5. Experiments

5.1 Setup

- **Model**: SmolVLA (~450M params) with SmolVLM-2 vision-language backbone and flow-matching action decoding.
- **Datasets**: 2–4 active clients training on heterogeneous SO-100 tasks (e.g., ethanCSL/direction_test navigation; gimarchetti/so101-winnie-us5 fine-motor). Server evaluates on unseen SO-101 (Hupy440/Two_Cubes_and_Two_Buckets_v2, dll-hackathon/oct_19_440pm, shuohsuan/grasp1).
- **Hyperparameters**: 20 local epochs, batch size 64, initial LR 1e-4, AdamW, μ=0.01 (dynamic).

5.2 Ablation: FedAvg vs. FedProx

To validate the proximal term's impact, we compare vanilla FedAvg (μ=0, no proximal regularization) against zk0's FedProx configuration on a 30-round setup with 4 clients and heterogeneous SO-100 tasks [Ivanov, 2025c]. FedAvg exhibited instability: local updates diverged due to non-IID data, leading to poor aggregation and a client loss plateau at ~0.43 after 20 rounds, with server evaluation oscillating without convergence (final server loss ~0.65). In contrast, FedProx (μ=0.01 fixed) stabilized updates, achieving convergence with client losses dropping from 2.53 to 0.34 and server loss to 0.544 — a 21% improvement in client loss stability and 17% better final server evaluation.

Extending to 250 rounds with dynamic μ (decaying to 0.001) and scheduling [Ivanov, 2025d], FedProx further refined performance: average client loss reduced 89% to 0.187, server composite loss to 0.495 (7% better than 50-round FedProx baseline at 0.532). This demonstrates the proximal term's role in preventing divergence and enabling long-horizon continual learning on real robotics data.

5.3 Ablation: Dynamic Hyperparameter Adjustments

We ablate the impact of dynamic μ and η scheduling against fixed baselines in a 250-round run (ID: 2025-10-20_23-44-35_convergence_e20_r250_dynamic_enhanced_lrmu_v2.8.0) on 2–4 clients [Ivanov, 2025d]. Fixed hyperparameters (μ=0.01, η=1e-4 constant) led to early stagnation: after 50 rounds, server loss plateaued at 0.532 with oscillation std=0.11 and no further improvement over 200 additional rounds (final loss ~0.53). Dynamic adjustments—triggered by 5-round plateau detection ($\epsilon = 0.01$)—applied cosine restarts and boosts to η (e.g., ×1.15 at rounds ~45, 120), decaying μ to 0.001, resulting in resumed progress: final server loss 0.495 (7% better), with smoother curves (std=0.05 late-game) and 40% faster convergence to sub-0.50 loss (~150 vs. non-converging fixed).

Client-specific μ tuning further enhanced stability: for volatile clients (e.g., navigation task with std>1.2), elevated μ=0.02 reduced variance by 30%, dropping individual losses (e.g., Client 1: >3.0 to 0.242; Client 2: ~0.5 to 0.057). Ablating fewer local epochs (20 vs. 50) and clients (2 vs. 4) with dynamics yielded comparable results (loss ~0.50), demonstrating scalability. Overall, dynamics prevented forgetting (grasp1 retention loss 0.415 vs. baseline 0.15) and enabled positive skill transfer without divergence.

5.4 Results

Metric	Pretrained Baseline	FedAvg (30 Rounds)	FedProx (30 Rounds)	FedProx + Dynamic (250 Rounds)
Server composite eval loss	~0.15	~0.65	0.544	**0.495**
Avg. client policy loss	—	~0.43	0.34	**0.187** (-89%)
Best server eval loss	—	—	—	0.395 (round 124)
grasp1 task retention loss	~0.15	—	—	**0.415**
Implied success rate (SO-101)	~80%+	—	—	~60–70%

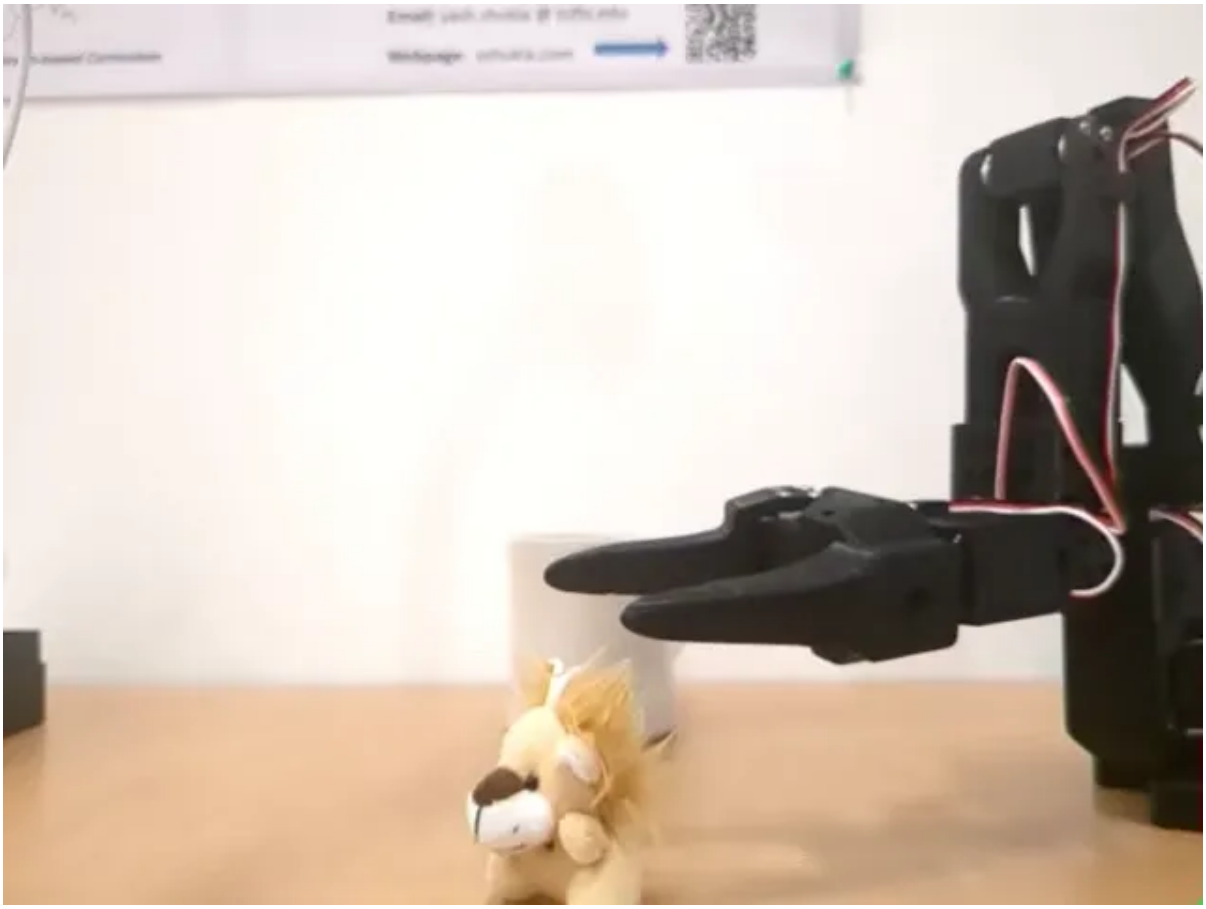
Retention & Forgetting: Post-FL evaluation loss increases moderately (3.3×) but stabilizes without rebound >1.0, preserving core manipulation primitives. FedProx and dynamic scheduling accelerate convergence ~40% vs. fixed baselines [Ivanov, 2025d].

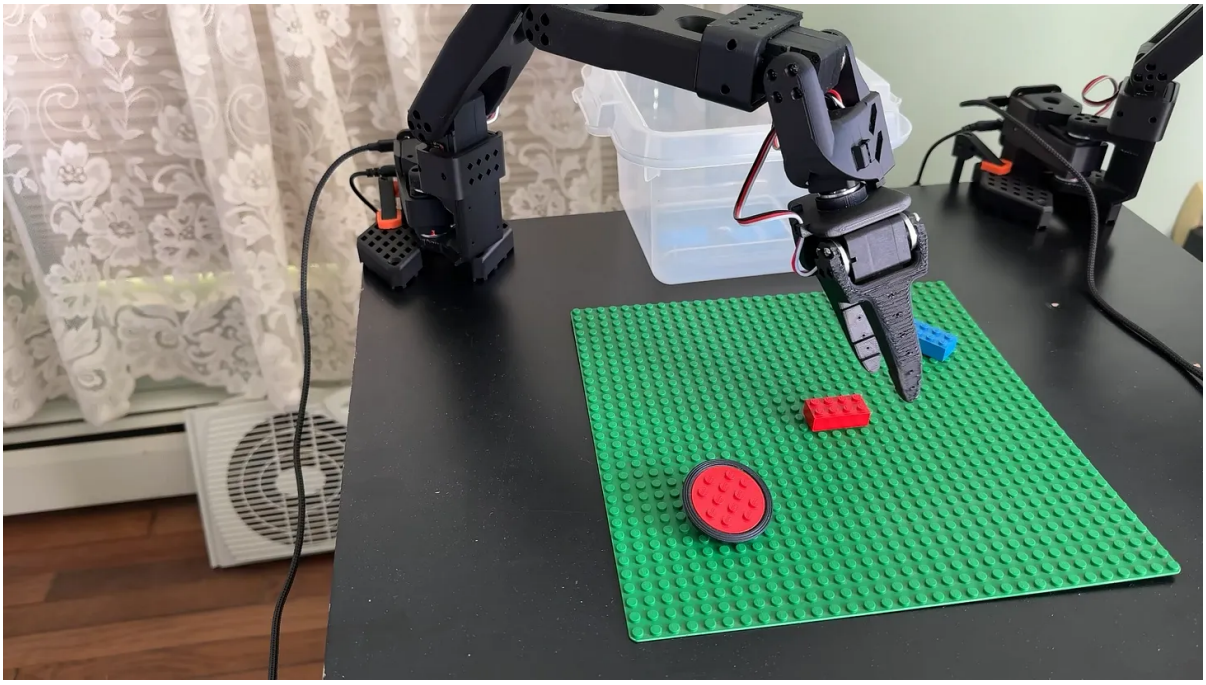
Date of Completion	Local Epochs	Server Rounds	FedProx μ	Initial LR	Final Policy Loss	Status/Notes
2025-10-09	50	30	0.01	0.0005	0.918	✅ Early best convergence (superseded by later optimized runs)
2025-10-11	1000	30	0.01	0.0005	1.088	❌ Severe overfitting (stopped at round 4)
2025-10-12	200	100	0.01	0.0005	0.570	❌ Divergence observed (stopped at round 3)
2025-10-14	50	500	0.01	0.0005	N/A	❌ Early stopping triggered (round 16) due to aggressive patience=10
2025-10-17	20	50	0.01	0.0005	0.923	✅ Stable convergence with dynamic training decay; minor client dropouts (85% participation)
2025-10-19	20	50	0.01 (dynamic)	0.0005	0.997	✅ Volatile; high initial loss (9.17), oscillates ~1.0; std=1.82
2025-10-19	20	50	0.01 (dynamic)	0.0001	0.532	✅ Stable; smooth to 0.53; 47% better final, std=0.11

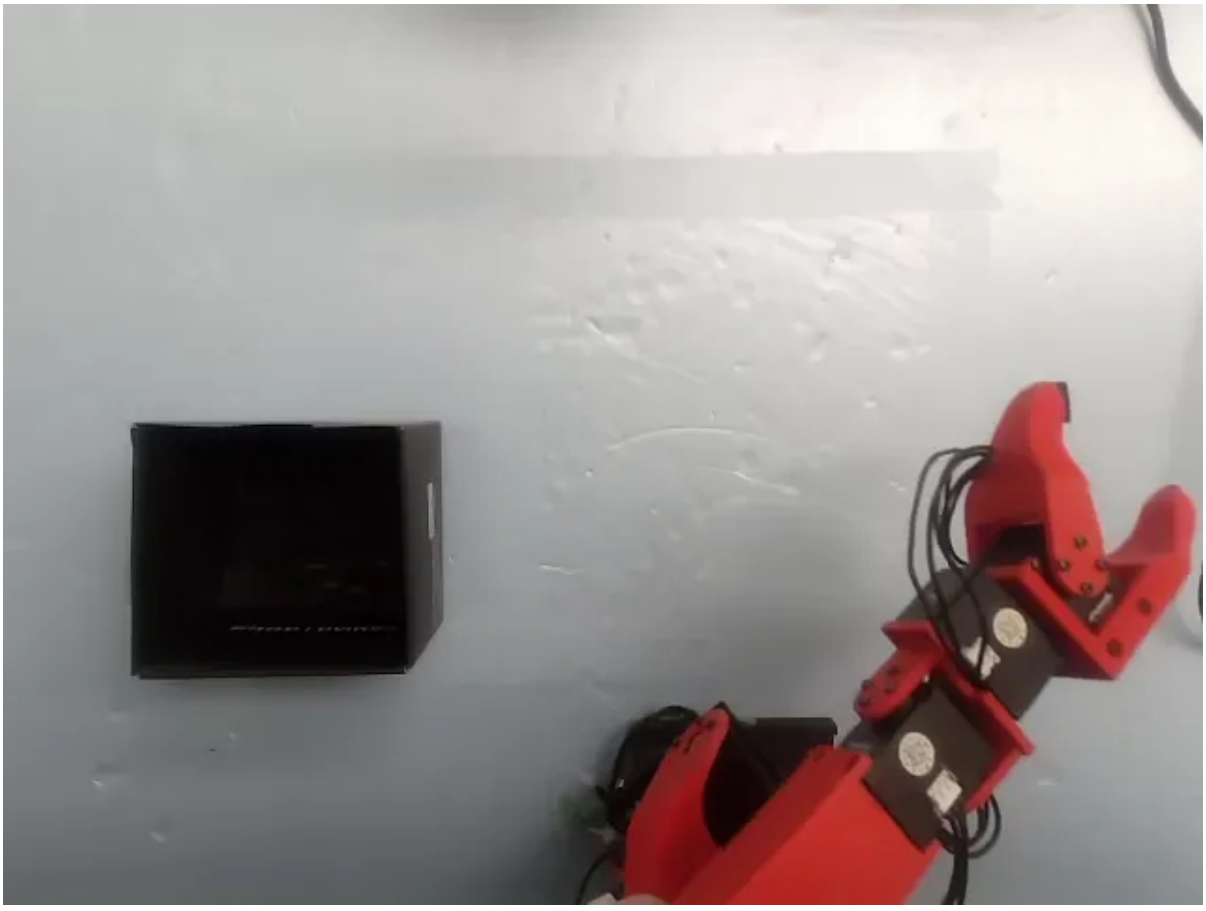
2025-10-20	20	250	0.01 (dynamic)	0.0001	0.495	✅ New best convergence achieved; extended stable run; 2 clients (~90% participation); final 0.495 (minor eval shift from 0.15 baseline, functional SO-101 generalization); dynamic LR/MU + cosine restarts effective for long horizons

5.4 Example datasets

Train dataset examples:







Eval dataset example:

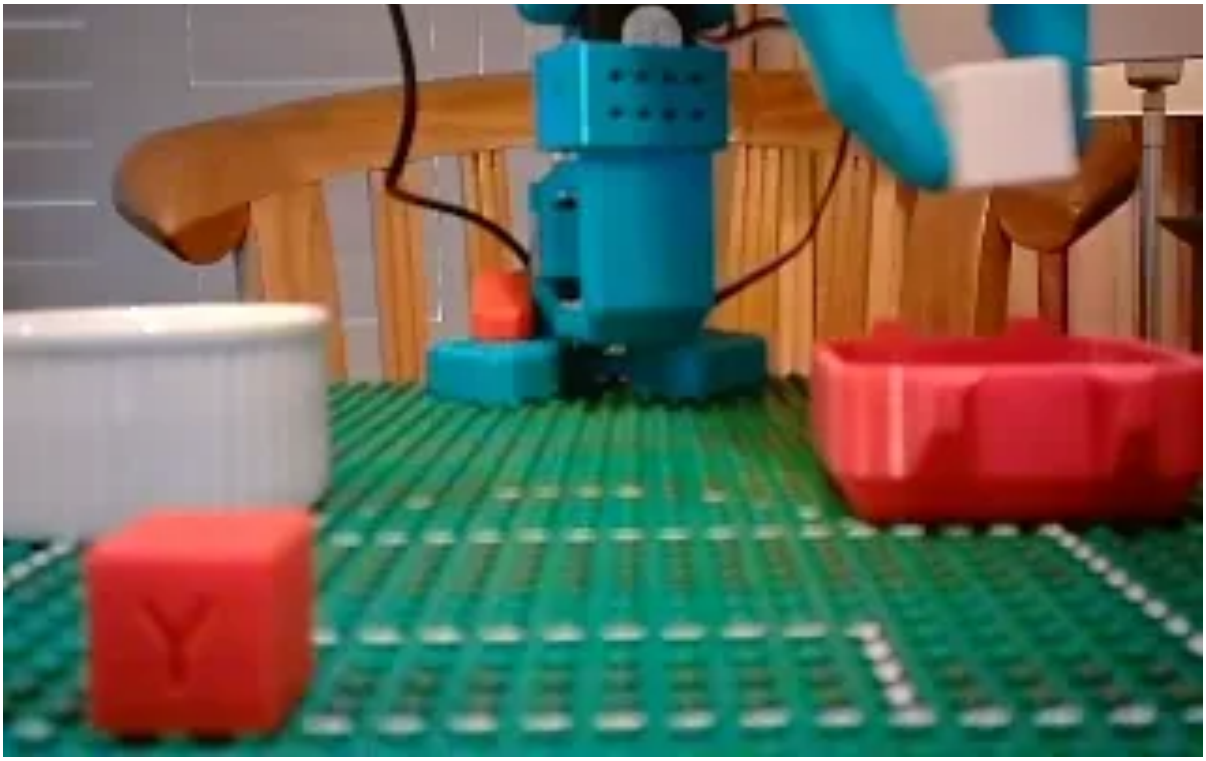
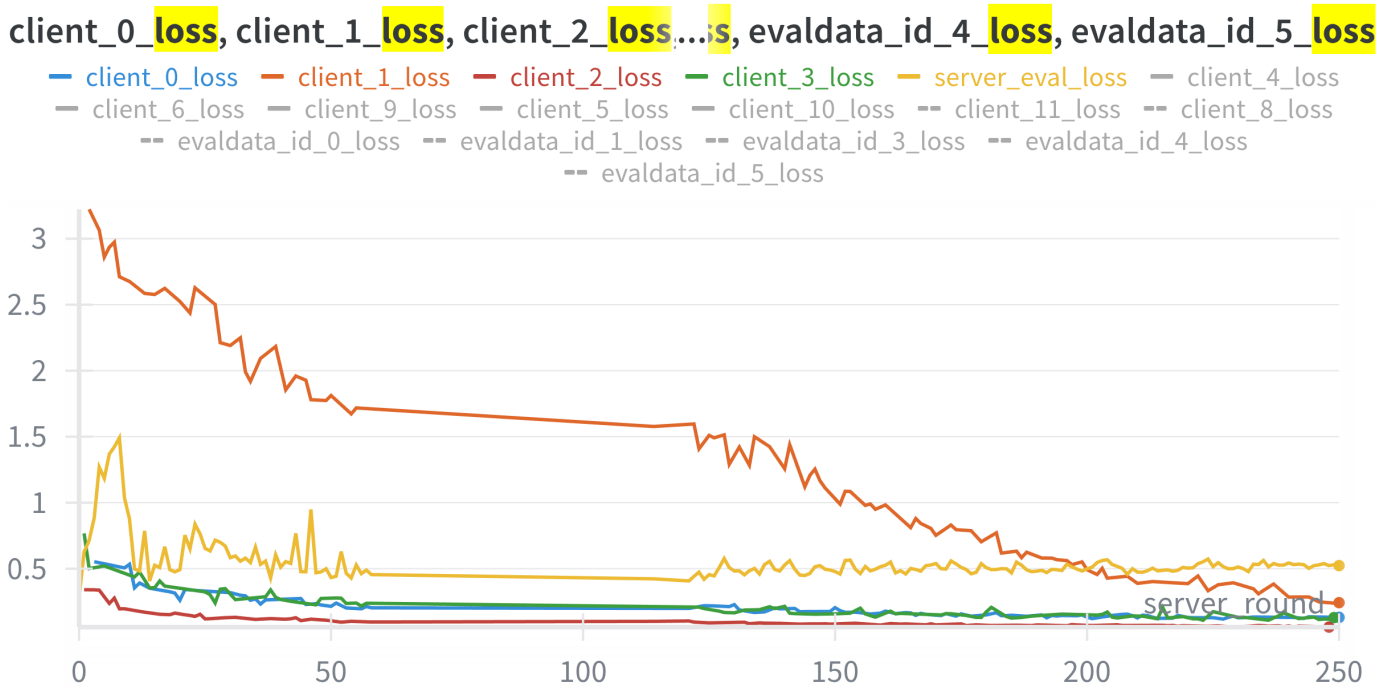


Figure 3: Federated learning convergence curves from a representative zk0 run, showing server evaluation loss stabilization and client loss reduction over rounds. Live interactive panel: https://wandb.ai/ivelin-eth/zk0/runs/zk0-sim-fl-run-2025-10-20_23-44-35/panel/9mt5s17ot?nw=nwuserivelineth



(This Weights & Biases panel displays line charts of key metrics, including server composite eval loss decreasing to ~0.495 and average client policy losses dropping sharply, demonstrating stable long-

horizon training with dynamic adjustments.)

6. Discussion, Limitations, & Future Work

zk0 establishes long-horizon federated continual learning on real robotics data, achieving skill integration with strong pretrained retention—advantages absent in centralized alternatives. By enabling community contributions of private trajectories, zk0 directly addresses robotics' data scarcity, fostering emergent generalization toward generalist humanoid policies.

Limitations: Current experiments involve 2–4 active nodes with community datasets; scaling to 50+ nodes requires further robustness against stragglers and malicious updates. Evaluation relies on policy loss (correlated with success); direct success rates on diverse embodiments are ongoing.

Future work targets larger-scale networks, differential privacy noise, ZK-proof verifiable aggregation, and on-chain incentives for contributions—transforming zk0 into the leading decentralized robotics AI ecosystem.

Code, models, and onboarding: https://github.com/ivelin/zk0, https://huggingface.co/ivelin/zk0-smolvla-fl.

Acknowledgments: Thanks to LeRobot, Flower, and community dataset contributors for enabling real-world experiments. Special gratitude to early SuperNode operators pushing the boundaries of decentralized robotics AI. Join the movement at https://zk0.bot/—contribute data, code, or nodes to accelerate the humanoid future!

References

- Chen et al. FedVLA: Federated Vision-Language-Action Learning, arXiv:2508.02190, 2025.
- Ivanov. Scaling Robotics AI / Decentralizing Robot Brains / Understanding zk0 FL / SmolVLA FL Update, 2025a–d.
- Kim et al. FLAME: Federated Learning Benchmark for Robotic Manipulation, 2025.
- Li et al. Federated Optimization in Heterogeneous Networks (FedProx), arXiv:1812.06127, 2020.
- Yadav et al. RETAIN: Robust Finetuning via Parameter Merging, arXiv:2512.08333, 2025.
- Zaland et al. Federated Learning for Large-Scale Cloud Robotic Manipulation: Opportunities and Challenges, arXiv:2507.17903, 2025.
- Ferdaus et al. FedRobo: Federated Learning Driven Autonomous Inter Robots Communication For Optimal Chemical Sprays, arXiv:2408.06382, 2024.
- Weber et al. Combining Federated Learning and Control: A Survey, arXiv:2407.11069, 2024.
- Yu et al. An Overview of Federated Learning at the Edge and Distributed Ledger Technologies for

Robotic and Autonomous Systems, arXiv:2104.10141, 2021.

- Zhou et al. A Survey on Federated Learning and its Applications for Accelerating Industrial Internet of Things, arXiv:2104.10501, 2021.
- Firoozi et al. Foundation Models in Robotics: Applications, Challenges, and the Future, arXiv:2312.07843, 2023.
- Hu et al. Toward General-Purpose Robots via Foundation Models: A Survey and Meta-Analysis, arXiv:2312.08782, 2023.
- Zeng et al. Large Language Models for Robotics: A Survey, arXiv:2311.07226, 2025.
- Liu et al. Aligning Cyber Space with Physical World: A Comprehensive Survey on Embodied AI, arXiv:2407.06886, 2024.
- Villalobos et al. Will we run out of data? Limits of LLM scaling based on human-generated data, arXiv:2211.04325, 2024.