FORGEVLA: FEDERATED VISION-LANGUAGE-ACTION LEARNING WITHOUT CLIENT-SIDE LANGUAGE ANNOTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-Language-Action (VLA) models hold great promise for general-purpose robotic intelligence, yet scaling up such models is severely bottlenecked by the high cost of acquiring annotated training data. Fortunately, vision-equipped robots deployed across various domains already produce abundant vision-action pairs that can be leveraged to scale up VLA training more efficiently. However, these raw data cannot be centrally aggregated due to various constraints and also exhibit severe heterogeneity. To address these challenges, in this paper, we propose ForgeVLA, a federated VLA training framework that learns VLA models from distributed vision-action pairs without centralizing client-side raw data or requiring client-side manual annotations. Specifically, each client in ForgeVLA is equipped with an embodied instruction classifier that maps vision-action trajectories to a predefined instruction set, recovering the missing language modality and forming complete vision-language-action triplets. Beyond triplet construction, we also identify vision-language feature collapse as a critical challenge that has been largely overlooked in prior federated VLA research. To mitigate this issue, ForgeVLA combines a client-side contrastive planning loss with a server-side adaptive aggregation strategy to learn task-discriminative representations efficiently. Extensive experiments across multiple benchmarks show that ForgeVLA significantly outperforms other baselines, and ablation studies further validate the contribution of each component.

1 INTRODUCTION

Vision-Language-Action (VLA) models integrate visual perception, language understanding, and motor control into a single policy, advancing toward general-purpose robotic intelligence (Driess et al., 2023; Kim et al., 2024; Black et al., 2025). As with LLMs and VLMs (Kaplan et al., 2020; Hoffmann et al., 2022; Muennighoff et al., 2023), VLA models benefit from larger and more heterogeneous training corpora, with stronger cross-embodiment transfer observed at greater data scale (Walke et al., 2023; Khazatsky et al., 2024; Open X-Embodiment Collaboration, 2024; Zheng et al., 2026). However, unlike text or image-text pairs that can be collected at near-zero marginal cost, VLA training samples typically require physical execution, making data collection a central bottleneck for scaling VLA capabilities (Dasari et al., 2019; Mandlekar et al., 2023).

Prior work has explored multiple directions to alleviate data scarcity in VLA training: data augmentation (Laskin et al., 2020; Kostrikov et al., 2021; Mandlekar et al., 2023) provides limited additional diversity; simulation-based pipelines (Tobin et al., 2017; James et al., 2020; Mu et al., 2021) are constrained by the reality gap; and generative models that synthesize visual trajectories (Du et al., 2023; Black et al., 2024; Ko et al., 2023) still struggle to produce physically plausible action labels. These approaches all expand the training corpus *synthetically*, raising a natural question: *can we source high-quality VLA training data at scale and low cost without synthetic generation*?

Historically, LLM and VLM scaling benefited from repurposing existing artifacts at low cost (Villalobos et al., 2022) (*e.g.*, decades of written text (Brown et al., 2020; Touvron et al., 2023)). A similar opportunity may exist in robotics (Kehoe et al., 2015) as well: millions of vision-equipped robots deployed in manufacturing (International Federation of Robotics, 2025), warehousing (Ama-

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.

zon, 2025), healthcare (Intuitive Surgical, Inc., 2025), autonomous driving (Sun et al., 2020), and other domains (Duckett et al., 2018; Abolhasani & Kumacheva, 2023; Wynn et al., 2014) could already log synchronized visual observations and action trajectories suitable for VLA training. The missing modality is natural-language task description, which is typically unnecessary at deployment time and difficult to recover post hoc because a continuous action stream admits multiple valid descriptions at different granularities. Thus, the bottleneck for scaling VLA may largely reduce to the absence of language annotations for otherwise valuable vision–action logs.

Closing this annotation gap at scale faces two tightly coupled challenges: (1) *Privacy*: robotic data is siloed within independent organizations (*e.g.*, factories and hospitals) under strict confidentiality constraints. (2) *Heterogeneity*: clients differ in embodiment, task distributions, sensing conditions, and environments, inducing severely non-independent and identically distributed (non-i.i.d.) data.

Federated learning (FL) (McMahan et al., 2017; Kairouz et al., 2021) provides a natural framework for this setting, allowing clients to exchange only model updates. However, existing federated VLA methods leave two issues unresolved: (1) *Data*: FedVLA (Cui et al., 2025) and FLAME (Bou Betran et al., 2025) assume fully annotated VLA triplets at each client, offering no mechanism to incorporate the far more abundant unannotated vision–action logs. (2) *Heterogeneity*: we empirically find that non-i.i.d. mitigations (*e.g.*, FedProx (Li et al., 2020)) do not transfer to VLA training and can even degrade performance (Table 2), suggesting a failure mode beyond conventional client drift.

To address these challenges, we propose **ForgeVLA**, a federated framework that repurposes legacy vision–action logs for VLA training without centralizing raw data or manually annotating task instructions. On the data side, ForgeVLA equips each client with an *embodied instruction classifier*, a pretrained VLM fine-tuned on a small public VLA dataset, that maps local vision-action trajectories to a predefined instruction set entirely on-device, recovering the missing language modality. On the heterogeneity side, we trace the root cause to vision-language feature collapse and introduce (i) a contrastive planning loss that promotes task-discriminative representations during local training, and (ii) an adaptive aggregation strategy that preserves client update directions on the server. In summary, our main contributions are:

1. We identify the installed base of vision-equipped robots as a largely untapped opportunity to scale VLA training beyond curated or synthetic datasets. Because privacy constraints preclude centralized access, we propose ForgeVLA, a federated paradigm that recovers the missing language modality on-device via classification over a predefined instruction set and converts distributed legacy logs into usable VLA training data.
2. We show that prior methods fail to address heterogeneity in federated VLA training, and we trace the root cause to vision-language feature collapse under task-distribution heterogeneity across federated clients. Motivated by this finding, we mitigate collapse with a contrastive planning loss for task-discriminative local representation learning and an adaptive aggregation strategy on the server.
3. Extensive experiments show that ForgeVLA significantly outperforms federated baselines across multiple benchmarks. Source code will be released after publication.

Scope. While the motivating deployment spans organizations, embodiments, and environments, this paper isolates one central axis: task-distribution heterogeneity across clients, with embodiment and environment held fixed. Evaluation across independent organizations, multiple embodiments, and diverse environments is left to future work.

2 Related Work

Vision-Language-Action Models: VLA models unify visual perception, language understanding, and action prediction for generalist robot control. RT-2 pioneered large-backbone VLA policies (Zitkovich et al., 2023), while OpenVLA and OpenVLA-OFT improved accessibility, performance, and efficiency (Kim et al., 2024; 2025). Subsequent work advances dexterous control, spatial reasoning, action modeling, and training frameworks (Black et al., 2025; Li et al., 2024b; Qu et al., 2025; Chen et al., 2025; Wen et al., 2025), with complementary efforts targeting cross-embodiment generalization (Octo Model Team et al., 2024; Doshi et al., 2024; Wang et al., 2024). These methods nevertheless depend heavily on curated language annotations, limiting scalable data acquisition.

Federated Vision-Language-Action Learning: Federated learning enables collaborative training through local optimization and server-side aggregation without centralizing raw data (McMahan et al., 2017; Zhou et al., 2021; 2023; 2025b). Non-i.i.d. client data can destabilize training, motivating methods based on proximal regularization, control variates, normalized aggregation, and personalized priors (Li et al., 2020; Karimireddy et al., 2020; Wang et al., 2020; Acar et al., 2021; Shi et al., 2023a; Zhou et al., 2024; 2025a). Secure aggregation and differential privacy can strengthen data protection (Bonawitz et al., 2017; Abadi et al., 2016), whereas unprotected updates may leak training information (Zhu et al., 2019; Geiping et al., 2020). For robotic learning, FLAME provides a federated manipulation benchmark (Bou Betran et al., 2025), and FedVLA introduces instruction-aware expert routing and aggregation (Cui et al., 2025). However, both assume annotated VLA triplets. ForgeVLA instead recovers instructions from local vision-action logs and directly addresses heterogeneous vision-language feature collapse. Additional discussion is provided in Appendix A.

3 Problem Formulation

We consider a standard FL setup consisting of one central server and N participating clients, indexed by $i \in [N] \triangleq \{1, 2, \ldots, N\}$. Each client i holds a private local dataset D_i containing K_i time-synchronized vision–action pairs, collected from its embodied robotic system:

$$D_i = \left\{(v_i^k, a_i^k)\right\}_{k=1}^{K_i}, \tag{1}$$

where $v_i^k \in \mathcal{V}$ denotes a visual observation sampled from the vision space $\mathcal{V}$, and $a_i^k \in \mathcal{A}$ denotes a corresponding robot action sampled from the action space $\mathcal{A}$. These pairs are grouped into Q_i trajectories, each recording one execution of a task. We write x_i^q for the q-th trajectory of client i and $q(k) \in [Q_i]$ for the index of the trajectory containing the k-th pair, so that $K_i = \sum_{q=1}^{Q_i} |x_i^q|$. A defining characteristic of our problem is that the local dataset D_i lacks language instructions $l_i^k \in \mathcal{I}$, because task descriptions are unnecessary at deployment time and remain ambiguous during data collection, where $\mathcal{I} = \{l_m\}_{m=1}^{M}$ is the instruction space with M distinct instructions in total. Throughout the paper we reserve $\mathcal{I}$ for the instruction space and $\mathcal{L}$ for loss functions.

The global VLA model follows an encoder-decoder architecture, parameterized by $\theta = (\theta_{\text{enc}}, \theta_{\text{dec}}) \in \Theta_{\text{vla}}$. The vision-language encoder $f_{\theta_{\text{enc}}} : \mathcal{V} \times \mathcal{I} \to \mathcal{Z}$ maps a visual observation and a natural language instruction to a joint latent representation $z \in \mathcal{Z}$, where $\mathcal{Z}$ is the latent embedding space with dimension d. The action decoder $f_{\theta_{\text{dec}}} : \mathcal{Z} \to \mathcal{A}$ takes the joint latent representation z and predicts the corresponding robot action. The overall VLA mapping is thus the composition: $f_\theta(v, l) = f_{\theta_{\text{dec}}}(f_{\theta_{\text{enc}}}(v, l))$. To bridge the missing language modality locally, each client i owns a pretrained instruction classifier c_ϕ, parameterized by $\phi \in \Phi$. A single visual observation paired with a single action is generally insufficient to identify the underlying task, so the classifier consumes an entire trajectory rather than an isolated pair. Let $\mathcal{X} \triangleq \bigcup_{T \geq 1} (\mathcal{V} \times \mathcal{A})^T$ denote the space of vision–action trajectories of arbitrary length. The classifier $c_\phi : \mathcal{X} \to \mathcal{I}$ then predicts one plausible language instruction per trajectory, and every pair inherits the instruction of the trajectory it belongs to, *i.e.*, $\hat{l}_i^k = c_\phi(x_i^{q(k)})$. Let $\ell(\cdot, \cdot)$ denote a task-specific loss function, the local empirical loss on client i can be defined as:

$$\mathcal{L}_i(\theta; D_i, \phi) = \frac{1}{K_i} \sum_{k=1}^{K_i} \ell\left(f_{\theta_{\text{dec}}}\left(f_{\theta_{\text{enc}}}\left(v_i^k, c_\phi(x_i^{q(k)})\right)\right), a_i^k\right). \tag{2}$$

Then, the objective of federated VLA learning is to find the optimal global VLA parameters $\theta = (\theta_{\text{enc}}, \theta_{\text{dec}})$ that minimize the weighted sum of all local empirical losses, while adhering to the aforementioned privacy and heterogeneity constraints:

$$\arg\min_{\theta \in \Theta_{\text{vla}}} \sum_{i=1}^{N} w_i \cdot \mathcal{L}_i(\theta; D_i, \phi), \tag{3}$$

where $w_i \geq 0$ are client-specific weights satisfying $\sum_{i=1}^{N} w_i = 1$ that help combine clients' local losses into a global optimization target. The most widely adopted w_i is proportional to the local dataset size, *i.e.*, $w_i = K_i / \sum_{j=1}^{N} K_j$.

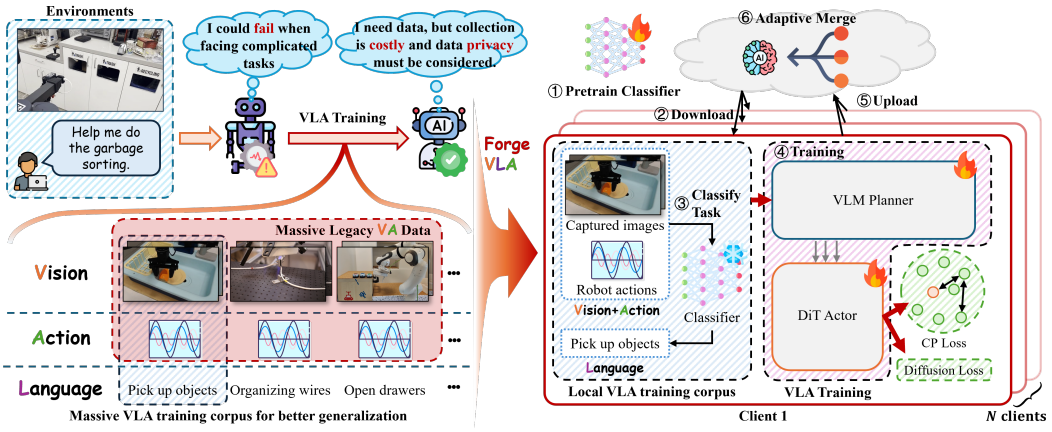


Figure 1: **[Left]**: The key bottleneck for scalable VLA training is data scarcity, as high-quality annotated VLA data are limited, while large volumes of vision–action logs remain underutilized. **[Right]**: ForgeVLA across N clients, enabling scalable VLA training from real-world robot logs at near-zero marginal cost while preserving privacy and mitigating data heterogeneity.

4 FORGEVLA

Figure 1 overviews the ForgeVLA architecture. On the server, we fine-tune a pretrained VLM on a small public VLA dataset to obtain an embodied instruction classifier c_ϕ. The server then broadcasts c_ϕ together with the initialized global VLA model f_θ to all clients. On each client, c_ϕ classifies local vision–action logs into the predefined instruction set, forming a VLA training corpus while keeping all raw data on-device to preserve privacy. The client then performs local VLA training with a contrastive planning loss to counteract heterogeneity-induced degradation. After local training, clients upload their model updates together with one mean representation per local task, and the server applies adaptive aggregation to reduce cross-client update conflicts. Overall, ForgeVLA addresses three bottlenecks for scaling VLA: leveraging underutilized real-world vision–action data, avoiding centralized collection of sensitive logs, and mitigating client heterogeneity.

4.1 EMBODIED INSTRUCTION CLASSIFIER

To recover the missing task description in raw vision–action logs and construct a valid local VLA training corpus, we propose an embodied instruction classifier built on a pretrained VLM. The classifier takes a whole trajectory as input and operates over a predefined set of M task instructions, which is a practical design choice that covers the structured deployment scenarios common in industrial and service robotics. Figure 2 summarizes the design: we augment the VLM backbone with a lightweight attention-pooling head and an instruction classification layer. Given a vision–action trajectory x_i^q, the pretrained VLM produces final-layer hidden states over the whole trajectory. The attention-pooling module computes weights over these states and aggregates them along the temporal axis into a single context vector via weighted summation. The classifier then maps the context vector to logits over the instruction set. Leveraging the pretrained VLM's strong generalization, this classifier requires only lightweight fine-tuning on a small public VLA dataset (as shown in Section 6), enabling efficient and robust on-device annotation.

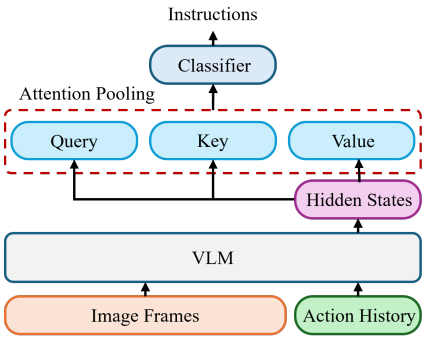
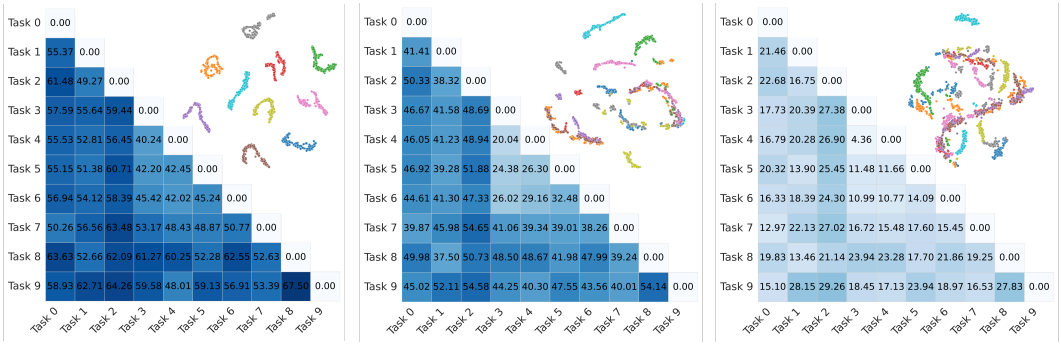


Figure 2: The Instruction Classifier.

4.2 VISION-LANGUAGE FEATURE COLLAPSE

Real-world robotic deployments exhibit pronounced cross-client heterogeneity. For example, manufacturing robots operating in different factories often share only a small subset of overlapping tasks,



(a) Centralized (b) FedAvg (c) FedProx

Figure 3: The illustrated distances and t-SNE (Van der Maaten & Hinton, 2008) projections of latent feature representations learned by different models across different tasks. All representations are computed with ground-truth task instructions.

yielding highly skewed and largely disjoint task distributions. This issue is further amplified in FL because privacy constraints preclude centralizing raw data to smooth such heterogeneity.

Although numerous FL methods target non-i.i.d. data, we observe that directly applying them to federated VLA can severely degrade performance and even destabilize training, even when each client performs only a single local optimization step (*i.e.*, classical client drift is largely eliminated). In preliminary experiments (Table 2), FedProx (Li et al., 2020) performs substantially worse than vanilla FedAvg (McMahan et al., 2017) and sometimes fails to converge, indicating that existing non-i.i.d. remedies are insufficient for federated VLA.

To diagnose the cause, we conduct a systematic analysis on LIBERO-Goal (Liu et al., 2023). Specifically, we compare three models: (i) a centralized upper-bound model trained on aggregated data directly, (ii) FedAvg with each client holding data from three tasks, and (iii) FedProx under the same client partition. We then extract latent feature representations and visualize their pairwise distances across tasks. As shown in Figure 3, relative to centralized training, both FedAvg and FedProx yield tightly clustered task embeddings with small inter-task margins, indicating reduced ability to discriminate among distinct manipulation goals. FedProx exacerbates this effect, consistent with FedProx's proximal term pulling local models toward a global model whose representations are already collapsed. These results reveal an underexplored failure mode in federated VLA: *vision–language feature collapse*, where heterogeneous federated training causes task-specific vision-language representations to lose discriminability. Notably, this mode is distinct from the conventional client drift studied in prior FL work, because it persists even with a single local step (*i.e.*, no client drift). Appendix I repeats this analysis for ForgeVLA under an identical protocol, showing that the proposed components restore inter-task margins. To mitigate this collapse, we propose a dual-strategy framework with complementary client-side and server-side components:

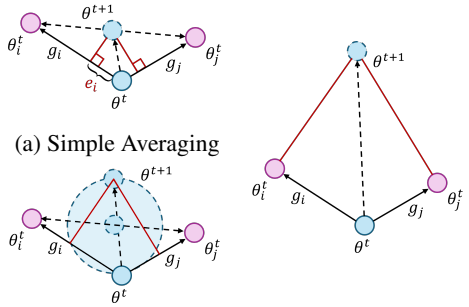
Contrastive Planning Loss: A direct way to mitigate vision–language feature collapse is to enforce discriminative margins between task representations in the training objective. To this end, we introduce a contrastive planning loss built on a global task representation bank maintained by the central server, which stores one entry per instruction. At the start of communication round e, the server broadcasts the current normalized bank $U^e = \{\hat{u}_1^e, \hat{u}_2^e, \ldots, \hat{u}_M^e\}$ to all clients, providing consistent anchors for local contrastive learning. For each local sample (v_i^k, a_i^k), let $\hat{l}_i^k = l_{t_i^k}$ denote the instruction predicted by c_ϕ, let $z_i^k = f_{\theta_{\text{enc}}}(v_i^k, \hat{l}_i^k)$, and let $\hat{z}_i^k = z_i^k / \|z_i^k\|_2$. During local training, client i computes the contrastive planning loss as follows:

$$\mathcal{L}_{\text{CP},i}^e = -\frac{\alpha_{\text{CP}}}{K_i} \sum_{k=1}^{K_i} \left(\frac{(\hat{z}_i^k)^\top \hat{u}_{t_i^k}^e}{\tau} - \log \sum_{m=1}^{M} \exp\left(\frac{(\hat{z}_i^k)^\top \hat{u}_m^e}{\tau} \right) \right), \tag{4}$$

where $t_i^k \in [M]$ is the instruction index associated with $\hat{l}_i^k$, τ is a fixed temperature hyperparameter, and each $\hat{u}_m^e$ is ℓ_2-normalized. Finally, $\mathcal{L}_{\text{CP},i}^e$ is added to the local VLA training objective. After local training, each client computes the updated mean latent representation $u_{i,m}^e$ for each of its local tasks m and uploads these task embeddings to the server. For every task that receives at least one upload,

the server forms the weighted aggregate $\tilde{u}_m^{e+1}$, normalizes it into the refreshed bank entry $\hat{u}_m^{e+1}$, keeps the previous entry for any task without a valid aggregate, and broadcasts the refreshed bank in the next communication round. This closed loop maintains discriminative task representations throughout federated training.

Adaptive Aggregation Strategy: As shown in Figure 4 (a), simple averaging of heterogeneous client updates can cancel conflicting directions, producing a global update whose projection onto each client's update direction is substantially attenuated. This cancellation slows convergence and degrades performance. Based on this observation, we propose a server-side adaptive aggregation strategy that solves a data-free optimization problem using only client updates. At communication round e, let $S^e \subseteq [N]$ denote the clients that upload local models, let $\theta_i^{e,P}$ denote client i's model after P local optimization steps, let $g_i^e = \theta_i^{e,P} - \theta^e$ denote the uploaded update, and let $\tilde{w}_i^e$ denote the server-side weight assigned to an uploaded update. Here P counts mini-batch steps, and the experiments instead specify the local budget in epochs, denoted by P_{ep} (Appendix E). We encourage the global update $\theta^{e+1} - \theta^e$ to match each uploaded client update g_i^e in projection by minimizing the deviation of the projection coefficient from 1, *i.e.*,



(a) Simple Averaging

(b) Constrained Adaptive Aggregation

(c) Unconstrained Adaptive Aggregation

Figure 4: Illustrations of Aggregations.

$$\arg\min_{\theta^{e+1}} \sum_{i \in S^e} \tilde{w}_i^e \left(\frac{(\theta^{e+1} - \theta^e)^\top g_i^e}{\|g_i^e\|_2^2 + \varepsilon_{\text{AG}}} - 1 \right)^2, \tag{5}$$

where $\varepsilon_{\text{AG}} > 0$ is a small numerical stabilizer when a client update is nearly zero. While this objective promotes per-client update preservation, it can become unstable when client updates are highly conflicting (Figure 4 (c)). For example, if two clients' updates are nearly anti-parallel, the unconstrained objective may drive $\|\theta^{e+1} - \theta^e\|_2$ to grow without bound. To stabilize training, we regularize the solution by anchoring θ^{e+1} to the sampled weighted average of the uploaded client models, yielding the final objective (Figure 4 (b)):

$$\arg\min_{\theta^{e+1}} \sum_{i \in S^e} \tilde{w}_i^e \left(\frac{(\theta^{e+1} - \theta^e)^\top g_i^e}{\|g_i^e\|_2^2 + \varepsilon_{\text{AG}}} - 1 \right)^2 + \alpha_{\text{AG}} \left\| \theta^{e+1} - \left(\theta^e + \sum_{i \in S^e} \tilde{w}_i^e g_i^e \right) \right\|_2^2, \tag{6}$$

where α_{AG} balances projection alignment and update regularization. Note that in the convergence analysis presented in the appendix, these server-side weights are defined as unbiased importance weights under partial client participation.

5 Algorithm, Complexity, and Privacy

ForgeVLA adds lightweight instruction recovery and representation alignment to the standard federated workflow. Specifically, the server trains the embodied instruction classifier once on a small labeled public dataset and broadcasts it to clients, while distributing the current global VLA model and task representation bank in each communication round. Each client uses the classifier only for local inference, optimizes the VLA task loss together with the contrastive planning loss, and uploads its model update and one mean representation per local task. The server then updates the representation bank and performs adaptive aggregation (Appendix B). These operations preserve FedAvg's $\mathcal{O}(NEP)$ asymptotic training complexity, while the global bank requires $\mathcal{O}(Md)$ storage for M task representations of dimension d (Appendix C). On the privacy side, ForgeVLA follows the standard federated data-locality protocol: raw client trajectories remain on-device, and communication is limited to model updates and task-level aggregated representations. Established mechanisms such as secure aggregation (Bonawitz et al., 2017) and differential privacy (Abadi et al., 2016) provide natural extensions for further strengthening privacy protection. Incorporating these mechanisms and evaluating their privacy-utility trade-offs are promising directions for future work (Appendix J).

Table 1: The performance comparisons between different methods. Default settings are marked in gray. **bold** marks the best-performing results.

Methods	Datasets	# Params (M)	# Trainable Params (M)	Success Rate (%)	Pass@50 (%)
Centralized	LIBERO-Goal	3882.72	128.10	75.8	100
FedAvg	LIBERO-Goal	3882.72	128.10	28.8	80
FedVLA*	LIBERO-Goal	79.23	79.23	0.2	10
FedVLA* + CLIP	LIBERO-Goal	519.82	92.90	6.2	20
FedVLA* + MetaCLIP-G	LIBERO-Goal	2510.68	139.41	10.4	20
ForgeVLA	LIBERO-Goal	3882.72	128.10	**55.2**↑26.4%	100
Centralized	LIBERO-Object	3882.72	128.10	98.8	100
FedAvg	LIBERO-Object	3882.72	128.10	97.6	100
FedVLA*	LIBERO-Object	79.23	79.23	2.2	10
FedVLA* + CLIP	LIBERO-Object	519.82	92.90	18.2	30
FedVLA* + MetaCLIP-G	LIBERO-Object	2510.68	139.41	28.2	40
ForgeVLA	LIBERO-Object	3882.72	128.10	**98.6**↑1.0%	100
Centralized	LIBERO-Spatial	3882.72	128.10	85.8	100
FedAvg	LIBERO-Spatial	3882.72	128.10	68.6	90
FedVLA*	LIBERO-Spatial	79.23	79.23	0.4	10
FedVLA* + CLIP	LIBERO-Spatial	519.82	92.90	11.6	20
FedVLA* + MetaCLIP-G	LIBERO-Spatial	2510.68	139.41	16.6	30
ForgeVLA	LIBERO-Spatial	3882.72	128.10	**72.6**↑4.0%	100
Centralized	LIBERO-10	3882.72	128.10	79.0	100
FedAvg	LIBERO-10	3882.72	128.10	52.8	100
FedVLA*	LIBERO-10	79.23	79.23	0.4	10
FedVLA* + CLIP	LIBERO-10	519.82	92.90	9.4	20
FedVLA* + MetaCLIP-G	LIBERO-10	2510.68	139.41	15.8	30
ForgeVLA	LIBERO-10	3882.72	128.10	**63.6**↑10.8%	100

6 EXPERIMENTS

Experimental Setup: We evaluate ForgeVLA on four LIBERO suites: LIBERO-Goal, LIBERO-Object, LIBERO-Spatial, and LIBERO-10 (Liu et al., 2023). Following standard federated evaluation protocols (Li et al., 2020; Yi et al., 2024; Li et al., 2024a), we construct task-level non-i.i.d. partitions by assigning $s = 3$ tasks to each of $N = 10$ fully participating clients. Local client datasets retain only vision-action pairs, while the instruction classifier is trained separately on a labeled public subset comprising a fraction p of the original annotated data. We use InternVLA-M1 (Chen et al., 2025) as the backbone, applying LoRA (Hu et al., 2022) with rank $r_{\text{vla}} = 32$ to the VLM encoder and fully fine-tuning the action decoder. The instruction classifier is adapted with a separate LoRA rank r_{cls}, and we write r_{vla} and r_{cls} throughout to keep the two ranks distinct. Unless otherwise specified, training runs for $E = 20$ communication rounds with $P_{\text{ep}} = 5$ local epochs per round and a batch size of 32.

Baselines and Evaluation: We mainly compare ForgeVLA with a centralized oracle, the same-backbone FedAvg (McMahan et al., 2017) and FedProx (Li et al., 2020) baselines, and FedVLA* variants (Cui et al., 2025), where * denotes a community implementation. We report total and trainable parameter counts, task success rate, and Pass@50 over 50 rollouts per task.

Complete dataset partitioning, baseline implementations, optimization hyperparameters, compute resources, and evaluation protocols are provided in Appendix G.

Table 2: Representative non-i.i.d. FL methods result in even worse performance than vanilla FedAvg.

Methods	λ	Success Rate (%)
LIBERO-Goal		
FedAvg	0.0	28.8
FedProto	1.0	29.4
FedProx	0.1	11.2
FedProx	0.2	8.4
FedProx	0.5	4.2
MOON	1.0	20.2
ForgeVLA	0.0	**55.2**

Main Results: Table 1 presents the main performance comparisons. ForgeVLA achieves the best performance among all federated methods on every suite while using the same parameter budget as FedAvg, and substantially narrows the gap to the centralized upper bound. On LIBERO-Goal it reaches 55.2%, a 26.4 pp improvement over FedAvg, whereas existing federated VLA methods stay below 30% even after we equip FedVLA* with MetaCLIP-G (Xu et al., 2024)

to match its trainable-parameter budget (Appendix G discusses the remaining backbone capacity difference). On LIBERO-Object it reaches 98.6%, nearly matching the 98.8% centralized upper bound, and it attains a perfect Pass@50 of 100% on all four suites.

Limitations of Non-i.i.d. FL Methods: We conduct experiments on LIBERO-Goal to evaluate MOON (Li et al., 2021), FedProto (Tan et al., 2022), and FedProx (Li et al., 2020) in Table 2, where λ denotes each baseline's own regularization intensity, that is, the proximal weight for FedProx and the contrastive or prototype weight for MOON and FedProto. From the table, we observe that FedProx exhibits severe performance degradation as λ increases. Specifically, FedProx degrades monotonically as λ grows, falling from 28.8% at $\lambda = 0$ to 4.2% at $\lambda = 0.5$, and the two representation-alignment baselines also fall short, with MOON at 20.2% and FedProto at 29.4% versus 55.2% for ForgeVLA. That neither model-level contrastive learning nor prototype exchange closes the gap corroborates that the failure mode is task-level vision-language feature collapse rather than conventional client drift. This directly supports our claim that classical client-drift mitigation designed for standard non-i.i.d. FL is insufficient for federated VLA (Section 4.2), and validates ForgeVLA's targeted design.

Data & Model Scale Analysis for the Classifier: We evaluate our instruction classifier across varying data scales and model capacities on LIBERO-Goal, with results shown in Table 3. Fixed at classifier LoRA rank $r_{\text{cls}} = 4$, the classifier retains over 92% accuracy with 20% of full training data ($p = 0.2$), and maintains 76.68% accuracy in the most challenging one-data-per-task (ODPT) setting. In the ODPT setting, increasing r_{cls} consistently improves accuracy, with performance saturating at $r_{\text{cls}} = 16$ (85.56%).

Table 3: Data & model scales for the classifier.

r_{cls}	p	# Trainable (M)	Accuracy (%)	ForgeVLA (%)
LIBERO-Goal				
-	-	-	100	58.4
4	0.9	3.36	98.34	57.8
4	0.5	3.36	97.96	-
4	0.2	3.36	92.30	55.2
4	0.1	3.36	83.56	-
4	ODPT	3.36	76.68	44.2
8	ODPT	4.61	83.34	-
16	ODPT	7.10	85.56	-
32	ODPT	12.10	85.67	-

Ablation Study: We conduct an ablation study on LIBERO-Goal to validate the contribution of our two core components in Table 4. As can be seen, removing both components recovers the vanilla FedAvg baseline (28.8%). Adding only the CP loss improves performance to 36.2%, demonstrating its effectiveness in preserving task-specific vision-language alignment. Additionally, adding only the adaptive aggregation yields a larger improvement to 49.0%. Combining both components achieves the full performance of ForgeVLA at 55.2%.

Table 4: Ablation Study of Components in ForgeVLA

CP Loss	Adaptive Aggregation	Success Rate (%)
		28.8
✓		36.2
	✓	49.0
✓	✓	55.2

Sensitivity Analysis: We conduct a sensitivity analysis on four key hyperparameters of ForgeVLA, with results shown in Table 5. First, for the number of tasks per client s, both FedAvg and ForgeVLA improve as s increases due to richer and more diverse local data, but ForgeVLA still significantly outperforms FedAvg across all s, especially in highly heterogeneous settings (*e.g.*, 2% vs 0.8% at $s = 1$, 41.6% vs 17.6% at $s = 3$). Second, for the VLM encoder LoRA rank r_{vla}, ForgeVLA's performance saturates at the default $r_{\text{vla}} = 32$ with only mild gains from larger r_{vla}, and it outperforms FedAvg at all ranks as well, showing no reliance on excessive trainable parameters. Third, for α_{CP}, performance first rises then falls, peaking at the default 0.2, indicating moderate contrastive planning loss preserves task specificity. Fourth, for α_{AG}, performance follows a similar trend, peaking at the default 0.1. Overall,

Table 6: Real-world success rates (%) on two SO-ARM101 tasks, evaluated over 50 physical rollouts per task and method.

Method	Task 1 (%)	Task 2 (%)	Overall (%)
FedAvg	48.0	22.0	35.0
FedProx	16.0	4.0	10.0
ForgeVLA	70.0	42.0	56.0

Table 5: Ablation study on ForgeVLA's hyperparameters with $E = 10$ and $P_{\text{ep}} = 10$. Default settings are marked in gray. **bold** marks the best-performing results.

(a) The number of task per client.

Method	s	Success Rate (%)
FedAvg	1	0.8
FedAvg	2	4.2
FedAvg	3	17.6
FedAvg	5	75.6
FedAvg	10	93.2
ForgeVLA	1	2.0
ForgeVLA	2	28.7
ForgeVLA	3	41.6
ForgeVLA	5	85.6
ForgeVLA	10	95.4

(b) The LoRA rank r_{vla} of the VLM encoder.

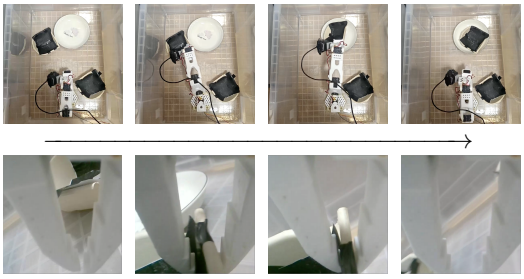
Method	r_{vla}	# Trainable (M)	Success Rate (%)
FedAvg	4	119.33	9.8
FedAvg	8	120.58	14.6
FedAvg	16	123.09	18.6
FedAvg	32	128.10	17.6
FedAvg	64	138.13	32.0
ForgeVLA	4	119.33	36.4
ForgeVLA	8	120.58	36.8
ForgeVLA	16	123.09	37.2
ForgeVLA	32	128.10	41.6
ForgeVLA	64	138.13	42.6

(c) The hyperparameter α_{CP}.

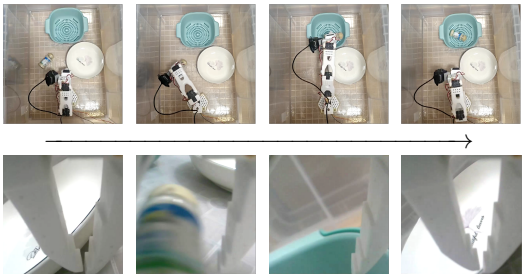
Method	α_{CP}	Success Rate (%)
ForgeVLA	0	35.2
ForgeVLA	0.1	36.0
ForgeVLA	0.2	**41.6**
ForgeVLA	0.5	37.0
ForgeVLA	1	36.0
ForgeVLA	2	33.8

(d) The hyperparameter α_{AG}.

Method	α_{AG}	Success Rate (%)
ForgeVLA	0	37.2
ForgeVLA	0.05	38.8
ForgeVLA	0.1	**41.6**
ForgeVLA	0.2	35.8



(a) Task 1: Pick up the black bowl next to the plate and place it on the plate.



(b) Task 2: Pick up the barbecue sauce and place it in the basket.

Figure 5: Real-world deployment of the trained ForgeVLA on the SO-ARM101 robotic platform.

ForgeVLA remains stable across reasonable hyperparameter ranges, with default settings achieving the best efficiency-performance trade-off.

Real-world Deployment: We further evaluate ForgeVLA on two SO-ARM101 manipulation tasks, with representative rollouts in Figure 5. We construct two clients with complementary task-skewed partitions (10%/90% and 90%/10% across Tasks 1 and 2), hold the training budget and evaluation protocol fixed across methods, and run 50 physical rollouts per task and method. ForgeVLA reaches 56.0% overall success, 21.0 pp above FedAvg and 46.0 pp above FedProx (Table 6), quantitative evidence of its effectiveness under task-level heterogeneity on physical hardware.

7 Conclusion

We propose ForgeVLA, a federated VLA training framework that turns distributed unannotated vision–action robotic logs into VLA training data without centralizing them. ForgeVLA recovers the missing language annotations on-device by classifying trajectories over a predefined instruction set, and mitigates the previously overlooked vision-language feature collapse under federated task heterogeneity through a contrastive planning loss and an adaptive aggregation strategy. Experiments show that ForgeVLA significantly outperforms federated baselines across benchmarks, and the Appendix discusses extensions such as cross-embodiment and cross-environment evaluation and open-vocabulary instruction recovery.

AI USE STATEMENT

AI was used solely for linguistic polishing and sentence refinement in this manuscript. All AI-generated content has been fully reviewed and revised by the authors, who retain full responsibility for all technical claims and conclusions.

ETHICS STATEMENT

Our work enables broader utilization of diverse vision-language data in federated VLA learning, and can facilitate the development of generalizable embodied and robotic VLA models. It brings positive impacts to downstream research including multimodal understanding, robotic manipulation, and privacy-preserving distributed learning. We do not identify any foreseeable negative societal or ethical impacts arising from this work.

REPRODUCIBILITY STATEMENT

Implementation details, hyperparameters, and compute are in Appendix G and Appendix L. The full algorithm is in Appendix B. All assumptions and complete proofs for the convergence result are in Appendix E. Source code will be released upon publication.

REFERENCES

Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *ACM SIGSAC Conference on Computer and Communications Security*, 2016.

Milad Abolhasani and Eugenia Kumacheva. The rise of self-driving labs in chemical and materials sciences. *Nature Synthesis*, 2(6):483–492, 2023. doi: 10.1038/s44160-022-00231-0.

Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N. Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *International Conference on Learning Representations*, 2021.

Amazon. Amazon launches a new AI foundation model to power its robotic fleet and deploys its 1 millionth robot, 2025. URL https://www.aboutamazon.com/news/operations/amazon-million-robots-ai-foundation-model.

Omar Besbes, Yonatan Gur, and Assaf Zeevi. Non-stationary stochastic optimization. *Operations Research*, 63(5):1227–1244, 2015. doi: 10.1287/opre.2015.1408. URL https://pubsonline.informs.org/doi/10.1287/opre.2015.1408.

Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models. In *International Conference on Learning Representations (ICLR)*, 2024.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. In *Robotics: Science and Systems (RSS)*, 2025.

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *ACM SIGSAC Conference on Computer and Communications Security*, 2017.

Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, Brendan McMahan, et al. Towards federated learning at scale: System design. *Proceedings of Machine Learning and Systems*, 1: 374–388, 2019.

Santiago Bou Betran, Alberta Longhini, Miguel Vasco, Yuchong Zhang, and Danica Kragic. FLAME: A federated learning benchmark for robotic manipulation. *arXiv preprint arXiv:2503.01729*, 2025.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901, 2020.

Xinyi Chen, Yilun Chen, Yanwei Fu, Ning Gao, Jiaya Jia, Weiyang Jin, Hao Li, Yao Mu, Jiangmiao Pang, Yu Qiao, et al. InternVLA-M1: A spatially guided vision-language-action framework for generalist robot policy. *arXiv preprint arXiv:2510.13778*, 2025.

Miao Cui, 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 *International Conference on Computer Vision (ICCV)*, 2025.

Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. RoboNet: Large-scale multi-robot learning. In *Conference on Robot Learning (CoRL)*, 2019.

Ria Doshi, Homer Rich Walke, Oier Mees, Sudeep Dasari, and Sergey Levine. Scaling cross-embodied learning: One policy for manipulation, navigation, locomotion and aviation. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 496–512. PMLR, 2024.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. PaLM-E: An embodied multimodal language model. In *International Conference on Machine Learning (ICML)*, 2023.

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. *Advances in neural information processing systems*, 36:9156–9172, 2023.

Tom Duckett, Simon Pearson, Simon Blackmore, and Bruce Grieve. Agricultural robotics: The future of robotic agriculture. Uk-ras white paper, EPSRC UK-RAS Network, 2018.

Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients – how easy is it to break privacy in federated learning? In *Advances in Neural Information Processing Systems*, 2020.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

International Federation of Robotics. World robotics 2025 – industrial robots. Technical report, IFR Statistical Department, Frankfurt, 2025. 4,664,000 industrial robots in operational use worldwide in 2024. Available at `https://ifr.org/ifr-press-releases/news/global-robot-demand-in-factories-doubles-over-10-years`.

Intuitive Surgical, Inc. 2025 annual report, 2025. URL `https://isrg.intuitive.com/static-files/d01bbc25-f8cf-433b-8ebb-b5afc1926236`.

Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. RLBench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 2020.

Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1–2), 2021.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5132–5143. PMLR, 2020. URL https://proceedings.mlr.press/v119/karimireddy20a.html.

Ben Kehoe, Sachin Patil, Pieter Abbeel, and Ken Goldberg. A survey of research on cloud robotics and automation. *IEEE Transactions on automation science and engineering*, 12(2):398–409, 2015.

Ahmed Khaled, Konstantin Mishchenko, and Peter Richtarik. Tighter theory for local SGD on identical and heterogeneous data. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 4519–4529. PMLR, 2020. URL https://proceedings.mlr.press/v108/bayoumi20a.html.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B. Tenenbaum. Learning to act from actionless videos through dense correspondences. *arXiv preprint arXiv:2310.08576*, 2023.

Ilya Kostrikov, Denis Yarats, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International Conference on Learning Representations (ICLR)*, 2021.

Michael Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 19884–19895, 2020.

Gihun Lee, Minchan Jeong, Yongjin Shin, Sangmin Bae, and Se-Young Yun. Preservation of the global knowledge by not-true distillation in federated learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Hongxia Li, Wei Huang, Jingya Wang, and Ye Shi. Global and local prompts cooperation via optimal transport for federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12151–12161, 2024a.

Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10713–10722, 2021.

Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, Xiaofan Wang, Bei Liu, Jianlong Fu, Jianmin Bao, Dong Chen, Yuanchun Shi, Jiaolong Yang, and Baining Guo. CogACT: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024b.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Conference on Machine Learning and Systems (MLSys)*, 2020.

Sen Lin, Daouda Sow, Kaiyi Ji, Yingbin Liang, and Ness Shroff. Non-convex bilevel optimization with time-varying objective functions. *Advances in Neural Information Processing Systems*, 36: 29692–29717, 2023.

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

Boyi Liu, Lujia Wang, Ming Liu, and Cheng-Zhong Xu. Federated imitation learning: A novel framework for cloud robotic systems with heterogeneous sensor data. *IEEE Robotics and Automation Letters*, 2020.

Julien Mairal. Optimization with first-order surrogate functions. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 783–791. PMLR, 2013.

Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. MimicGen: A data generation system for scalable robot learning using human demonstrations. In *Conference on Robot Learning (CoRL)*, 2023.

H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

Tongzhou Mu, Zhan Ling, Fanbo Xiang, Derek Yang, Xuanlin Li, Stone Tao, Zhiao Huang, Zhiwei Jia, and Hao Su. Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2021.

Niklas Muennighoff, Alexander M. Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. Scaling data-constrained language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

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

Open X-Embodiment Collaboration. Open X-embodiment: Robotic learning datasets and RT-X models. *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, and Xuelong Li. SpatialVLA: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Meisam Razaviyayn, Mingyi Hong, and Zhi-Quan Luo. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM Journal on Optimization*, 23(2):1126–1153, 2013. doi: 10.1137/120891009.

Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and H. Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations (ICLR)*, 2021. URL `https://openreview.net/forum?id=LkFG3lB13U5`.

Desik Rengarajan, Nitin Ragothaman, Dileep Kalathil, and Srinivas Shakkottai. FEDORA: Federated ensemble-directed offline reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Moritz Reuss, Ömer Erdinç Yağmurlu, Fabian Wenzel, and Rudolf Lioutikov. Multimodal diffusion transformer: Learning versatile behavior from multimodal goals. In *Robotics: Science and Systems (RSS)*, 2024.

Seonguk Seo, Jinkyu Kim, Geeho Kim, and Bohyung Han. Relaxed contrastive learning for federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Mingjia Shi, Yuhao Zhou, Kai Wang, Huaizheng Zhang, Shudong Huang, Qing Ye, and Jiancheng Lv. Prior: Personalized prior for reactivating the information overlooked in federated learning. *Advances in Neural Information Processing Systems*, 36:28378–28392, 2023a.

Yujun Shi, Jian Liang, Wenqing Zhang, Vincent Y. F. Tan, and Song Bai. Towards understanding and mitigating dimensional collapse in heterogeneous federated learning. In *International Conference on Learning Representations (ICLR)*, 2023b.

Sebastian U. Stich. Local SGD converges fast and communicates little. In *International Conference on Learning Representations (ICLR)*, 2019. URL `https://openreview.net/forum?id=S1g2JnRcFX`.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2446–2454, 2020.

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.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 23–30, 2017.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. Will we run out of data? limits of LLM scaling based on human-generated data. *arXiv preprint arXiv:2211.04325*, 2022.

Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, pp. 1723–1736. PMLR, 2023.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H. Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Lirui Wang, Xinlei Chen, Jialiang Zhao, and Kaiming He. Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers. In *Advances in Neural Information Processing Systems*, 2024.

Junjie Wen, Yichen Zhu, Jinming Li, Zhibin Tang, Chaomin Shen, and Feifei Feng. DexVLA: Vision-language model with plug-in diffusion expert for general robot control. *arXiv preprint arXiv:2502.05855*, 2025.

Russell B. Wynn, Veerle A.I. Huvenne, Timothy P. Le Bas, Bramley J. Murton, Douglas P. Connelly, Brian J. Bett, Henry A. Ruhl, Kirsty J. Morris, Jeffrey Peakall, Daniel R. Parsons, Esther J. Sumner, Stephen E. Darby, Robert M. Dorrell, and James E. Hunt. Autonomous underwater vehicles (AUVs): Their past, present and future contributions to the advancement of marine geoscience. *Marine Geology*, 352:451–468, 2014. doi: 10.1016/j.margeo.2014.03.012.

Hu Xu, Saining Xie, Xiaoqing Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. Demystifying clip data. In *International Conference on Learning Representations*, volume 2024, pp. 47812–47831, 2024.

Liping Yi, Han Yu, Chao Ren, Gang Wang, Xiaoguang Liu, and Xiaoxiao Li. Federated model heterogeneous matryoshka representation learning. *Advances in Neural Information Processing Systems*, 37:66431–66454, 2024.

Qiying Yu, Yang Liu, Yimu Wang, Ke Xu, and Jingjing Liu. Multimodal federated learning via contrastive representation ensemble. In *International Conference on Learning Representations (ICLR)*, 2023a.

Tianhe Yu, Ted Xiao, Austin Stone, Jonathan Tompson, Anthony Brohan, Su Wang, Jaspiar Singh, Clayton Tan, Jodilyn Peralta, Brian Ichter, et al. Scaling robot learning with semantically imagined experience. *arXiv preprint arXiv:2302.11550*, 2023b.

Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-IID data. *arXiv preprint arXiv:1806.00582*, 2018.

Ruijie Zheng, Dantong Niu, Yuqi Xie, Jing Wang, Mengda Xu, Yunfan Jiang, Fernando Castañeda, Fengyuan Hu, You Liang Tan, Letian Fu, et al. Egoscale: Scaling dexterous manipulation with diverse egocentric human data. *arXiv preprint arXiv:2602.16710*, 2026.

Kaiwen Zhou and Xin Eric Wang. FedVLN: Privacy-preserving federated vision-and-language navigation. In *European Conference on Computer Vision (ECCV)*, 2022.

Yuhao Zhou, Qing Ye, and Jiancheng Lv. Communication-efficient federated learning with compensated overlap-fedavg. *IEEE Transactions on Parallel and Distributed Systems*, 33(1):192–205, 2021.

Yuhao Zhou, Mingjia Shi, Yuanxi Li, Yanan Sun, Qing Ye, and Jiancheng Lv. Communication-efficient federated learning with single-step synthetic features compressor for faster convergence. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5008–5017, 2023.

Yuhao Zhou, Mingjia Shi, Yuxin Tian, Yuanxi Li, Qing Ye, and Jiancheng Lv. Federated cinn clustering for accurate clustered federated learning. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5590–5594. IEEE, 2024.

Yuhao Zhou, Jindi Lv, Yuxin Tian, Dan Si, Qing Ye, and Jiancheng Lv. Deploying models to non-participating clients in federated learning without fine-tuning: A hypernetwork-based approach. *arXiv preprint arXiv:2508.12673*, 2025a.

Yuhao Zhou, Yuxin Tian, Mingjia Shi, Yuanxi Li, Yanan Sun, Qing Ye, and Jiancheng Lv. E-3sfc: Communication-efficient federated learning with double-way features synthesizing. *IEEE Transactions on Neural Networks and Learning Systems*, 2025b.

Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. *Advances in Neural Information Processing Systems*, 2019.

Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *International Conference on Machine Learning (ICML)*, 2021.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pp. 2165–2183. PMLR, 2023.

TABLE OF CONTENTS

A Additional Related Work **16**
A.1 Vision-Language-Action Models and Data Scaling **16**
A.2 Federated Learning for Robot Policies and VLA **16**
A.3 Representation Alignment in Federated Training **17**
B Algorithm of ForgeVLA **17**
C Complexity Analysis **17**
D Notation **19**
E Convergence Analysis **19**
F Empirical Convergence **26**
G Additional Experimental Settings **26**
H Additional Evaluation Results **27**
I Effect of the Contrastive Planning Loss on Feature Separation **27**
J Privacy Implication **29**
K Limitations **29**
L Experiments Compute Resources **29**

A ADDITIONAL RELATED WORK

A.1 VISION-LANGUAGE-ACTION MODELS AND DATA SCALING

Vision-Language-Action models treat robot control as a single prediction problem over images, instructions, and actions. RT-2 (Zitkovich et al., 2023) showed that large vision-language backbones can be adapted to robotic trajectories, while OpenVLA (Kim et al., 2024) and OpenVLA-OFT (Kim et al., 2025) made this line easier to study at scale and improved practical efficiency. Recent systems push the frontier in different ways. π_0 (Black et al., 2025), CogACT (Li et al., 2024b), SpatialVLA (Qu et al., 2025), InternVLA-M1 (Chen et al., 2025), and DexVLA (Wen et al., 2025) improve dexterous control, spatial reasoning, and action modeling. Octo (Octo Model Team et al., 2024), CrossFormer (Doshi et al., 2024), and HPT (Wang et al., 2024) study transfer across embodiments and datasets. A common pattern is that these methods still depend on carefully annotated demonstrations.

That dependence is hard to scale in robotics. Prior work has tried to ease the bottleneck with data augmentation, simulation, and generated trajectories (Laskin et al., 2020; Kostrikov et al., 2021; Mandlekar et al., 2023; James et al., 2020; Mu et al., 2021; Tobin et al., 2017; Du et al., 2023; Black et al., 2024; Ko et al., 2023; Yu et al., 2023b). These approaches are useful, but they still rely on curated seed data or they struggle to produce reliable action supervision in the real world. Another practical line focuses on parameter-efficient adaptation. LoRA (Hu et al., 2022) is especially relevant because it makes large VLA backbones easier to tune and reduces the communication burden in distributed training (Kim et al., 2025; Chen et al., 2025). ForgeVLA is complementary to all of these efforts. Instead of synthesizing more trajectories, it turns existing vision–action logs into usable VLA training data.

A.2 FEDERATED LEARNING FOR ROBOT POLICIES AND VLA

Federated learning lets multiple clients train a shared model without uploading raw data (McMahan et al., 2017; Bonawitz et al., 2019; Kairouz et al., 2021). Its main difficulty is heterogeneity across clients. When local data distributions differ, client updates drift apart and the global model becomes harder to optimize (Zhao et al., 2018; Li et al., 2020; Karimireddy et al., 2020; Wang et al., 2020;

Acar et al., 2021; Reddi et al., 2021). This issue is especially sharp in robot learning because tasks, embodiments, and operating environments vary widely from one client to another.

Several studies bring federated training closer to robotics. FLAME (Bou Betran et al., 2025) provides a benchmark for federated manipulation and makes the performance gap under task heterogeneity easy to measure. FedVLA (Cui et al., 2025) adapts federated training to VLA models with instruction-oriented scene parsing, expert routing, and expert-aware aggregation. Related work on cloud robotics, federated imitation learning, vision-and-language navigation, and offline reinforcement learning also shows that privacy-preserving collaboration is feasible for distributed robot systems (Kehoe et al., 2015; Liu et al., 2020; Zhou & Wang, 2022; Rengarajan et al., 2024). Still, most prior work assumes either fully annotated multimodal data or simpler policy classes. ForgeVLA studies a more practical setting in which each client only stores vision–action logs and must recover the missing language signal locally.

A.3 Representation Alignment in Federated Training

Another relevant thread studies representation alignment as a way to handle non-i.i.d. federated training. MOON (Li et al., 2021) uses contrastive objectives to keep local representations close to the global model, and FedProto (Tan et al., 2022) exchanges prototypes so that clients can align features around shared anchors. Beyond these two foundational methods, subsequent work has expanded the toolkit considerably. FedNTD (Lee et al., 2022) draws an analogy to continual learning and preserves global dark knowledge on non-ground-truth classes via self-distillation during local training. FedGen (Zhu et al., 2021) trains a lightweight server-side generator to ensemble client knowledge without any proxy data and broadcasts it to regularize local updates. FedDecorr (Shi et al., 2023b) identifies that data heterogeneity causes dimensional collapse, where learned features reside in a lower-dimensional subspace, and applies a decorrelation regularizer to encourage full-rank representations. Seo et al. (2024) show that supervised contrastive learning in FL can itself trigger representation collapse and propose a relaxed contrastive loss that penalizes excessively similar within-class pairs. Complementary directions include optimal-transport-based prompt cooperation for aligning global and local representations (Li et al., 2024a), matryoshka-style nested representations that handle model heterogeneity across clients (Yi et al., 2024), and personalized priors that reactivate information overlooked during standard aggregation (Shi et al., 2023a). Collectively, these methods show that feature-level regularization can be more effective than only constraining parameter updates.

Multimodal federated settings introduce additional challenges. CreamFL (Yu et al., 2023a) tackles heterogeneous modalities and model architectures by combining inter-modal and intra-modal contrastive losses with a global-local representation ensemble, demonstrating that cross-modal alignment is critical when clients observe different data modalities. Federated VLA makes this problem even harder because the model must preserve agreement between visual content, language instructions, and action predictions in one shared latent space. In our setting, weak alignment leads to vision-language feature collapse, where task representations lose separation across clients. Generic contrastive FL methods do not fully address this multimodal failure mode. ForgeVLA therefore uses a contrastive planning loss with a shared task representation bank so that inter-task structure remains stable throughout training.

B Algorithm of ForgeVLA

The complete pseudocode of ForgeVLA is presented in Algorithm 1.

C Complexity Analysis

In each communication round, each participating client uses the pretrained embodied instruction classifier to convert raw vision–action logs into a VLA training corpus, and then performs local optimization by jointly minimizing the VLA task loss and the contrastive planning loss. The server aggregates the uploaded client updates via adaptive aggregation and updates the global task representation bank. Throughout, it never accesses raw client data. Therefore, ForgeVLA incurs no additional asymptotic time complexity beyond vanilla FedAvg, and its runtime remains $\mathcal{O}(NEP)$,

Algorithm 1 ForgeVLA

Input: initial global model parameters θ^0 and ϕ, local dataset $D_i = \{(v_i^k, a_i^k)\}_{k=1}^{K_i}$, local learning rate η_i
Parameter: number of communication rounds E, number of local optimization steps P, number of clients N, participation rule $\{\pi_i^e\}$
Output: global model parameters θ^E
Server:

fine-tune ϕ on public VLA trajectories
initialize the normalized global task bank $U^0 = \{\hat{u}_1^0, \hat{u}_2^0, \ldots, \hat{u}_M^0\}$
for each communication round e from 0 to $E-1$ **do**
  sample a participating set $S^e \subseteq [N]$ according to $\{\pi_i^e\}_{i=1}^N$
  distribute θ^e, ϕ, and U^e to the clients in S^e
  each client in S^e executes the routine below in parallel
  receive $\theta_i^{e,P}$ and $\{u_{i,1}^e, u_{i,2}^e, \ldots, u_{i,M}^e\}$ from each client $i \in S^e$
  update U^{e+1} using weighted aggregation over the uploaded prototypes; if a task receives no valid aggregate, keep its previous bank entry, and renormalize each updated entry
  update θ^{e+1} using Equation 6 on the uploaded client set S^e
end for
return θ^E

Client Routine (executed on participating client $i \in S^e$ at round e):

initialize $\theta_i^{e,0} = \theta^e$
recover an instruction for every local trajectory, $\hat{l}_i^k = c_\phi(x_i^{q(k)})$, to complete the local VLA corpus
for each local iteration j from 1 to P **do**
  sample a mini-batch $\mathcal{B}_i^{e,j}$ from D_i
  form an unbiased stochastic gradient $h_i^{e,j}$ of $\mathcal{L}_i + \mathcal{L}_{\mathrm{CP},i}^e$ at $(\theta_i^{e,j-1}, \mathcal{B}_i^{e,j})$
  $\theta_i^{e,j} = \theta_i^{e,j-1} - \eta_i h_i^{e,j}$
end for
update $\{u_{i,1}^e, u_{i,2}^e, \ldots, u_{i,M}^e\}$ using $\theta_i^{e,P}$
return $\theta_i^{e,P}$ and $\{u_{i,1}^e, u_{i,2}^e, \ldots, u_{i,M}^e\}$

where E is the number of communication rounds and P is the number of local optimization steps per round. In space, ForgeVLA adds two terms on top of the $\mathcal{O}(N)$ storage that FedAvg already needs for the uploaded client updates. The instruction classifier introduces only constant extra storage, and the global task representation bank requires $\mathcal{O}(Md)$ storage for M task representations of dimension d. Both terms are independent of the local dataset sizes and of the number of communication rounds, so the total server-side storage is $\mathcal{O}(N + Md)$ and stays lightweight in practice because M is the size of the instruction set rather than a data-dependent quantity.

Table 7: The glossary of notations. We reserve $\mathcal{I}$ for the instruction space, and every symbol of the form $\mathcal{L}_\bullet$ denotes a loss.

Notation	Implication
N	Total number of clients
i	Index of a client
k	Sample index within the local dataset D_i
e	Communication-round index
j	Local optimization-iteration index within round e
S^e	Participating client set at communication round e
ξ_i^e	Participation indicator of client i at round e
π_i^e	Participation probability of client i at round e
$\mathcal{B}_i^{e,j}$	Mini-batch sampled on client i at local iteration j of round e
w_i	Normalized analysis weight of the i-th client, *e.g.*, $w_i = \lvert D_i\rvert / \sum_{j=1}^N \lvert D_j\rvert$, satisfying $\sum_{i=1}^N w_i = 1$
$\tilde{w}_i^e$	Unbiased sampled server weight for client i at round e
E	Number of communication rounds
P	Number of local optimization steps per round
P_{ep}	Number of local epochs per round used in the experiments, inducing $P = P_{\text{ep}}\lceil K_i/B\rceil$ steps at batch size B
η_i	Local learning rate for client i
D_i	Local dataset of client i
K_i	Number of samples in the local dataset D_i
B	Local mini-batch size
(v_i^k, a_i^k)	The k-th vision–action pair in D_i
l_m	The m-th instruction in the instruction space $\mathcal{I}$
l_i^k	Ground-truth instruction of the k-th pair of client i, which is unavailable in our setting
$\hat{l}_i^k$	Predicted instruction from the embodied instruction classifier
t_i^k	Instruction index satisfying $\hat{l}_i^k = l_{t_i^k}$
s	Number of tasks assigned per client
p	Fraction of annotated data reserved for classifier fine-tuning
$\mathcal{V}$	Vision observation space
$\mathcal{A}$	Action space
$\mathcal{I} = \{l_m\}_{m=1}^M$	Instruction space
$\mathcal{Z}$	Joint latent embedding space
d	Dimension of the joint latent embedding space $\mathcal{Z}$
M	Total number of distinct language instructions

D Notation

The main notations in this paper are shown in Table 7 and Table 8.

E Convergence Analysis

We analyze ForgeVLA as local SGD on the round-frozen objectives $\{F_i^e\}$ induced by the current bank U^e (Stich, 2019; Khaled et al., 2020; Karimireddy et al., 2020; Reddi et al., 2021). The purpose of this analysis is to provide a *stability-style guarantee*: under bounded algorithm-induced perturbations, adaptive aggregation and the evolving task bank preserve the FedAvg/local-SGD stationarity order. The theorem does not claim that these components explain the empirical advantage

Table 8: The glossary of notations, continued from Table 7.

Notation	Implication
$\theta = (\theta_{\text{enc}}, \theta_{\text{dec}}) \in \Theta_{\text{vla}}$	Global VLA model parameters (encoder and decoder) and their hypothesis space
θ^e	Global VLA model at the start of round e
$\theta_i^{e,j}$	Local VLA model of client i after j local iterations in round e
$f_\theta(v, l)$	Full VLA model mapping from vision-language input to action
$f_{\theta_{\text{enc}}} : \mathcal{V} \times \mathcal{I} \to \mathcal{Z}$	Vision-language encoder
$f_{\theta_{\text{dec}}} : \mathcal{Z} \to \mathcal{A}$	Action decoder
z_i^k	Joint latent representation of sample (v_i^k, a_i^k)
$\hat{z}_i^k$	ℓ_2-normalized latent representation of sample (v_i^k, a_i^k)
r_{vla}	LoRA rank applied to the VLM encoder of the VLA model
r_{cls}	LoRA rank applied to the embodied instruction classifier
$c_\phi : \mathcal{X} \to \mathcal{I}$	Embodied instruction classifier
$\mathcal{X} = \bigcup_{T \geq 1} (\mathcal{V} \times \mathcal{A})^T$	Space of vision–action trajectories of arbitrary length
Q_i	Number of trajectories in the local dataset D_i
x_i^q	The q-th vision–action trajectory of client i
$q(k)$	Index of the trajectory containing the k-th pair of client i
$\phi \in \Phi$	Parameters of the instruction classifier and their hypothesis space
$\ell(\cdot, \cdot)$	Task-specific loss function
$\mathcal{L}_i(\theta; D_i, \phi)$	Local empirical VLA loss on client i
$\mathcal{L}_{\text{CP},i}^e$	Contrastive planning loss on client i at round e
α_{CP}	Scaling hyperparameter for the contrastive planning loss
τ	Fixed temperature hyperparameter in the contrastive planning loss
$U^e = \{\hat{u}_1^e, \hat{u}_2^e, \ldots, \hat{u}_M^e\}$	Normalized global task representation bank broadcast at round e
$\hat{u}_m^e$	Normalized bank entry of instruction m at round e
$\tilde{u}_m^e$	Weighted aggregate of the uploaded representations of instruction m at round e, before normalization
$u_{i,m}^e$	Local task representation of instruction m on client i at round e
g_i^e	Local update of client i at round e, *i.e.*, $\theta_i^{e,P} - \theta^e$
α_{AG}	Hyperparameter balancing projection alignment and regularization
ε_{AG}	Numerical stabilizer in adaptive aggregation
ε_{n}	Numerical stabilizer of the smoothed normalization used only in the convergence analysis
Pass@K	Probability that a task succeeds at least once within K evaluation rollouts, where K is unrelated to the dataset size K_i
λ	Regularization intensity of a compared federated baseline, *i.e.*, the proximal weight in FedProx and the contrastive or prototype weight in MOON and FedProto

of ForgeVLA over baselines. The analysis is stated as one theorem in which stochastic noise, client heterogeneity, partial participation, adaptive aggregation, and bank drift appear in a single bound.

Recall that $K_i = |D_i|$, and define the normalized client weights $w_i \triangleq K_i / \sum_{j=1}^{N} K_j$, so $w_i \geq 0$ and $\sum_{i=1}^{N} w_i = 1$. We assume a common local step size $\eta > 0$ and $P \in \mathbb{N}_+$ local mini-batch steps per round. In the experiments, the local budget is instead specified in epochs, denoted by P_{ep}. One local epoch on client i corresponds to $\lceil K_i/B \rceil$ mini-batch steps at batch size $B = 32$, so P_{ep} local epochs induce $P = P_{\text{ep}} \lceil K_i/B \rceil$ steps. The two extra ForgeVLA hyperparameters are coupled to the local-SGD scale as

$$\alpha_{\text{CP}} = \lambda_{\text{CP}} \eta, \qquad \alpha_{\text{AG}} = \frac{\lambda_{\text{AG}}}{\eta P}, \qquad \lambda_{\text{CP}} \geq 0, \quad \lambda_{\text{AG}} > 0. \tag{7}$$

The first scaling keeps the bank term at the same order as the usual stochastic floor, and the second serves as a convenient asymptotic parametrization of the server anchor strength as the local-update scale shrinks. The theorem itself relies on Assumption 6 to quantify the objective-level perturbation induced by adaptive aggregation, rather than on this heuristic alone.

As in the rest of the paper, the bank entries are row-wise normalized; for analysis we replace the raw normalization $\hat{z} = z/\|z\|_2$ by the smooth map

$$\hat{z} \triangleq \text{norm}_{\varepsilon_{\text{n}}}(z) \triangleq \frac{z}{\sqrt{\|z\|_2^2 + \varepsilon_{\text{n}}^2}}, \qquad \varepsilon_{\text{n}} > 0, \tag{8}$$

so that $\|\hat{z}\|_2 \leq 1$ deterministically. This ε_{n}-smoothed map is used only as an analytical surrogate for the proof: the implemented algorithm uses standard row-wise normalization, while the smoothed version removes the singularity at $z = 0$ and approximates the raw normalization arbitrarily well once $\|z\|_2 \gg \varepsilon_{\text{n}}$. At communication round e, the server broadcasts a bank $U^e = [\hat{u}_1^e; \ldots; \hat{u}_M^e]$ with $\|\hat{u}_m^e\|_2 \leq 1$ for every m, and this bank is held fixed throughout the subsequent P local updates. For one sample with target task t, and with the round-e bank U^e, define

$$\ell_{\text{CP}}(\hat{z}; U^e, t) \triangleq -\alpha_{\text{CP}} \left(\frac{\hat{z}^\top \hat{u}_t^e}{\tau} - \log \sum_{m=1}^{M} \exp\left(\frac{\hat{z}^\top \hat{u}_m^e}{\tau} \right) \right), \qquad \tau > 0. \tag{9}$$

Since $\log \sum_{m=1}^{M} \exp(a_m) \geq a_t$, we have $\ell_{\text{CP}}(\hat{z}; U^e, t) \geq 0$ for every $(\hat{z}, U^e, t)$. Throughout this subsection, $\mathcal{L}_i(\theta)$ denotes the empirical-average VLA loss on D_i, and $\mathcal{L}_{\text{CP},i}^e(\theta)$ denotes the empirical average of $\ell_{\text{CP}}(\hat{z}; U^e, t)$ over the samples in D_i. For brevity, write $\mathcal{L}_i(\theta) \equiv \mathcal{L}_i(\theta; D_i, \phi)$ and $\mathcal{L}_{\text{CP},i}^e(\theta) \equiv \mathcal{L}_{\text{CP},i}^e(\theta; D_i, \phi, U^e)$, and define the round-frozen local and global objectives as

$$F_i^e(\theta) \triangleq \mathcal{L}_i(\theta) + \mathcal{L}_{\text{CP},i}^e(\theta), \qquad F^e(\theta) \triangleq \sum_{i=1}^{N} w_i F_i^e(\theta). \tag{10}$$

Let $\mathcal{F}_e$ denote the sigma-field generated by all randomness up to the start of round e, and let $\mathcal{F}_{e,j}$ denote the sigma-field generated by all randomness up to and including the j-th local update on every client in the coupled full-participation round-e recursion, with $\mathcal{F}_{e,0} = \mathcal{F}_e$. Then θ^e and U^e are $\mathcal{F}_e$-measurable, and conditioned on $\mathcal{F}_e$ the inner loop is standard local SGD on $\{F_i^e\}_{i=1}^N$.

Conditioned on the round-start sigma-field $\mathcal{F}_e$, each client starts from $\theta_i^{e,0} = \theta^e$ and performs P local SGD steps on F_i^e:

$$\theta_i^{e,j} = \theta_i^{e,j-1} - \eta h_i^{e,j}, \qquad j = 1, \ldots, P, \tag{11}$$

where $h_i^{e,j}$ is a stochastic gradient estimator of F_i^e at $\theta_i^{e,j-1}$. After the local stage, define

$$\bar{\theta}^{e+1} \triangleq \sum_{i=1}^{N} w_i \theta_i^{e,P}, \qquad g_i^e \triangleq \theta_i^{e,P} - \theta^e, \qquad \bar{g}^e \triangleq \bar{\theta}^{e+1} - \theta^e. \tag{12}$$

To model partial participation, let $\xi_i^e \in \{0, 1\}$ denote the round-e participation indicator of client i, let $\pi_i^e \triangleq \mathbb{P}(\xi_i^e = 1 \mid \mathcal{F}_e) \in (0, 1]$, and define the unbiased sampled server weights and sampled FedAvg anchor by

$$\tilde{w}_i^e \triangleq \frac{w_i \xi_i^e}{\pi_i^e}, \qquad \tilde{\theta}^{e+1} \triangleq \theta^e + \sum_{i=1}^{N} \tilde{w}_i^e g_i^e. \tag{13}$$

Equivalently, all sums below may be restricted to the uploaded client set because $\tilde{w}_i^e = 0$ for non-participating clients. The server then applies the adaptive-aggregation update

$$\theta^{e+1} \triangleq \arg\min_{\theta} \sum_{i=1}^{N} \tilde{w}_i^e \left(\frac{(\theta - \theta^e)^\top g_i^e}{\|g_i^e\|_2^2 + \varepsilon_{\text{AG}}} - 1 \right)^2 + \alpha_{\text{AG}} \left\| \theta - \tilde{\theta}^{e+1} \right\|_2^2, \tag{14}$$

where $\varepsilon_{\text{AG}} > 0$. Because the objective in Eq. equation 14 is strongly convex, θ^{e+1} is uniquely defined.

To analyze the bank update, let $U_i(\theta) = [u_{i,1}(\theta); \ldots; u_{i,M}(\theta)]$ denote the task-prototype matrix uploaded by client i when its local model is θ. Let $\mathcal{M}_{\text{act}} \subseteq [M]$ denote this active row set, and write $M_{\text{act}} \triangleq |\mathcal{M}_{\text{act}}|$. For each active task $m \in \mathcal{M}_{\text{act}}$, the server computes

$$\tilde{u}_m^{e+1} \triangleq \sum_{i=1}^{N} \rho_{i,m}^e \, u_{i,m}(\theta_i^{e,P}), \qquad \hat{u}_m^{e+1} \triangleq \text{norm}_{\varepsilon_{\text{n}}}(\tilde{u}_m^{e+1}), \tag{15}$$

where $\rho_{i,m}^e \geq 0$, $\sum_{i=1}^{N} \rho_{i,m}^e = 1$, and $\rho_{i,m}^e = 0$ whenever client i does not upload task m in round e. For $m \notin \mathcal{M}_{\text{act}}$, the server keeps $\hat{u}_m^{e+1} = \hat{u}_m^e$.

Assumption 1 (Round-Wise Smoothness)**.** *For every round e and client i, the objective F_i^e is L-smooth, i.e., for all θ, θ',*

$$\|\nabla F_i^e(\theta) - \nabla F_i^e(\theta')\|_2 \leq L\|\theta - \theta'\|_2. \tag{16}$$

The smoothed normalization keeps the contrastive term well defined, and the assumption above is the standard round-wise smoothness condition used in nonconvex local-SGD and federated optimization analyses (Stich, 2019; Khaled et al., 2020; Karimireddy et al., 2020; Reddi et al., 2021).

Assumption 2 (Uniform Lower Bound on the Round-Wise Objectives)**.** *There exists a deterministic constant $F_{\text{lb}} > -\infty$ such that*

$$F^e(\theta) \geq F_{\text{lb}} \qquad \textit{for all rounds } e \textit{ and all } \theta. \tag{17}$$

Uniform lower boundedness is the usual stationarity-analysis condition for smooth nonconvex optimization and local SGD (Stich, 2019; Reddi et al., 2021).

Because only the contrastive term depends on the bank, the inter-round objective drift is controlled directly by the bank movement. For the sample loss $\ell_{\text{CP}}(\hat{z}; U, t)$,

$$\nabla_{\hat{u}_m} \ell_{\text{CP}} = \frac{\alpha_{\text{CP}}}{\tau} \left(\omega_m - \mathbf{1}\{m = t\} \right) \hat{z}, \tag{18}$$

where ω_m is the corresponding softmax probability. Hence

$$\|\nabla_U \ell_{\text{CP}}\|_F^2 = \frac{\alpha_{\text{CP}}^2}{\tau^2} \|\hat{z}\|_2^2 \sum_{m=1}^{M} \left(\omega_m - \mathbf{1}\{m = t\} \right)^2 \leq \frac{2\alpha_{\text{CP}}^2}{\tau^2}, \tag{19}$$

because $\|\hat{z}\|_2 \leq 1$ and $\|\omega - \mathbf{1}_t\|_2 \leq \sqrt{2}$, where $\omega = (\omega_1, \ldots, \omega_M)$ and $\mathbf{1}_t$ is the one-hot vector at index t. Therefore ℓ_{CP} is $(\sqrt{2}\alpha_{\text{CP}}/\tau)$-Lipschitz in U, and taking empirical averages and the client-weighted sum preserves this constant:

$$|F^{e+1}(\theta) - F^e(\theta)| \leq \frac{\sqrt{2}\alpha_{\text{CP}}}{\tau} \|U^{e+1} - U^e\|_F \qquad \text{for all rounds } e \text{ and all } \theta. \tag{20}$$

Lemma 1 (Bank Variation Implies Objective Drift)**.** *Suppose there exists $C_U \geq 0$ such that, for every round e,*

$$\mathbb{E}\left[\|U^{e+1} - U^e\|_F \middle| \mathcal{F}_e \right] \leq C_U \eta P. \tag{21}$$

Then, under $\alpha_{\text{CP}} = \lambda_{\text{CP}} \eta$,

$$\mathbb{E}\left[F^{e+1}(\theta^{e+1}) - F^e(\theta^{e+1}) \middle| \mathcal{F}_e \right] \leq \frac{\sqrt{2}\lambda_{\text{CP}} C_U}{\tau} \eta^2 P. \tag{22}$$

Proof. Eq. equation 20 holds pointwise for any θ, including the random iterate θ^{e+1}. Taking conditional expectation and applying Eq. equation 21 gives Eq. equation 22. □

Eq. equation 21 is a bank path-length condition, not an optimal-bank convergence claim. It is mild for prototype banks because U^e is an averaged representation statistic rather than an adversarial objective sequence, matching the prototype-sharing view used in federated representation learning (Tan et al., 2022). If the joint model-bank dynamics approach a stable representation fixed point, then $\|U^{e+1} - U^e\|_F \to 0$; the rate theorem assumes the quantitative per-round control needed for an $\mathcal{O}(E^{-1/2})$ stationarity bound. Bounded variation/path-length controls are standard in non-stationary and time-varying optimization (Besbes et al., 2015; Lin et al., 2023); related surrogate analyses also control changes between successive optimization surrogates (Mairal, 2013; Razaviyayn et al., 2013).

Assumption 3 (Conditionally Unbiased Stochastic Gradients with Bounded Noise)**.** *For every round e, client i, and local step j,*

$$\mathbb{E}\Big[h_i^{e,j} \mid \mathcal{F}_{e,j-1}\Big] = \nabla F_i^e(\theta_i^{e,j-1}), \qquad \mathbb{E}\left[\left\|h_i^{e,j} - \nabla F_i^e(\theta_i^{e,j-1})\right\|_2^2 \middle| \mathcal{F}_{e,j-1}\right] \le \sigma^2. \tag{23}$$

This conditional unbiasedness and bounded-noise model is the standard stochastic-gradient oracle used in local-SGD/FedAvg convergence analyses (Stich, 2019; Khaled et al., 2020; Reddi et al., 2021).

Assumption 4 (Round-Wise Gradient Dissimilarity)**.** *There exists $\zeta \ge 0$ such that for every round e and every θ,*

$$\Xi_e(\theta) \triangleq \sum_{i=1}^{N} w_i \left\|\nabla F_i^e(\theta) - \nabla F^e(\theta)\right\|_2^2 \le \zeta^2. \tag{24}$$

This is the standard bounded gradient-dissimilarity condition used to quantify non-i.i.d. client drift in federated optimization (Li et al., 2020; Khaled et al., 2020; Karimireddy et al., 2020; Reddi et al., 2021). Here it is imposed on the round-frozen ForgeVLA objectives F_i^e, so ζ^2 is algorithm-dependent and captures any reduction in cross-client disagreement induced by the shared bank.

Assumption 5 (Unbiased Partial Participation)**.** *Conditioned on $\mathcal{F}_e$, the participation indicators $\{\xi_i^e\}_{i=1}^N$ introduced above are mutually independent and are also independent of the local mini-batch noise used to generate the coupled full-participation local updates $\{g_i^e\}_{i=1}^N$. Consequently, the sampled server weights in Eq. equation 13 satisfy*

$$\mathbb{E}[\tilde{w}_i^e \mid \mathcal{F}_e] = w_i. \tag{25}$$

Define the participation inflation factor

$$\Gamma_\pi \triangleq \max_{0 \le e \le E-1} \max_{1 \le i \le N} w_i \frac{1 - \pi_i^e}{\pi_i^e}. \tag{26}$$

Unbiased client sampling with bounded participation inflation is the partial-participation analogue of the client-sampling assumptions used in federated optimization analyses (Karimireddy et al., 2020; Reddi et al., 2021).

Lemma 2 (Anchored Adaptive-Aggregation Perturbation)**.** *Let $A^e(\theta)$ denote the first quadratic term in Eq. equation 14. Suppose that for some $G_{\mathrm{AG}} \ge 0$,*

$$\mathbb{E}\Big[\|\nabla A^e(\tilde{\theta}^{e+1})\|_2^2 \Big| \mathcal{F}_e\Big] \le G_{\mathrm{AG}}^2 \eta^2 P^2. \tag{27}$$

Then the solution of Eq. equation 14, with $\alpha_{\mathrm{AG}} = \lambda_{\mathrm{AG}}/(\eta P)$, satisfies

$$\mathbb{E}\Big[\|\theta^{e+1} - \tilde{\theta}^{e+1}\|_2^2 \Big| \mathcal{F}_e\Big] \le \frac{G_{\mathrm{AG}}^2}{4\lambda_{\mathrm{AG}}^2} \eta^4 P^4. \tag{28}$$

Consequently, by L-smoothness, Young's inequality, and the local-SGD moment bound in Eq. equation 39, Eq. equation 29 holds for constants ρ_{AG} and C_{AG} independent of E.

Proof. The function A^e is convex quadratic in θ. With $\delta^e = \theta^{e+1} - \tilde{\theta}^{e+1}$, first-order optimality gives $\nabla A^e(\theta^{e+1}) + 2\alpha_{\mathrm{AG}}\delta^e = 0$. Monotonicity of ∇A^e implies $2\alpha_{\mathrm{AG}}\|\delta^e\|_2 \le \|\nabla A^e(\tilde{\theta}^{e+1})\|_2$, which gives Eq. equation 28 after conditioning. Applying smoothness to $F^e(\theta^{e+1}) - F^e(\tilde{\theta}^{e+1})$, replacing $\nabla F^e(\tilde{\theta}^{e+1})$ by $\nabla F^e(\theta^e)$ plus an $L\|\tilde{\theta}^{e+1} - \theta^e\|_2$ term, and using Eq. equation 39 yields Eq. equation 29. □

Assumption 6 (ForgeVLA Objective-Level Perturbation Control)**.** *Let $\tilde{\theta}^{e+1}$ denote the sampled FedAvg reference point defined in Eq. equation 13. There exist deterministic constants $\rho_{\mathrm{AG}}, C_{\mathrm{AG}}, C_B \geq 0$, with ρ_{AG} a sufficiently small universal constant, such that, for every round e,*

$$\mathbb{E}\Big[F^e(\theta^{e+1}) - F^e(\tilde{\theta}^{e+1})\Big|\mathcal{F}_e\Big] \leq \rho_{\mathrm{AG}}\eta P\|\nabla F^e(\theta^e)\|_2^2 + C_{\mathrm{AG}}\eta^3 P^3, \tag{29}$$

$$\mathbb{E}\big[F^{e+1}(\theta^{e+1}) - F^e(\theta^{e+1})\big|\mathcal{F}_e\big] \leq C_B\eta^2 P. \tag{30}$$

This assumption isolates the two ForgeVLA-specific deviations from vanilla FedAvg at the objective level: adaptive aggregation contributes the perturbation term $C_{\mathrm{AG}}\eta^3 P^3$, and bank evolution contributes the drift term $C_B\eta^2 P$. Lemma 2 gives a sufficient anchored-aggregation condition for Eq. equation 29. Lemma 1 shows that Eq. equation 30 follows from the bank-variation bound in Eq. equation 21, with $C_B = \sqrt{2}\lambda_{\mathrm{CP}}C_U/\tau$. The theorem only requires the objective-level bounds.

Interpretation. The role of Assumption 6 is to capture, in a single unified statement, the two algorithm-specific perturbations that ForgeVLA introduces beyond standard local SGD. The first term (Eq. equation 29) bounds how much the adaptive-aggregation step can worsen the current-round objective relative to the sampled FedAvg baseline $\tilde{\theta}^{e+1}$. Because the adaptive-aggregation objective (Eq. equation 14) is anchored to $\tilde{\theta}^{e+1}$, Lemma 2 reduces this requirement to a moment bound on the projection-gradient at the anchor. The second term (Eq. equation 30) bounds the inter-round objective change due to bank evolution; Lemma 1 reduces it to bounded bank path length. Thus, ForgeVLA preserves the FedAvg/local-SGD convergence order under standard stochastic optimization assumptions plus bounded adaptive-aggregation and bank-variation perturbations.

For the optional comparison to a fixed bank, define

$$F_U(\theta) \triangleq \sum_{i=1}^{N} w_i\left(\mathcal{L}_i(\theta) + \mathcal{L}_{\mathrm{CP},i}(\theta; U)\right), \tag{31}$$

and, when needed, assume the bank-to-gradient continuity condition

$$\|\nabla F_U(\theta) - \nabla F_{U'}(\theta)\|_2 \leq L_U\|U - U'\|_F \qquad \text{for all } \theta \text{ and all banks } U, U', \tag{32}$$

for some deterministic constant $L_U \geq 0$.

Theorem 1 (FedAvg-Order Stationarity of ForgeVLA)**.** *Assume Assumptions 1, 2, 3, 4, 5, and 6, together with $\mathbb{E}[F^0(\theta^0)] < \infty$, the standard local-SGD small-step condition $L\eta P \leq 1$, and the participation-stability condition $L\eta P\Gamma_\pi \leq c_0$ for a sufficiently small universal constant $c_0 > 0$. Then there exist universal constants $c_\sigma, c_{\mathrm{het}}, c_\pi > 0$ such that, with*

$$\begin{aligned}\mathcal{R}_E \triangleq \frac{8\left(\mathbb{E}[F^0(\theta^0)] - F_{\mathrm{lb}}\right)}{\eta PE} \;+\; c_\sigma L\eta\sigma^2 \;+\; c_{\mathrm{het}}L^2\eta^2 P(P-1)\zeta^2 \\ +\; c_\pi L\eta P\Gamma_\pi(\sigma^2 + \zeta^2) \;+\; 8C_{\mathrm{AG}}\eta^2 P^2 \;+\; 8C_B\eta,\end{aligned} \tag{33}$$

the following main bound holds:

$$\frac{1}{E}\sum_{e=0}^{E-1}\mathbb{E}\|\nabla F^e(\theta^e)\|_2^2 \leq \mathcal{R}_E, \tag{34}$$

If, in addition, Eq. equation 32 holds, then for any reference bank $U^\star$,

$$\frac{1}{E}\sum_{e=0}^{E-1}\mathbb{E}\|\nabla F_{U^\star}(\theta^e)\|_2^2 \leq 2\mathcal{R}_E + \frac{2L_U^2}{E}\sum_{e=0}^{E-1}\mathbb{E}\|U^e - U^\star\|_F^2. \tag{35}$$

Consequently, if the bank sequence stabilizes in the Cesàro sense around some $U^\star$, then this additional continuity condition transfers the round-wise stationarity guarantee directly to the fixed objective $F_{U^\star}$. Moreover, under the stated bounded-perturbation conditions, fixed P, bounded Γ_π, and E-independent C_{AG}, C_B, choosing $\eta = \Theta(E^{-1/2})$ gives $\mathcal{R}_E = \mathcal{O}(E^{-1/2})$, matching the standard nonconvex local-SGD/FedAvg order. If Eq. equation 32 also holds with E-independent L_U and $\frac{1}{E}\sum_{e=0}^{E-1}\mathbb{E}\|U^e - U^\star\|_F^2 = \mathcal{O}(E^{-1/2})$, then the same order transfers to $F_{U^\star}$. When adaptive aggregation and bank evolution are absent, i.e., $C_{\mathrm{AG}} = C_B = 0$ and $F_i^e(\theta) \equiv \mathcal{L}_i(\theta)$, Eq. equation 34 reduces to the usual heterogeneity- and participation-aware FedAvg/local-SGD guarantee up to universal constants.

Proof. Conditioned on $\mathcal{F}_e$, couple the sampled round with the hypothetical full-participation local-SGD recursion driven by the same mini-batch draws. The standard nonconvex local-SGD descent estimate under bounded conditional noise and bounded gradient dissimilarity yields (Li et al., 2020; Khaled et al., 2020; Karimireddy et al., 2020; Reddi et al., 2021)

$$\mathbb{E}\big[F^e(\bar{\theta}^{e+1}) \mid \mathcal{F}_e\big] \le F^e(\theta^e) - \frac{\eta P}{4}\|\nabla F^e(\theta^e)\|_2^2 + c'_\sigma L\eta^2 P\sigma^2 + c'_{\mathrm{het}} L^2\eta^3 P^2(P-1)\Xi_e(\theta^e), \quad (36)$$

for universal constants $c'_\sigma, c'_{\mathrm{het}} > 0$.

Write the sampled displacement as $\tilde{g}^e = \sum_{i=1}^N \tilde{w}_i^e g_i^e = \bar{g}^e + \varepsilon_\pi^e$, where $\bar{g}^e = \sum_{i=1}^N w_i g_i^e$. Define $\varepsilon_\pi^e = \sum_{i=1}^N w_i \left(\frac{\xi_i^e}{\pi_i^e} - 1\right) g_i^e$. By Assumption 5, conditioned on $\mathcal{F}_e$ and $\{g_i^e\}_{i=1}^N$, the coefficients $w_i(\xi_i^e/\pi_i^e - 1)$ are mean-zero and mutually independent, hence the cross terms vanish and $\mathbb{E}[\varepsilon_\pi^e \mid \mathcal{F}_e] = 0$ and

$$\mathbb{E}\big[\|\varepsilon_\pi^e\|_2^2 \big| \mathcal{F}_e, \{g_i^e\}_{i=1}^N\big] = \sum_{i=1}^N w_i^2 \frac{1-\pi_i^e}{\pi_i^e}\|g_i^e\|_2^2. \quad (37)$$

Using $w_i^2 \frac{1-\pi_i^e}{\pi_i^e} \le \Gamma_\pi w_i$ for every i, e and then averaging over the local mini-batch noise yields

$$\mathbb{E}\big[\|\varepsilon_\pi^e\|_2^2 \mid \mathcal{F}_e\big] \le \Gamma_\pi \sum_{i=1}^N w_i \mathbb{E}\big[\|g_i^e\|_2^2 \mid \mathcal{F}_e\big]. \quad (38)$$

The same recursion also gives

$$\sum_{i=1}^N w_i \mathbb{E}\big[\|g_i^e\|_2^2 \mid \mathcal{F}_e\big] \le c_g \eta^2 P^2 \left(\|\nabla F^e(\theta^e)\|_2^2 + \sigma^2 + \zeta^2\right) \quad (39)$$

for a universal constant $c_g > 0$. Applying L-smoothness to $F^e(\tilde{\theta}^{e+1})$ and using Eqs. equation 36–equation 39 yields

$$\begin{aligned}\mathbb{E}\Big[F^e(\tilde{\theta}^{e+1}) \mid \mathcal{F}_e\Big] \le F^e(\theta^e) &- \frac{\eta P}{4}\|\nabla F^e(\theta^e)\|_2^2 + c'_\sigma L\eta^2 P\sigma^2 \\ &+ c'_{\mathrm{het}} L^2\eta^3 P^2(P-1)\Xi_e(\theta^e) + c'_\pi L\eta^2 P^2\Gamma_\pi \left(\|\nabla F^e(\theta^e)\|_2^2 + \sigma^2 + \zeta^2\right),\end{aligned} \quad (40)$$

for a universal constant $c'_\pi > 0$.

Assumption 6 gives

$$\mathbb{E}\big[F^e(\theta^{e+1}) \mid \mathcal{F}_e\big] \le \mathbb{E}\Big[F^e(\tilde{\theta}^{e+1}) \mid \mathcal{F}_e\Big] + \rho_{\mathrm{AG}}\eta P\|\nabla F^e(\theta^e)\|_2^2 + C_{\mathrm{AG}}\eta^3 P^3, \quad (41)$$

while Eq. equation 30 gives

$$\mathbb{E}\big[F^{e+1}(\theta^{e+1}) \mid \mathcal{F}_e\big] \le \mathbb{E}\big[F^e(\theta^{e+1}) \mid \mathcal{F}_e\big] + C_B\eta^2 P. \quad (42)$$

Combining Eqs. equation 40–equation 42, using $\Xi_e(\theta^e) \le \zeta^2$, and absorbing the gradient-dependent perturbations under the sufficiently small universal constants ρ_{AG} and $L\eta P\Gamma_\pi \le c_0$ yields

$$\begin{aligned}\mathbb{E}\big[F^{e+1}(\theta^{e+1}) \mid \mathcal{F}_e\big] \le F^e(\theta^e) &- \frac{\eta P}{8}\|\nabla F^e(\theta^e)\|_2^2 + c''_\sigma L\eta^2 P\sigma^2 \\ &+ c''_{\mathrm{het}} L^2\eta^3 P^2(P-1)\zeta^2 + c''_\pi L\eta^2 P^2\Gamma_\pi(\sigma^2 + \zeta^2) \\ &+ C_{\mathrm{AG}}\eta^3 P^3 + C_B\eta^2 P,\end{aligned} \quad (43)$$

for universal constants $c''_\sigma, c''_{\mathrm{het}}, c''_\pi > 0$. Summing over $e = 0, \dots, E-1$, telescoping, using Assumption 2, and dividing by $\eta PE/8$ gives Eq. equation 34 after absorbing constants into $c_\sigma, c_{\mathrm{het}}, c_\pi$ and Eq. equation 33.

For the fixed-objective part, use $F^e(\theta) = F_{U^e}(\theta)$ and Eq. equation 32:

$$\|\nabla F_{U^\star}(\theta^e)\|_2 \le \|\nabla F^e(\theta^e)\|_2 + \|\nabla F_{U^\star}(\theta^e) - \nabla F_{U^e}(\theta^e)\|_2 \le \|\nabla F^e(\theta^e)\|_2 + L_U\|U^e - U^\star\|_F.$$

Squaring, averaging over e, and using Eq. equation 34 gives Eq. equation 35. This transfer from round-wise stationarity to a fixed objective is standard for optimization with evolving surrogates or time-varying losses (Mairal, 2013; Razaviyayn et al., 2013; Lin et al., 2023). □

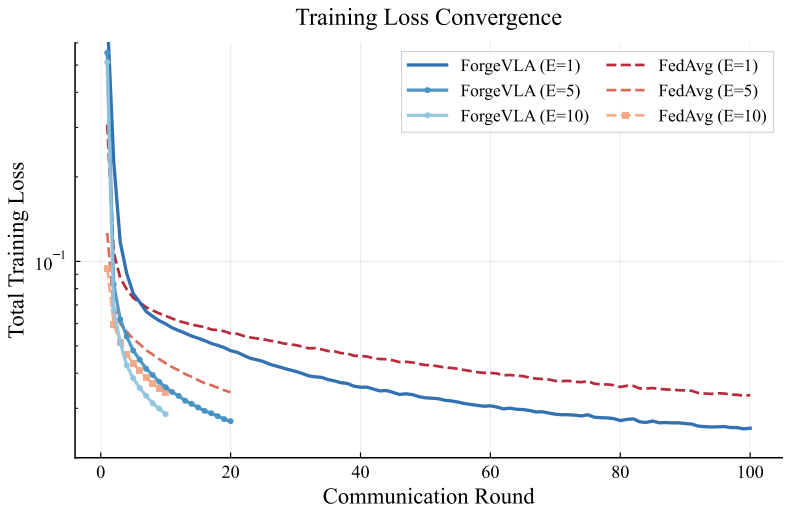


Figure 6: The training loss curves of ForgeVLA and FedAvg.

F Empirical Convergence

Figure 6 shows the training loss curves of ForgeVLA and FedAvg on the LIBERO-Goal benchmark under different training configurations. ForgeVLA achieves comparable training loss to FedAvg across all configurations, empirically validating the convergence stability of our method and complementing the theoretical analysis in Section E.

G Additional Experimental Settings

Datasets and Models: We conduct experiments on four LIBERO benchmarks (LIBERO-Goal, LIBERO-Object, LIBERO-Spatial, LIBERO-10) (Liu et al., 2023), which are widely used VLA evaluation suites (Kim et al., 2024; Qu et al., 2025; Reuss et al., 2024; Chen et al., 2025). For training data, we simulate non-i.i.d. heterogeneity via task-level partitioning, assigning s tasks per client following community practices (Li et al., 2020; Yi et al., 2024; Li et al., 2024a). All local client data retain only vision–action pairs, with a fraction p of the full annotated dataset reserved to fine-tune the embodied instruction classifier, ensuring heterogeneous client data distributions. We use the InternVLA-M1 as our backbone (Chen et al., 2025), which follows the encoder-decoder VLA architecture. For training, we apply LoRA (Hu et al., 2022) with rank r_{vla} to the VLM encoder that holds most parameters, and full fine-tuning to the action decoder. The embodied instruction classifier shares the same VLM backbone and is adapted with its own LoRA rank r_{cls}, so r_{vla} and r_{cls} are tuned independently and are reported separately in Table 3 and Table 5b.

Baselines: We compare ForgeVLA against a set of representative baselines, including Centralized, FedAvg (McMahan et al., 2017), FedProx (Li et al., 2020), FedVLA*[1], and a FedVLA* + CLIP (Radford et al., 2021) variant. Specifically, Centralized is a centralized oracle upper bound trained on full aggregated data, and FedVLA* + CLIP is enhanced with pre-trained CLIP features for better vision-language alignment.

An identical-backbone control is directly applicable to FedAvg, since FedAvg specifies an aggregation rule independently of the VLA architecture. We therefore treat the same-backbone ForgeVLA–FedAvg comparison (both 3882.72M total / 128.10M trainable) as the primary algorithmic control. For FedVLA*, the architecture (*i.e.*, instruction-oriented scene parsing with YOLOv8 and CLIP, a Transformer encoder, DGMoE, and an MLP action predictor) is integral to the method, so replacing it with ForgeVLA's backbone would change FedVLA's defining design rather than control its capacity. We instead provide a complementary trainable-parameter-matched control by equip-

[1]Since official code for FedVLA (Cui et al., 2025) is unavailable, our FedVLA baseline uses a community implementation. Link is omitted for anonymity.

ping FedVLA* with MetaCLIP-G (2510.68M total / 139.41M trainable, exceeding ForgeVLA's 128.10M), under which ForgeVLA still leads by 44.8–70.4 pp.

Metrics: We report four key evaluation metrics: (1) the number of trainable parameters, (2) the total number of model parameters, (3) task success rate, and (4) Pass@K, defined as the probability that a task succeeds at least once within K attempts.

Implementation Details: We run all experiments with $N = 10$ clients with full participation ratio of 1.0 and $s = 3$ tasks per client. Moreover, we use $E = 20$ communication rounds and set $P_{\text{ep}} = 5$ local epochs per round to avoid overfitting, with a per-client local batch size of 32. Additionally, a constant learning rate schedule with warmup is adopted for the VLM encoder (10^{-5}) and the action decoder (10^{-4}). Training uses the AdamW optimizer with zero weight decay, with hyperparameters $\alpha_{\text{CP}} = 0.2$, $\alpha_{\text{AG}} = 0.1$, $\tau = 0.07$, and $\varepsilon_{\text{AG}} = 10^{-8}$. The VLA model uses LoRA rank $r_{\text{vla}} = 32$ on the VLM encoder, and the embodied instruction classifier uses LoRA rank $r_{\text{cls}} = 4$ trained on a labeled public subset with $p = 0.2$, which is the configuration reaching 92.30% instruction accuracy in Table 3. All experiments are conducted using Nvidia H800 GPUs with CUDA 12.8, Python 3.10.19, and PyTorch 2.6.0. For other baselines, we follow the default hyperparameters from their original works or corresponding implementations. Unless otherwise specified, all experiments use the mentioned settings.

H Additional Evaluation Results

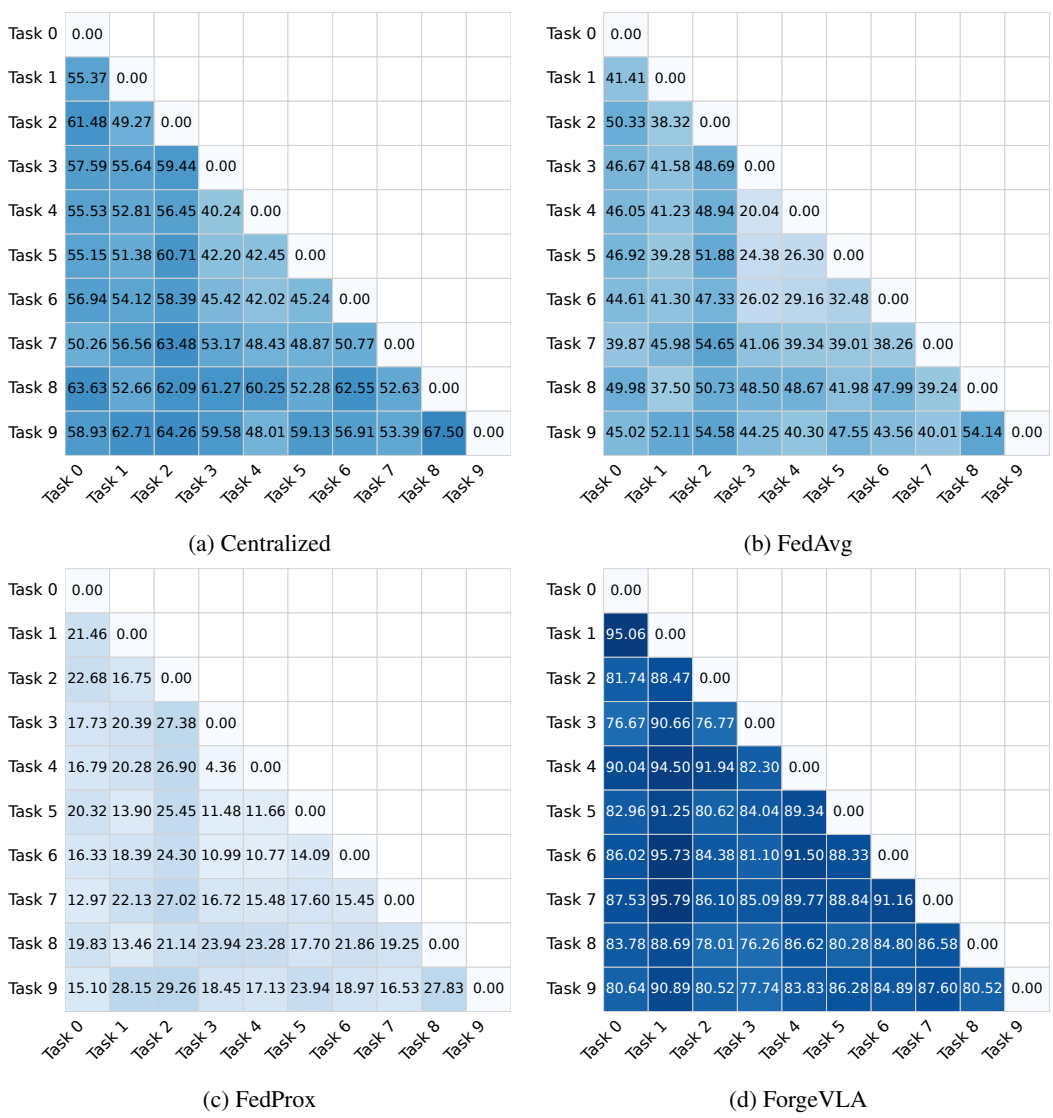
We present additional evaluation results to analyze the training efficiency and robustness of ForgeVLA under different local epoch budgets P_{ep} in Table 9. When trained with $P_{\text{ep}} = 10$ local epochs per communication round, ForgeVLA achieves performance that surpasses FedAvg with $P_{\text{ep}} = 5$ on LIBERO-Goal and is comparable to FedAvg with $P_{\text{ep}} = 1$ across all benchmarks. In contrast, vanilla FedAvg suffers from significant performance degradation as P_{ep} increases, primarily due to exacerbated client drift and vision-language feature collapse. By leveraging the proposed contrastive planning loss and adaptive aggregation strategy, ForgeVLA effectively alleviates representation collapse and conflicting local updates, maintaining stable and high-performance learning even with intensive local optimization.

Notably, in the E=100, P_{ep}=1 configuration, ForgeVLA achieves 81.2% on LIBERO-Goal, which slightly exceeds the centralized upper bound (75.8%). We attribute this to the contrastive planning loss, which explicitly encourages inter-task discriminability in the embedding space, a regularization effect absent from the standard centralized training objective. We emphasize that this result is based on a single seed and that reporting mean and standard deviation over multiple runs would provide a more reliable comparison; we leave such variance analysis for future work.

I Effect of the Contrastive Planning Loss on Feature Separation

Section 4.2 diagnoses vision–language feature collapse by contrasting centralized training with FedAvg and FedProx. Here we complete that analysis with the corresponding ForgeVLA visualization, so that the diagnosed failure mode and the proposed remedy are assessed under one identical protocol. Concretely, we repeat the procedure of Figure 3 on LIBERO-Goal while holding the client partition ($N = 10$, $s = 3$), the ground-truth task instructions, the feature-extraction procedure fixed, and varying only the training algorithm. For each model we extract the joint vision–language representation of every evaluation trajectory, average these representations within each task, and report all pairwise distances among the LIBERO-Goal tasks.

Figure 7 reveals a clear monotone ordering. The mean pairwise task distance is 54.83 under centralized training, falls to 42.60 under FedAvg, and collapses further to 19.01 under FedProx, confirming that federated training under task-level heterogeneity erodes inter-task margins and that proximal regularization aggravates rather than mitigates the effect. ForgeVLA reverses this trend and reaches a mean distance of 85.90, more than twice that of FedAvg. Moreover, the improvement is uniform rather than driven by a few outlying task pairs: ForgeVLA's *smallest* pairwise distance (76.26) already exceeds the *largest* distance under FedAvg (54.65) and under centralized training (67.50), so every pair of tasks is more distinguishable than the most separated pair produced by either reference.



(a) Centralized (b) FedAvg

(c) FedProx (d) ForgeVLA

Figure 7: Pairwise distances between task-level latent representations on LIBERO-Goal. Only the training algorithm differs across panels, and the client partitions, ground-truth task instructions, and feature extraction are held identical. Relative to centralized training (mean pairwise distance 54.83), FedAvg contracts inter-task margins (42.60) and FedProx contracts them substantially further (19.01), whereas ForgeVLA expands them well beyond the centralized reference (85.90). Because ground-truth instructions are used throughout, neither the collapse nor its mitigation is an artifact of instruction-classifier errors. All four panels share a single color scale (0 to 100), so the shading is directly comparable across panels.

These observations provide representation-level evidence that complements the downstream ablation in Table 4, in which the contrastive planning loss, the component that explicitly optimizes inter-task margins, raises LIBERO-Goal success from 28.8% to 36.2% on its own and from 49.0% to 55.2% on top of adaptive aggregation. They also suggest a mechanism for the result in Table 9, where ForgeVLA slightly exceeds the centralized upper bound under the $E{=}100$, $P_{\text{ep}}{=}1$ budget: the centralized objective contains no term that rewards task-wise discriminability, whereas ForgeVLA optimizes it directly.

J PRIVACY IMPLICATION

ForgeVLA follows the standard FL data-retention principle: raw client trajectories never leave client devices. Clients communicate model updates, as in the vanilla FedAvg workflow, together with per-task mean representations. We consider the standard honest-but-curious server threat model commonly adopted in federated learning (Kairouz et al., 2021): the server faithfully executes the protocol but may attempt to infer client information from received updates. Under this model, ForgeVLA exposes essentially the same information as FedAvg, with two additional transmissions: (i) a one-time server-to-client broadcast of the embodied instruction classifier trained on public data (which contains no client-private information) and (ii) per-task mean representations. These per-task prototypes are aggregated statistics that have been argued to carry lower reconstruction risk than full gradient updates (Tan et al., 2022), though we note they are not formally private. Consequently, existing privacy-enhancing techniques for FL, such as secure aggregation (Bonawitz et al., 2017) and differential privacy (Abadi et al., 2016), can be applied on top of ForgeVLA without modifying the core algorithm to provide stronger formal guarantees if required.

K LIMITATIONS

While ForgeVLA demonstrates strong performance across all benchmarks and validates real-world transferability, we acknowledge several limitations. First, the embodied instruction classifier maps vision–action trajectories to a fixed set of M predefined language instructions. This closed-set design works well for structured industrial and service robotics scenarios but cannot handle open-ended task descriptions. Second, ForgeVLA follows the standard honest-but-curious threat model and lacks formal privacy mechanisms. While transmitted prototypes have lower reconstruction risk than full gradients, they are not formally private and remain vulnerable to stronger gradient inversion attacks. Third, our experiments isolate task-distribution heterogeneity. Cross-embodiment, cross-environment, and cross-organization heterogeneity require action-space alignment and perception-domain adaptation and are not evaluated here. Fourth, training uses a single seed. Although each task is evaluated over 50 rollouts and gains are consistent across four LIBERO suites and two physical tasks, multi-seed mean/std would give a more reliable comparison. Fifth, a less accurate classifier injects noisy instruction labels. Dedicated noisy-label mitigation is orthogonal to this work and a promising direction. Thus, alleviating these limitations is important directions for future work.

L EXPERIMENTS COMPUTE RESOURCES

All experiments are conducted on NVIDIA H800 GPUs. Each H800 is equipped with 128 GB CPU memory and a 16-core CPU. For each ForgeVLA experiment, the training budget amounts to two H800 GPUs running for 48 hours in total.

Table 9: The performance comparisons between different methods. **bold** marks the best-performing results.

Methods	Datasets	# Params (M)	# Trainable Params (M)	Success Rate (%)	Pass@50 (%)
		$E = 100$, $P_{\text{ep}} = 1$			
Centralized	LIBERO-Goal	3882.72	128.10	75.8	100
FedAvg	LIBERO-Goal	3882.72	128.10	57.6	90
ForgeVLA	LIBERO-Goal	3882.72	128.10	**81.2**$^{\uparrow 23.6\%}$	100
Centralized	LIBERO-Object	3882.72	128.10	98.8	100
FedAvg	LIBERO-Object	3882.72	128.10	98.2	100
ForgeVLA	LIBERO-Object	3882.72	128.10	**99.4**$^{\uparrow 1.2\%}$	100
Centralized	LIBERO-Spatial	3882.72	128.10	85.8	100
FedAvg	LIBERO-Spatial	3882.72	128.10	80.6	100
ForgeVLA	LIBERO-Spatial	3882.72	128.10	**85.4**$^{\uparrow 4.8\%}$	100
Centralized	LIBERO-10	3882.72	128.10	79.0	100
FedAvg	LIBERO-10	3882.72	128.10	71.8	100
ForgeVLA	LIBERO-10	3882.72	128.10	**78.8**$^{\uparrow 7.0\%}$	100
		$E = 10$, $P_{\text{ep}} = 10$			
Centralized	LIBERO-Goal	3882.72	128.10	75.8	100
FedAvg	LIBERO-Goal	3882.72	128.10	17.6	60
ForgeVLA	LIBERO-Goal	3882.72	128.10	**41.6**$^{\uparrow 24.0\%}$	100
Centralized	LIBERO-Object	3882.72	128.10	98.8	100
FedAvg	LIBERO-Object	3882.72	128.10	94.0	100
ForgeVLA	LIBERO-Object	3882.72	128.10	**95.6**$^{\uparrow 1.6\%}$	100
Centralized	LIBERO-Spatial	3882.72	128.10	85.8	100
FedAvg	LIBERO-Spatial	3882.72	128.10	50.8	90
ForgeVLA	LIBERO-Spatial	3882.72	128.10	**68.2**$^{\uparrow 17.4\%}$	90
Centralized	LIBERO-10	3882.72	128.10	79.0	100
FedAvg	LIBERO-10	3882.72	128.10	23.4	80
ForgeVLA	LIBERO-10	3882.72	128.10	**36.4**$^{\uparrow 13.0\%}$	90