MPIRE: Explicit Manipulation Planning as a Learnable Intermediate Representation for Egocentric Hand-Motion Forecasting

Wen Wang[1,2], Ruibing Hou[1]*, Hong Chang[1,2], Shiguang Shan[1,2], Xilin Chen[1,2]

[1]Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, China

[2]University of Chinese Academy of Sciences, China

Correspondence to: Ruibing Hou houruibing@ict.ac.cn

Abstract

Forecasting dexterous hand motions from egocentric observations is fundamental to intelligent interactive systems. Existing VLM-based methods typically map observations directly to future motions, overlooking the underlying manipulation process that governs hand-object interactions. Moreover, end-to-end optimization couples manipulation learning with motion synthesis, causing motion-generation gradients to interfere with the pre-learned manipulation-aware representations. To overcome these limitations, we propose EMPIRE, a two-stage framework that introduces **E**xplicit **M**anipulation **P**lanning as an **I**ntermediate **R**epresentation for **E**gocentric hand-motion forecasting. ***Stage I: Learn to Plan***. EMPIRE first learns explicit manipulation plans from multimodal context to capture the hand-object interactions progression. ***Stage II: Learn to Act***. A motion generator synthesizes future bimanual-hand motions conditioned on frozen plan's representations, preventing motion-generation gradients from affecting manipulation planning To support our method, we further construct **EMPIRE-651K**, a bimanual hand-motion forecasting dataset comprising 650,910 training windows across 111 tasks, each paired with an explicit per-hand manipulation plan. Under identical training and evaluation protocols, EMPIRE achieves state-of-the-art forecasting accuracy,, with an MPJPE of 84.53 mm and a finger-relative error of 38.97 mm. We release the code and Dataset at https://github.com/wangwen-banban/EMPIRE.

Introduction

Forecasting future dexterous hand movements from egocentric observations is fundamental to intelligent interactive systems. It has broad applications in robot learning, virtual and augmented reality, and human–robot collaboration. With the growing availability of large-scale human videos, recent studies have explored their potential as scalable sources of semantic and physical knowledge for embodied manipulation learning (Feng et al. 2026). Unlike holistic human-motion prediction, egocentric hand-motion forecasting focuses on fine-grained bi-manual coordination and complex finger articulation during object manipulation. Such intricate dynamics make accurate long-horizon forecasting particularly challenging. Nevertheless, recent advances in large-scale egocentric datasets and vision–language foundation models have driven significant progress in this field (Grauman et al. 2022; Hoque et al. 2026; Steiner et al. 2024; Bai et al. 2025a).

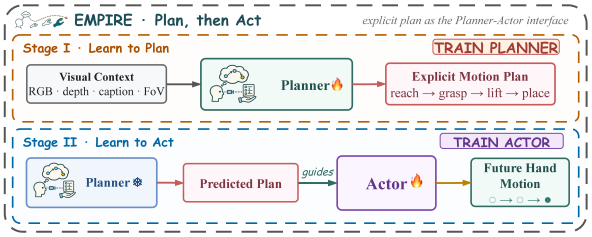


Figure 1: **EMPIRE at a glance.** Stage I learns to plan; Stage II learns to act with the planner frozen.

Hand–object motion generation has consequently shifted from text-conditioned synthesis (Taheri et al. 2022; Ghosh et al. 2023; Cha et al. 2024; Christen et al. 2024; Huang et al. 2025) towards vision–language-guided forecasting (Li et al. 2025; Luo et al. 2025; Zhou et al. 2026), where egocentric observations provide the scene, object, and viewpoint cues which are unavaiable from language alone. These approaches mainly follow two paradigms. Diffusion-based methods such as VITRA generate continuous hand motion conditioned on a VLM representation, whereas autoregressive models such as Being-H0 jointly model multimodal observations and discretized future hand motion in a unified sequence. Despite their architectural differences, these methods share a common design: directly mapping VLM representations to future hand motion without explicitly modeling the intermediate manipulation process that drives hand-motion generation.

While recent progress, existing methods remain fundamentally constrained by two overlooked design limitations. **1) Implicit supervision of manipulation planning.** Future hand movements are generated through a sequence of manipulation decisions, including object interaction, hand coordination, and task progression. Accurate forecasting therefore requires understanding not only how the hand moves but also what manipulation process should occur next. Existing methods, however, supervise the final trajectory or motion-token sequence, leaving manipulation planning to be inferred implicitly during motion prediction. Without an explicit representation of the manipulation process, generated motion may remain locally plausible while deviating from the in-

*Corresponding author.

tended interaction progression, particularly in long-horizon tasks where early planning errors accumulate over time.

2) Coupled optimization of semantic learning and motion generation. Existing egocentric hand forecasting methods typically jointly optimize the VLM backbone and the motion generation model (Brohan et al. 2023; Kim et al. 2025; Black et al. 2025). Consequently, motion-generation gradients continually update the semantic representation used for conditioning, causing the generator to adapt to a dynamically changing feature space. Meanwhile, optimizing directly for spatial motion objectives may overwrite manipulation-aware structure acquired during VLM pretraining, weakening the representation required for scene understanding and interaction anticipation. Therefore, this coupled optimization may limit the ability of VLMs to provide stable semantic guidance and ultimately constrain forecasting performance.

As summarized in Figure 1, we introduce **EMPIRE**, a two-stage method that first learns Explicit Manipulation Planning as an Intermediate Representation and then leverages it to forecast future hand motion from Egocentric-view. In ***Stage I: Learn to Plan***, a VLM predicts an explicit, motion-oriented manipulation plan from an egocentric RGB observation, RGB-derived monocular depth, a coarse instruction, and camera field of view. Monocular depth provides complementary geometric context because general-purpose VLMs remain unreliable at metric distance and complex 3D spatial reasoning (Chen et al. 2024a). Rather than predicting hand motion, the plan decomposes the anticipated interaction into temporally ordered steps for both hands, providing an interpretable intermediate representation. In ***Stage II: Learn to Act***, the entire planner is frozen, and a flow-matching DiT (Lipman et al. 2023) predicts future two-hand motion from the current hand state and the hidden representation of the predicted plan. Training on plans predicted by Stage I rather than ground-truth plans matches the condition available at inference and reduces the train–deployment gap. By preventing motion gradients from updating the planner, this decoupled design preserves manipulation-aware representations, provides stable semantic conditioning.

For training, we construct **EMPIRE-651K** from EgoDex (Hoque et al. 2026) by augmenting $650{,}910$ windows across 111 manipulation tasks with explicit per-hand manipulation plans. Extensive experiments demonstrate that EMPIRE achieves accurate and efficient egocentric hand-motion forecasting, improving both global hand placement and fine-grained finger articulation. It consistently outperforms existing methods under a unified evaluation protocol, reducing MPJPE by **19.8%** compared with VITRA while decreasing optimization time by **38.8%**. Moreover, EMPIRE achieves comparable accuracy to the much larger Being-H0-14B model with **83.5× faster** end-to-end inference.

Related Work

Vision–Language Models

Vision–language models (VLMs), enabled by visual instruction tuning and large-scale multimodal pretraining, have demonstrated strong open-vocabulary scene understanding from visual and linguistic inputs (Liu et al. 2023; Steiner et al. 2024; Chen et al. 2024b; Bai et al. 2025b,a). However, general-purpose VLMs remain limited in metric-scale perception and fine-grained 3D spatial reasoning, which are critical for spatial interaction (Chen et al. 2024a; Yang et al. 2025b; Wang et al. 2025). To bridge this gap, recent embodied systems leverage VLMs as high-level planners. For example, SayCan grounds language-model plans with environmental affordances (Ichter et al. 2023), PaLM-E enables multimodal sequential planning for embodied tasks (Driess et al. 2023), RT-H predicts language-conditioned motion intentions before execution (Belkhale et al. 2024), and Embodied Chain-of-Thought introduces intermediate reasoning over plans, subtasks, motions, and visual grounding (Zawalski et al. 2025). These studies demonstrate the potential of VLMs for embodied planning.

Hand Motion Generation and Forecasting

Hand Motion Generators connect multimodal perception to hand actions. RT-2 and OpenVLA predict robot actions from visual-language inputs (Brohan et al. 2023; Kim et al. 2025), Octo learns a generalist policy across heterogeneous embodiments (Octo Model Team et al. 2024), and π_0 integrates a pretrained VLM with a flow-based action expert for action generation (Black et al. 2025). XL-VLA further extends this paradigm by adopting a π_0-style flow-based action expert in a shared latent action space across heterogeneous dexterous robotic hands (Jiang et al. 2026). For dexterous hand-motion generation, VITRA conditions a DiT-based motion generator on learned VLM representations (Li et al. 2025), Being-H0 autoregressively predicts discretized motion tokens conditioned on multimodal observations (Luo et al. 2025), and MEgoHand combines VLM-derived motion priors, monocular depth, and a flow-matching policy for egocentric hand control (Zhou et al. 2026). These advances are enabled by increasingly large-scale interaction datasets, including GRAB with whole-body grasping and object meshes (Taheri et al. 2020), H2O and ARCTIC with hand–object manipulation sequences (Kwon et al. 2021; Fan et al. 2023), HOT3D with multi-view egocentric 3D hand-object tracking (Banerjee et al. 2025), and EgoDex with large-scale egocentric manipulation videos and tracked 3D hands (Hoque et al. 2026). Despite these advances, existing hand-motion forecasting approaches typically generate future motions directly from implicit multimodal representations, without explicitly modeling the intermediate manipulation process that governs hand-object interactions.

Method

As shown in Figure 2, EMPIRE follows a two-stage plan-then-act pipeline composed of a Planner and an Actor. The Planner generates an explicit manipulation plan from multimodal egocentric observations, and the Actor transforms the plan hidden states into future two-hand motion.

Problem Formulation

We formulate egocentric dexterous hand-motion forecasting as predicting future bimanual motion from multimodal

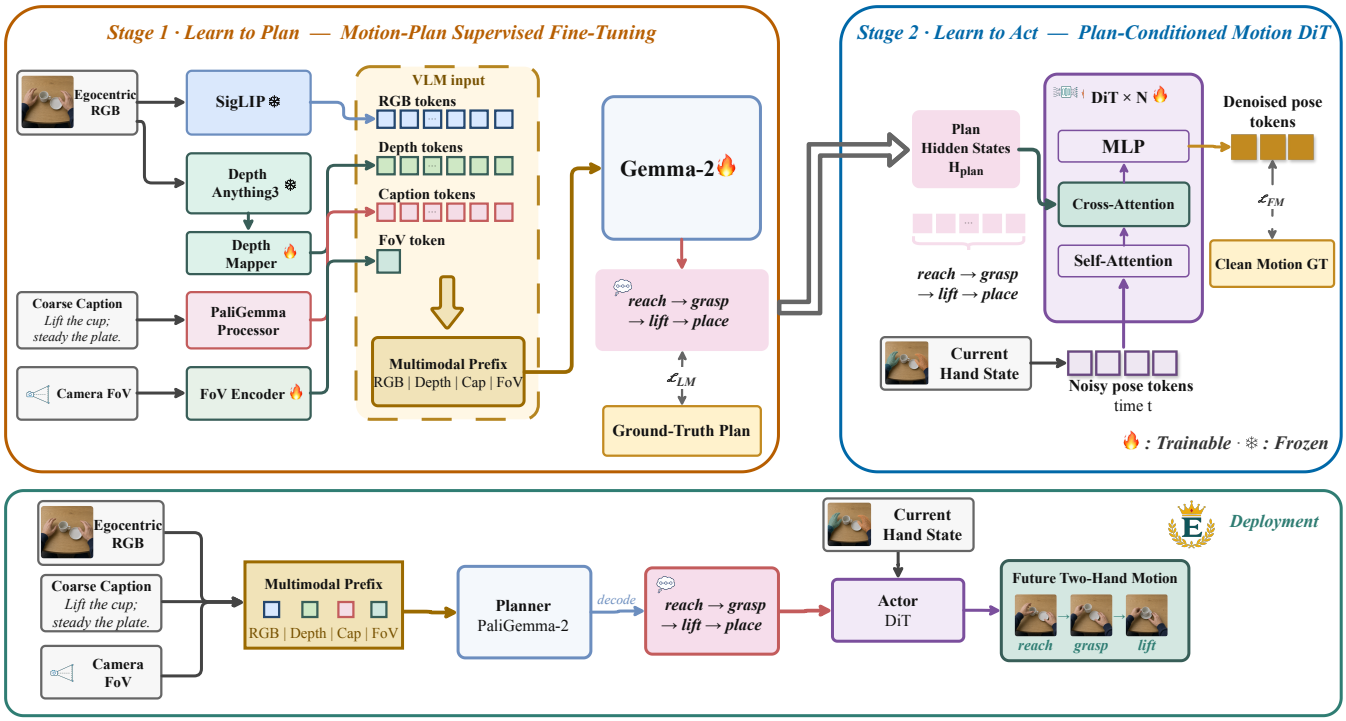


Figure 2: **Overview of EMPIRE.** *Stage I (left):* RGB and inferred monocular-depth features, together with the caption and camera FoV, form a multimodal prefix that conditions explicit motion-plan generation. *Stage II (right):* The entire Stage-I VLM is frozen, while the motion generator is trained with flow matching by conditioning on the current hand state and cross-attending to the hidden states H_{plan} of Stage-I prediction. *Bottom:* During inference, the frozen VLM generates a plan with hidden states, which directly condition the DiT to generate future motion.

egocentric observations. Let $O = (I, c, f)$ denote an egocentric RGB observation I, a coarse action caption c, and the camera field of view f. Given O and the current two-hand state s_0, the goal is to generate a future trajectory $A = (a^{(1)}, \ldots, a^{(T)}) \in \mathbb{R}^{T \times d_a}$, where T is the prediction horizon and d_a is the dimension of the two-hand pose at each step. The generated trajectory should remain continuous with the current state s_0 while consistent with the manipulation context described by O.

Directly learning $p(A \mid O, s_0)$ leaves the manipulation process that connects perception to motion implicitly. We instead introduce a structured motion plan P as an intermediate representation. The plan describes the anticipated interaction as temporally ordered, hand-specific sub-actions, while H_P denotes the hidden states of the Planner's tokens after the Planner has conditioned on O. These hidden states serve as the interface between planning and motion generation, yielding the following two-stage decomposition:

$$p(A \mid O, s_0) = \sum_P p_\theta(A \mid H_P, s_0)\, p_\phi(P \mid O), \qquad (1)$$

where $p_\phi(P \mid O)$ denotes the Planner learned in Stage I and $p_\theta(A \mid H_P, s_0)$ denotes the Actor learned in Stage II. During deployment, the latent plan variable is instantiated by the Planner prediction, whose hidden states are subsequently consumed by the Actor.

Stage I: Learn to Plan

Stage I instantiates the Planner $p_\phi(P \mid O)$ with a PaliGemma-2 VLM (Steiner et al. 2024). Given the observation O, we construct a multimodal prefix from RGB appearance (Zhai et al. 2023), inferred monocular depth feature (Lin et al. 2026), caption, and horizontal and vertical FoV:

$$Z = [Z_{\text{rgb}}; Z_{\text{depth}}; Z_{\text{cap}}; Z_{\text{fov}}], \qquad (2)$$

in a fixed order. RGB tokens provide semantic appearance cues, while depth tokens expose complementary scene geometry. Caption and FoV tokens specify the task and camera configuration, respectively.

The supervision target is a motion-oriented plan $P = (P_1, \ldots, P_{|P|})$ which decomposes the coarse action caption into temporally ordered, hand-specific sub-actions, such as reaching, stabilizing, grasping, lifting, and placing. We mask the multimodal prefix from the language loss and optimize autoregressive next-token prediction only over the plan:

$$\mathcal{L}_{\text{plan}} = -\sum_{i=1}^{|P|} \log p_\phi(P_i \mid Z, P_{<i}). \qquad (3)$$

Here, p_ϕ denotes the Planner's next-token distribution. During training, the visual and depth encoders remain frozen, while the planner and lightweight modality adapters are optimized. Therefore, Stage I learns both an explicit manipulation plan and its corresponding hidden representation, which serves as the planning interface for the Actor.

Stage II: Learn to Act

Stage II instantiates the Actor $p_\theta(A \mid H_P, s_0)$ with a flow-matching diffusion transformer. Before Actor training, the Stage-I Planner generates a predicted plan $\hat{P}$ for each training window which is cached offline as the planning condition. The ground-truth plan P is only used for Stage-I supervision. Conditioning Stage II on planner-generated plans rather than ground-truth plans eliminates the mismatch between training-time and deployment-time conditions (see the supplementary analysis of Stage II training with predicted versus ground-truth plans).

We initialize the Planner with the Stage-I learned parameters and freeze it during Stage-II training. For each Stage-II example, the frozen Planner processes the multimodal prefix together with the cached predicted plan:

$$H = [h_i]_{i=1}^{|Z|+|\hat{P}|} = \mathcal{M}_\phi(Z, \hat{P}), \tag{4}$$

where $\mathcal{M}_\phi$ denotes the VLM planner. We extract the hidden states corresponding to the generated plan span:

$$H_{\text{plan}} = [h_i]_{i=|Z|}^{|Z|+|\hat{P}|} \tag{5}$$

as the manipulation representation. A lightweight projector $\mathcal{G}_\psi$ maps this variable-length span to the Actor conditioning space, while $\mathcal{E}_{\text{state}}$ encodes the current hand state:

$$Z_{\text{state}} = \mathcal{E}_{\text{state}}(s_0), \qquad Z_{\text{plan}} = \mathcal{G}_\psi(H_{\text{plan}}). \tag{6}$$

The DiT action expert treats noisy motion tokens as queries and cross-attends to Z_{plan} as keys and values; Z_{state} supplies the initial-pose condition. Thus, the Actor generates future motion conditioned on the Planner's manipulation-aware representation rather than directly relying on an entangled image-caption feature.

We train the DiT-based Actor (Peebles and Xie 2023) and its condition projector with flow matching (Lipman et al. 2023). Given a ground-truth trajectory $A^\star$, Gaussian noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$, and $\tau \sim \mathcal{U}(0, 1)$, we construct the linear interpolation

$$A_\tau = (1 - \tau)\epsilon + \tau A^\star \tag{7}$$

and optimize the Actor to predict the velocity field:

$$\mathcal{L}_{\text{act}} = \mathrm{E}_{\epsilon, A^\star, \tau}\left[\|\mathcal{V}_\theta(A_\tau, \tau, Z_{\text{state}}, Z_{\text{plan}}) - (A^\star - \epsilon)\|_2^2\right], \tag{8}$$

where $\mathcal{V}_\theta$ denotes the Actor. The Actor predicts the velocity field over the full two-hand trajectory instead of separately modeling the two hands. Freezing the Planner prevents the motion generation objective from altering the learned planning representation, while avoiding back-propagation through the Planner.

Deployment: Plan Then Act

At deployment, the frozen Stage-I Planner first predicts the structured motion plan $\hat{P}$ online. We prefill the VLM with Z and retain the final-layer hidden state of each generated plan token, where the predefined plan delimiters identify the span used for extracting H_{plan}. The Stage-II Actor then conditions on this cached span together with s_0. The explicit plan is therefore generated once, and provides a stable, manipulation-aware representation for motion generation.

Dataset: EMPIRE-651K

To enable long-horizon egocentric bimanual motion forecasting with explicit manipulation plans, we construct **EMPIRE-651K** from the EgoDex dataset (Hoque et al. 2026). EgoDex contains 829 hours of egocentric Apple Vision Pro recordings at 30 FPS across 194 tabletop manipulation tasks, with synchronized RGB observations, camera calibration, language task descriptions, and two-hand ARKit skeleton annotations. Following the official split, we use the five training partitions for dataset construction and keep the test partition completely held out. After processing below, EMPIRE-651K contains 650,910 five-second forecasting windows across 111 manipulation tasks. The held-out test partition contains an additional 6,836 windows covering the same task vocabulary. Each sample consists of a current egocentric RGB observation, the initial two-hand state, a task caption, an explicit manipulation plan, and the future two-hand trajectory. Figure 3 shows the dataset construction.

(A) Skeleton-to-MANO conversion. EgoDex provides 21 tracked 3D hand joints but does not include a parametric hand representation required for motion forecasting. We therefore convert the original skeleton annotations into MANO-based hand representations. Specifically, we transform finger joints into the wrist coordinate frame, resolve left-hand chirality inconsistencies, and perform sequence-level MANO fitting (Romero, Tzionas, and Black 2017). The fitting optimizes a 15-dimensional PCA pose representation with a neutral mean shape $\beta{=}0$, guided by joint reconstruction and first-order temporal smoothness constraints. We set the weights for the reconstruction and smoothness terms to 100 and $0, 2$, respectively, and perform optimization using Adam (Kingma and Ba 2015) for 500 iterations with a learning rate of 0.01. The resulting MANO finger articulations are combined with the original ARKit wrist $SE(3)$ transformations, preserving both global hand motion and fine-grained finger articulation to obtain temporally consistent two-hand motion targets.

(B) Temporal window construction. EgoDex recordings may contain multiple manipulation episodes within a single video. Directly treating an entire recording as one sequence would introduce task transitions unrelated to the target action, and weaken the correspondence between observation, instruction, and future motion. We therefore first divide each recording into task-specific episodes and further partition each episode into non-overlapping five-second windows. Each window contains 150 frames at the original 30 FPS and is resampled to 12 FPS for motion forecasting. Under our forecasting setting, the model receives only the current egocentric observation and initial two-hand state, while predicting the following 60 frames of bimanual motion. Each example therefore requires forecasting a full five seconds of future motion, enabling the study of long-horizon manipulation evolution rather than short-term motion continuation.

(C) Caption and motion-plan annotation. To supervise explicit manipulation planning, we construct task captions and manipulation plans through a two-stage VLM-assisted annotation pipeline. Given each five-second forecasting segment, the pipeline first identifies the overall manipulation

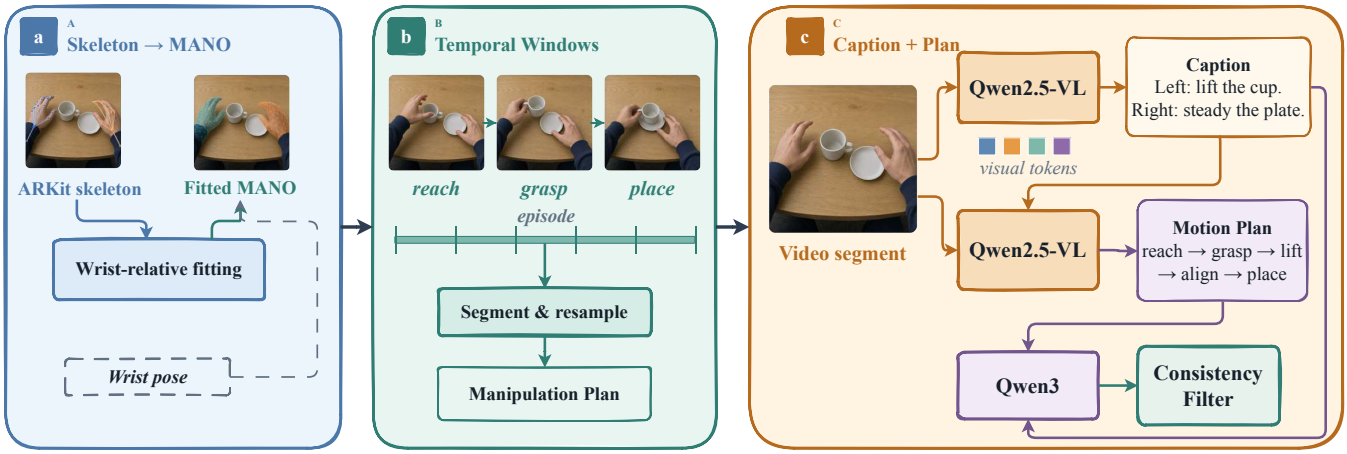


Figure 3: **The EMPIRE-651K construction and plan-supervision pipeline.** (A) Raw ARKit skeletons are converted into wrist-aligned MANO motion targets. (B) Multi-task EgoDex videos are split into task-specific episodes, which are then segmented and resampled into aligned forecasting windows. (C) Qwen2.5-VL first captions each video segment and then generates a caption-grounded motion plan from the segment and caption; Qwen3 labels caption–plan consistency, and inconsistent samples are discarded.

intent and then decomposes it into temporally ordered, hand-specific sub-actions. **i) Caption generation.** For each five-second segment, Qwen2.5-VL-7B-Instruct (Bai et al. 2025b) observes video frames sampled at 3 FPS and generates a concise caption describing the dominant manipulation intent and involved interactions. The caption provides coarse semantic guidance without requiring detailed descriptions of hand motion evolution. **ii) Manipulation plan generation.** Conditioned on the video segment and generated caption, Qwen2.5-VL produces a structured manipulation plan that decomposes the interaction into temporally ordered, hand-specific sub-actions, such as reaching, grasping, stabilizing, lifting, and placing. The caption acts as a semantic constraint to maintain consistency with the observed task and reduce unsupported actions. **iii) Annotation filtering.** To improve annotation reliability, Qwen3-30B-A3B-Instruct-2507 (Yang et al. 2025a) evaluates the consistency between each caption and its corresponding manipulation plan using binary labels. Samples with inconsistent annotations are removed. The detailed prompts for caption generation, plan construction, and consistency evaluation are provided in the supplementary material under *Data-Annotation Prompts*.

Human annotation audit. To evaluate annotation quality, we conduct a human audit over randomly sampled training windows. We select 20 samples from each EgoDex training partition, resulting in 100 manually inspected cases from distinct manipulation episodes. Each sample is evaluated according to six criteria, including caption grounding, hand attribution, plan grounding, plan coverage, temporal coordination, and usefulness for motion forecasting. All sampled videos are successfully interpretable. As shown in Table 1, the final annotation quality reaches 90.41/100, and 84.0% of caption-plan pairs are considered suitable for downstream learning. These results indicate that the generated captions and manipulation plans are sufficiently grounded in the observed interactions and provide reliable supervision for learning the planning interface between perception and motion generation. Detailed evaluation protocols and criterion definitions are provided in the supplementary material

Metric	Score
Caption quality	4.57
Motion-plan quality	4.66
Overall quality	**4.62**
Usable pairs	**84.0%**

Table 1: **Human audit of EMPIRE-651K annotations.** Final results over 100 stratified training windows. Quality scores use a 1–5 scale; A pair is usable when both its caption and plan means are at least $4/5$.

Experiment

Training

Data splits. The final EMPIRE model and our re-implemented VITRA baseline are trained on all five training partitions of EMPIRE-651K, comprising 650,910 forecasting windows from 111 manipulation tasks. The held-out test partition is excluded from all training runs.

Implementation Details. Our backbone is PaliGemma-2-3B with a 182M DiT-Base flow-matching generator. The model is trained following the two-stage framework. In Stage I, the planner model is fine-tuned for 1 epoch. In Stage II, the VLM is frozen and only the DiT action generator is optimized for 4 epochs. Training is performed with a total batch size of 64 across 8 A40 GPUs, more details in the Supplementary material.

Evaluation Protocol. All models are evaluated on the held-out test partition containing 6,836 five-second forecast-

Method	Params	MPJPE ↓	Wrist MPJPE ↓	Finger- relative MPJPE ↓	Infer.(s/window) ↓
VITRA re-impl. (Li et al. 2025)	3.2B	105.37	91.42	<u>45.37</u>	**0.4**
Being-H0-1B (Luo et al. 2025)	1.2B	115.61	100.23	66.87	15.6
Being-H0-8B (Luo et al. 2025)	8.2B	85.39	<u>68.61</u>	50.95	28.3
Being-H0-14B (Luo et al. 2025)	15.4B	<u>84.90</u>	**66.54**	51.65	83.5
EMPIRE (ours)	3.2B	**84.53**	73.25	**38.97**	<u>1.0</u>

Table 2: **Main results on EMPIRE-651K.** EMPIRE is compared with baselines on the same held-out test set. See the Evaluation subsection for protocol details and metric definitions.

ing windows from 111 manipulation tasks. The same test set is used for models trained on all five partitions and for all Part 1 ablations. For Part 1 ablations, test windows are further categorized as *seen* or *unseen* based on task overlap with the training set. Specifically, 1,247 windows from the 26 training tasks are labeled as seen, while 5,589 windows from 85 unseen tasks are labeled as unseen. This split is only used for Part 1 ablations due to their limited training task coverage. And since the motion generator is stochastic, we sample $K{=}8$ trajectories per input and report best-of-K performance following prior motion-generation works (Li et al. 2025; Luo et al. 2025). Each trajectory is generated with 4 Euler steps, using the same sampling protocol for all models.

Metrics. All methods are evaluated using MPJPE and its variants. Errors are reported in millimeters, with lower values indicating better performance. Unless otherwise specified, global-coordinate metrics are computed in the absolute camera coordinate frame. **i) MPJPE** measures the average Euclidean distance over both hands, all valid prediction frames, and all MANO joints. Under the best-of-8 setting, we select the trajectory with the lowest MPJPE for each test sample and compute all other metrics based on the selected trajectory. **ii) Wrist MPJPE** measures global hand placement accuracy by averaging the error of wrist joints over time. **iii) Finger-relative MPJPE** removes wrist translation before computing joint errors to evaluate finger articulation independent of global hand motion.

Main Results

Forecasting accuracy. Table 2 compares EMPIRE with VITRA (Li et al. 2025) and Being-H0 (Luo et al. 2025). EMPIRE achieves the best overall performance among the evaluated methods, reducing MPJPE by 19.8% compared with VITRA while using a substantially smaller model than the larger Being-H0 variants. Although the largest Being-H0 model achieves better wrist localization, EMPIRE obtains substantially more accurate finger articulation, demonstrating the advantage of explicit manipulation planning for fine-grained hand motion forecasting over long horizons. Figure 4 shows the same trend qualitatively: EMPIRE remains stable over the long horizon, while the other methods are relatively stable early but degrade at later steps. Additional cases are provided in the supplementary material under *Qualitative Hand-Motion Forecasts*.

Training and inference efficiency. Beyond forecasting accuracy, EMPIRE also achieves improved efficiency. Training the complete two-stage framework requires 71 hours on 8 A40 GPUs, compared with 116 hours for the VITRA re-implementation, reducing training time by 38.8%. During inference, EMPIRE generates each forecasting window in 1.0 second, substantially faster than Being-H0 models. In particular, EMPIRE is $83.5\times$ faster than Being-H0-14B in inference, while achieving comparable overall MPJPE and substantially better finger articulation accuracy. Detailed latency analysis is provided in the supplementary material under *Inference Cost*.

Difficulty (baseline MPJPE)	#tasks	Mean Δ	Plan helps
Easy (< 100)	43	-4.6	$25/43$
Medium (100–175)	49	-26.2	$47/49$
Hard (≥ 175)	19	-49.7	$17/19$
All	111	-22.3	$89/111$

Table 3: **Planning benefit by task difficulty.** Δ means that ours minus baseline; Negative means improvement.

When does motion planning help? To investigate when explicit planning is most beneficial, we analyze planning gains across different task difficulties. Table 3 groups test tasks according to their baseline MPJPE and shows that the benefit of planning increases with task complexity. The average MPJPE reduction grows from 4.6 mm on easy tasks to 49.7 mm on hard tasks, indicating that explicit plans are particularly effective for challenging manipulation scenarios. Overall, planning improves 89 out of 111 tasks, achieving an average MPJPE reduction of 22.3 mm. This trend is further supported by the negative correlation between baseline MPJPE and planning gain ($r = -0.567$) (Pearson 1895), suggesting that tasks with larger initial errors benefit more from explicit planning. Furthermore, planning generalizes beyond the training task categories, improving 74 out of 85 unseen tasks and 15 out of 26 seen tasks. These results suggest that explicit manipulation plans provide structured temporal guidance, which is particularly valuable for long-horizon and compositional manipulation tasks.

Ablation and Discussion

To ensure computationally efficient and controlled ablation studies, we conduct all ablations on 26-task Part 1 subset. This setting enables evaluation of both in-distribution per-

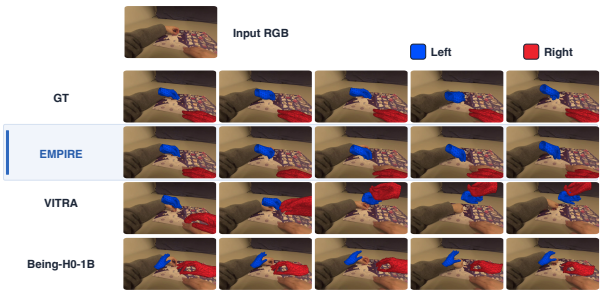


Figure 4: Qualitative comparison of motion predictions against baselines. `Task: peel_place_sticker`.

Configuration	Overall ↓	Seen ↓	Unseen ↓
VITRA re-impl. (caption only)	124.52	92.71	129.73
+ motion plan	102.26	**90.87**	104.80
+ depth (cross-attention)	104.71	110.10	103.51
+ **depth (concatenation)**	**95.42**	99.33	**94.55**

Table 4: **Model-component ablation.** All results in the table are reported using MPJPE.

formance and generalization to the 85 tasks excluded from ablation training. Unless otherwise specified, each ablation modifies only the component indicated by its name, while keeping the input representation, motion targets, optimization schedule, and evaluation protocol fixed.

Model components. Table 4 evaluates the contribution of explicit motion plans and monocular depth features. Starting from the VITRA re-implementation with caption-only conditioning, introducing the explicit motion plan reduces overall MPJPE from 124.52 mm to 102.26 mm. The improvement is particularly large on unseen tasks, where the error decreases from 129.73 mm to 104.80 mm, suggesting that the plan provides transferable temporal guidance beyond the semantic information contained in the caption alone. This result highlights the importance of explicitly modeling the intermediate manipulation process rather than directly mapping observations and instructions to future hand motions.

We further evaluate different strategies for incorporating depth information. Direct depth concatenation achieves the best performance, reducing overall MPJPE to 95.42 mm, while cross-attention over the same depth features does not improve over the plan-only model. This indicates that geometric cues are more effective when directly integrated into the multimodal representation. Interestingly, the depth-augmented model mainly benefits unseen tasks, suggesting a trade-off between cross-task generalization and fitting to the smaller ablation training subset.

Preventing motion gradients from updating the VLM. Table 5 investigates whether isolating motion gradients is necessary beyond the two-stage training strategy. In the two-stage trainable setting, updating the VLM with motion gradients results in an MPJPE of 118.44 mm. In contrast, freezing the VLM after Stage I and optimizing only the motion generator reduces MPJPE to 94.67 mm, improving by 23.77 mm (20.1%). This result indicates that freezing is not merely a constraint, but preserves stable semantic representations and provides a consistent interface for motion generation.

Setting	VLM grad.?	LM loss	MPJPE ↓
Single-stage	Yes	–	124.52
Two-stage, trainable	Yes	0.5	118.44
Two-stage, frozen (ours)	**No**	0	**94.67**

Table 5: **Motion-gradient isolation in Stage II.** The plan loss coefficient is set to 0.5.

Action head	Params	Overall ↓	Seen ↓	Unseen ↓
DiT-M	46M	107.93	114.15	106.55
DiT-B (default)	182M	**95.42**	99.33	**94.55**
DiT-L	637M	100.08	**95.37**	101.13

Table 6: **DiT-capacity ablation.** All results in the table are reported using MPJPE

DiT capacity. Table 6 studies the effect of actor capacity while fixing plan-span conditioning, concatenated depth features, and a frozen VLM. DiT-M consistently underperforms across all evaluation settings, indicating insufficient capacity to capture complex bimanual hand motion dynamics. Increasing the capacity to DiT-L improves performance on seen tasks but leads to degraded generalization on unseen tasks, suggesting that excessive model capacity may overfit the observed manipulation patterns under the available training data scale. In contrast, DiT-B achieves the best overall and unseen-task performance, providing a better balance between motion modeling capability and generalization. We therefore select DiT-B (182M parameters) as the default actor.

Conclusion

We presented EMPIRE, a two-stage framework that introduces an explicit motion plan as an intermediate interface between vision-language understanding and long-horizon bimanual hand-motion forecasting. Stage I learns structured, manipulation-aware plans for hand-object interactions, while Stage II trains a flow-matching motion generator conditioned on the learned plans with the VLM frozen. This decoupled optimization prevents gradients from low-level motion generation objectives from disrupting the manipulation-aware representations learned during planning. To support this task, we introduced EMPIRE-651K, a large-scale benchmark constructed from EgoDex by converting skeleton annotations into temporally aligned MANO trajectories, coarse instructions, and motion-plan supervision, covering 650,910 training windows across 111 manipulation tasks. Extensive experiments demonstrate that EMPIRE achieves state-of-the-art forecasting accuracy. These results validate explicit manipulation planning combined with decoupled motion learning as an effective paradigm for accurate and efficient egocentric dexterous-motion forecasting.

References

Bai, S.; Cai, Y.; Chen, R.; et al. 2025a. Qwen3-VL Technical Report. arXiv:2511.21631.

Bai, S.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; Song, S.; Dang, K.; Wang, P.; Wang, S.; Tang, J.; et al. 2025b. Qwen2.5-VL Technical Report. arXiv:2502.13923.

Banerjee, P.; Shkodrani, S.; Moulon, P.; Hampali, S.; Han, S.; Zhang, F.; Zhang, L.; Fountain, J.; Miller, E.; Basol, S.; et al. 2025. HOT3D: Hand and Object Tracking in 3D from Egocentric Multi-View Videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7061–7071.

Belkhale, S.; Ding, T.; Xiao, T.; Sermanet, P.; Vuong, Q.; Tompson, J.; Chebotar, Y.; Dwibedi, D.; and Sadigh, D. 2024. RT-H: Action Hierarchies Using Language. In *Proceedings of Robotics: Science and Systems*.

Black, K.; Brown, N.; Driess, D.; Esmail, A.; Equi, M. R.; Finn, C.; Fusai, N.; Groom, L.; Hausman, K.; Ichter, B.; Jakubczak, S.; Jones, T.; Ke, L.; Levine, S.; Li-Bell, A.; Mothukuri, M.; Nair, S.; Pertsch, K.; Shi, L. X.; Smith, L.; Tanner, J.; Vuong, Q.; Walling, A.; Wang, H.; and Zhilinsky, U. 2025. π_0: A Vision-Language-Action Flow Model for General Robot Control. In *Proceedings of Robotics: Science and Systems*.

Brohan, A.; Brown, N.; Carbajal, J.; et al. 2023. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, 2165–2183. PMLR.

Cha, J.; Kim, J.; Yoon, J. S.; and Baek, S. 2024. Text2hoi: Text-guided 3d motion generation for hand-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1577–1585.

Chen, B.; Xu, Z.; Kirmani, S.; Ichter, B.; Sadigh, D.; Guibas, L.; and Xia, F. 2024a. SpatialVLM: Endowing Vision-Language Models with Spatial Reasoning Capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 14455–14465.

Chen, Z.; Wu, J.; Wang, W.; Su, W.; Chen, G.; Xing, S.; Zhong, M.; Zhang, Q.; Zhu, X.; Lu, L.; et al. 2024b. Intern VL: Scaling up Vision Foundation Models and Aligning for Generic Visual-Linguistic Tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 24185–24198.

Christen, S.; Hampali, S.; Sener, F.; Remelli, E.; Hodan, T.; Sauser, E.; Ma, S.; and Tekin, B. 2024. DiffH2O: Diffusion-Based Synthesis of Hand-Object Interactions from Textual Descriptions. In *SIGGRAPH Asia Conference Papers*.

Driess, D.; Xia, F.; Sajjadi, M. S. M.; Lynch, C.; Chowdhery, A.; Ichter, B.; Wahid, A.; Tompson, J.; Vuong, Q.; Yu, T.; et al. 2023. PaLM-E: An Embodied Multimodal Language Model. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, 8469–8488. PMLR.

Fan, Z.; Taheri, O.; Tzionas, D.; Kocabas, M.; Kaufmann, M.; Black, M. J.; and Hilliges, O. 2023. ARCTIC: A Dataset for Dexterous Bimanual Hand-Object Manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12943–12954.

Feng, Z.; Li, Q.; Liang, H.; Yang, R.; Shen, Y.; Du, Z.; Zhang, Z.; Deng, Y.; Zhao, L.; Zhao, H.; Lu, Z.; Mees, O.; Pollefeys, M.; Yang, J.; and Guo, B. 2026. From Human Videos to Robot Manipulation: A Survey on Scalable Vision-Language-Action Learning with Human-Centric Data. arXiv:2606.00054.

Ghosh, A.; Dabral, R.; Golyanik, V.; Theobalt, C.; and Slusallek, P. 2023. IMoS: Intent-Driven Full-Body Motion Synthesis for Human-Object Interactions. 42(2).

Grauman, K.; Westbury, A.; Byrne, E.; Chavis, Z.; Furnari, A.; Girdhar, R.; Hamburger, J.; Jiang, H.; et al. 2022. Ego4D: Around the World in 3,000 Hours of Egocentric Video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 18995–19012.

Hoque, R.; Huang, P.; Yoon, D. J.; Sivapurapu, M.; and Zhang, J. 2026. EgoDex: Learning Dexterous Manipulation from Large-Scale Egocentric Video. In *International Conference on Learning Representations (ICLR)*.

Huang, M.; Chu, F.-J.; Tekin, B.; Liang, K. J.; Ma, H.; Wang, W.; Chen, X.; Gleize, P.; Xue, H.; Lyu, S.; Kitani, K.; Feiszli, M.; and Tang, H. 2025. HOIGPT: Learning Long-Sequence Hand-Object Interaction with Language Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7136–7146.

Ichter, B.; Brohan, A.; Chebotar, Y.; Finn, C.; Hausman, K.; Herzog, A.; Ho, D.; Ibarz, J.; Irpan, A.; Jang, E.; et al. 2023. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. In *Proceedings of the 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, 287–318. PMLR.

Jiang, G.; Liang, Y.; Ye, J.; Huang, J.-Y.; Jing, C.; Duan, R.; Abbeel, P.; Wang, X.; and Zou, X. 2026. Cross-Hand Latent Representation for Vision-Language-Action Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 13496–13507.

Kim, M. J.; Pertsch, K.; Karamcheti, S.; Xiao, T.; Balakrishna, A.; Nair, S.; Rafailov, R.; Foster, E. P.; Sanketi, P. R.; Vuong, Q.; Kollar, T.; Burchfiel, B.; Tedrake, R.; Sadigh, D.; Levine, S.; Liang, P.; and Finn, C. 2025. OpenVLA: An Open-Source Vision-Language-Action Model. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, 2679–2713. PMLR.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*.

Kwon, T.; Tekin, B.; Stühmer, J.; Bogo, F.; and Pollefeys, M. 2021. H2O: Two Hands Manipulating Objects for First Person Interaction Recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 10138–10148.

Li, Q.; Deng, Y.; Liang, Y.; Luo, L.; Zhou, L.; Yao, C.; Zeng, L.; Feng, Z.; Liang, H.; Xu, S.; Zhang, Y.; Chen, X.; Chen, H.; Sun, L.; Chen, D.; Yang, J.; and Guo, B.

2025. Scalable Vision-Language-Action Model Pretraining for Robotic Manipulation with Real-Life Human Activity Videos. arXiv:2510.21571.

Lin, H.; Chen, S.; Liew, J. H.; Chen, D. Y.; Li, Z.; Zhao, Y.; Peng, S.; Guo, H.; Zhou, X.; Shi, G.; Feng, J.; and Kang, B. 2026. Depth Anything 3: Recovering the Visual Space from Any Views. In *International Conference on Learning Representations (ICLR)*.

Lipman, Y.; Chen, R. T. Q.; Ben-Hamu, H.; Nickel, M.; and Le, M. 2023. Flow Matching for Generative Modeling. In *International Conference on Learning Representations (ICLR)*.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2023. Visual Instruction Tuning. In *Advances in Neural Information Processing Systems*, volume 36, 34892–34916. Curran Associates, Inc.

Luo, H.; Feng, Y.; Zhang, W.; Zheng, S.; Wang, Y.; Yuan, H.; Liu, J.; Xu, C.; Jin, Q.; and Lu, Z. 2025. Being-H0: Vision-Language-Action Pretraining from Large-Scale Human Videos. arXiv:2507.15597.

Octo Model Team; Ghosh, D.; Walke, H. R.; Pertsch, K.; Black, K.; Mees, O.; Dasari, S.; Hejna, J.; Kreiman, T.; Xu, C.; Luo, J.; et al. 2024. Octo: An Open-Source Generalist Robot Policy. In *Proceedings of Robotics: Science and Systems*.

Pearson, K. 1895. VII. Note on Regression and Inheritance in the Case of Two Parents. *Proceedings of the Royal Society of London*, 58: 240–242.

Peebles, W.; and Xie, S. 2023. Scalable Diffusion Models with Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 4195–4205.

Romero, J.; Tzionas, D.; and Black, M. J. 2017. Embodied Hands: Modeling and Capturing Hands and Bodies Together. *ACM Transactions on Graphics (ToG)*, 36(6).

Steiner, A.; Susano Pinto, A.; Tschannen, M.; Keysers, D.; Wang, X.; Bitton, Y.; Gritsenko, A.; Minderer, M.; Sherbondy, A.; Long, S.; Qin, S.; Ingle, R.; Bugliarello, E.; Kazemzadeh, S.; Mesnard, T.; Alabdulmohsin, I.; Beyer, L.; and Zhai, X. 2024. PaliGemma 2: A Family of Versatile VLMs for Transfer. arXiv:2412.03555.

Taheri, O.; Choutas, V.; Black, M. J.; and Tzionas, D. 2022. GOAL: Generating 4D Whole-Body Motion for Hand-Object Grasping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 13253–13263.

Taheri, O.; Ghorbani, N.; Black, M. J.; and Tzionas, D. 2020. GRAB: A Dataset of Whole-Body Human Grasping of Objects. In *European Conference on Computer Vision (ECCV)*.

Wang, X.; Ma, W.; Zhang, T.; de Melo, C. M.; Chen, J.; and Yuille, A. 2025. Spatial457: A Diagnostic Benchmark for 6D Spatial Reasoning of Large Multimodal Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 24669–24679.

Yang, A.; Li, A.; Yang, B.; et al. 2025a. Qwen3 Technical Report. arXiv:2505.09388.

Yang, J.; Yang, S.; Gupta, A. W.; Han, R.; Fei-Fei, L.; and Xie, S. 2025b. Thinking in Space: How Multimodal Large Language Models See, Remember, and Recall Spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10632–10643.

Zawalski, M.; Chen, W.; Pertsch, K.; Mees, O.; Finn, C.; and Levine, S. 2025. Robotic Control via Embodied Chain-of-Thought Reasoning. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, 3157–3181. PMLR.

Zhai, X.; Mustafa, B.; Kolesnikov, A.; and Beyer, L. 2023. Sigmoid Loss for Language Image Pre-Training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 11975–11986.

Zhou, B.; Zhan, Y.; Zhang, Z.; and Lu, Z. 2026. Megohand: Multimodal egocentric hand-object interaction motion generation. volume 38, 49464–49490.

Data-Annotation Prompts

We reproduce the three prompts used to construct and filter the caption–plan annotations in EMPIRE-651K. Specifically, Qwen2.5-VL first generates a coarse caption from the five-second egocentric video. The video and generated caption are then jointly provided to the model in a second pass to derive the corresponding motion plan. Finally, Qwen3 evaluates each caption–plan pair to assess semantic consistency and filters out inconsistent annotations.

Caption generation (user prompt).

```
Describe hand actions in this video
sequence. Focus on what left hand and
right hand are doing. Use very simple
sentences. Output format must be
exactly:
"Left hand: [action description]. Right
hand: [action description or None]."

For example:
"Left hand: Pick up the cup. Right
hand: None."
"Left hand: Pour water into the glass.
Right hand: Hold the glass."
"Left hand: None. Right hand: Close the
door."

What are the hands doing?
```

Caption-grounded motion-plan construction (user prompt).

```
You are given an egocentric hand-object
manipulation video and a coarse
caption.
Caption: {caption}
Generate one fine-grained action
decomposition of the hand actions in
this clip. The decomposition must be
consistent with both the video and the
caption.

Output exactly one XML-like tag:
<cot>1. [first visible hand-action
step] 2. [next hand-action step] 3.
[next or final hand-action step]</cot>

Rules:
- Use a numbered list inside the <cot>
tag: 1., 2., 3. and optionally 4. or 5.
- Each numbered step should describe
one clear sub-action in temporal order.
- Mention left hand and right hand when
they are visible or active.
- Use temporal words such as first,
then, while, next, finally when
helpful.
- Describe low-level hand motion,
contact, grasp, release, lift, place,
open, close, rotate, or stabilize
events when visible.
- If a hand is inactive, missing, or
captioned as None, state that briefly.
- Do not invent objects or actions that
contradict the caption.
- Keep the whole content 40 to 120
words.
- Do not use markdown bullets,
headings, or text outside <cot>.
```

Caption–plan consistency labeling (user prompt).

```
Determine whether the generated caption
and motion plan describe the same
hand-object manipulation task.

Caption:
{caption}

Motion plan:
{motion plan}

Label the pair CONSISTENT when the plan
preserves the caption's task, object,
and hand roles while providing
compatible fine-grained steps. Label it
INCONSISTENT when the plan changes the
task or object, conflicts with the
stated hand roles, or introduces an
incompatible goal.
Output exactly one label: CONSISTENT or
INCONSISTENT.
```

Samples labeled `INCONSISTENT` are removed before constructing the final aligned supervision tuples.

Human Audit of Annotation Quality

Sampling and review. The audit evaluates the quality of the training annotations. Using random seed `20260725`, we randomly sample 20 windows from each of the five training partitions. The resulting 100 cases are drawn from 100 different episodes, preventing the evaluation from being biased toward adjacent windows within the same episode. Each case follows the same temporal configuration as training, consisting of a 60-frame, 12 FPS, five-second egocentric video window. A human reviewer jointly examines the source video, caption, and ground-truth motion plan, while predicted plans are withheld during evaluation. All sampled videos contain sufficient visual evidence for reliable assessment.

Criteria and quality measurement. Each criterion is rated on a scale from 1 (highly inconsistent with the video) to 5 (fully consistent and directly usable). A score of 4 indicates that the main semantics are correctly captured with only minor issues, whereas a score of 3 indicates an evident discrepancy requiring correction. Table 7 summarizes the six human-audit criteria and their corresponding average scores. Let C denote the average score of the two caption-related criteria and P denote the average score of the four plan-related criteria. We compute the overall audit quality as:

$$Q_{\text{audit}} = \frac{C + P}{2}.$$

An annotation pair is considered usable if both $C \geq 4$ and $P \geq 4$. Table 8 shows that the overall audit quality reaches $4.62/5$, with a usable-pair rate of 84.0%. The quality remains

Target	Criterion	Definition	Score
Caption	Grounding	Described actions and objects are consistent with the visual evidence in the video.	$4.38/5$
Caption	Hand attribution	Left/right roles, hand activities, and `None` assignments are correctly identified.	$4.55/5$
Plan	Grounding	Described actions, objects, and contact states are supported by the visual evidence.	$4.69/5$
Plan	Coverage	Key sub-actions and all visibly active hands are covered.	$4.68/5$
Plan	Temporal coordination	Step order, simultaneous motion, and bimanual coordination are correct.	$4.60/5$
Plan	Conditioning utility	The plan is specific, non-redundant, complete, and useful for motion generation.	$4.35/5$

Table 7: **Human-audit criteria and final scores.**

Partition	Caption	Plan	Overall	Usable (%)
Part 1	4.45	4.51	4.48	75.0%
Part 2	4.63	4.72	4.68	85.0%
Part 3	4.80	4.71	4.76	95.0%
Part 4	4.35	4.56	4.46	75.0%
Part 5	4.63	4.80	4.71	90.0%
Overall	4.57	4.66	**4.62**	**84.0%**

Table 8: **Human-audit quality scores across training partitions.** Each partition contains 20 cases sampled from distinct episodes. The overall row reports the aggregate results over all 100 audited cases.

consistently high across all five training partitions, with overall scores ranging from 4.46 to 4.76, indicating that the audit results are not dominated by any single subset.

The structured error analysis further shows that the remaining annotation issues are primarily localized to hand attribution and plan details. The most frequent error categories correspond to caption hand/`None` attribution errors ($9/100$ cases), plan hand-role assignment errors ($8/100$ cases), and hallucinated plan steps ($6/100$ cases). No cases are identified as source-video ambiguous, containing unsupported caption details, or involving coarse, repetitive, or truncated motion plans.

Qualitative examples. Figure 5 visualizes three cases sampled from distinct training partitions. The ordered ground-truth frames demonstrate that the captions correctly identify the visible hand roles and manipulation task, while the corresponding motion plan decomposes the same interaction into temporally ordered and semantically consistent steps.

Training and Evaluation

Evaluation Protocol

All models are evaluated on the same EgoDex test set, which contains $6{,}836$ five-second forecasting windows covering all 111 tasks. Given each observation window, the model predicts 60 future frames at 12 FPS for both hands. For each test window, we generate $K{=}8$ stochastic predictions and recover the corresponding joint trajectories for evaluation. We report the *best-of-8* performance by selecting the prediction with the lowest overall MPJPE. The MPJPE is computed as the mean Euclidean distance over valid frames and joints across both hands. Wrist error is measured using joint 0, while finger-relative error is computed after subtracting the wrist position from all joints before comparison. All errors are reported in millimeters in the absolute camera coordinate frame and averaged across all test windows.

Baseline Implementations

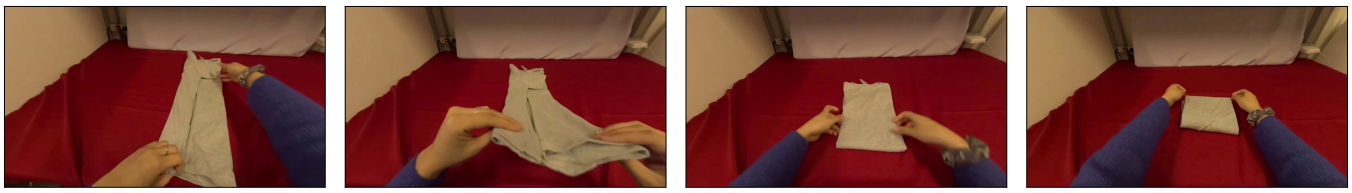
VITRA re-implementation. We adopt a re-implementation of VITRA (Li et al. 2025) as the no-plan baseline. Since the released VITRA model is trained under a different data configuration and predicts short action chunks rather than five-second future trajectories, we re-train the model under our experimental setting for a fair comparison. Our implementation follows the same architecture and training configuration as our Stage II model, including the PaliGemma-2-3B backbone, DiT action expert, EMPIRE-651K training data, four-epoch schedule, and flow-matching objective. This baseline is trained in a single stage directly from coarse captions, without explicit motion plans or monocular depth inputs, and jointly optimizes the VLM and DiT modules.

Being-H0 adaptation. We evaluate the released Being-H0 models (1B, 8B, 14B) (Luo et al. 2025) using their official inference protocol. Motion generation follows the original sampling procedure with fixed block-length and block-count control. The input consists of the current EgoDex RGB frame, warped from the original camera intrinsics to Being-H0's canonical camera space, and the same caption instruction used by our models. Since Being-H0 does not use the current hand state as input, no pose initialization is provided.

We generate 5 seconds of motion generation, (75 frames at 15 FPS). The generated wrist and finger tokens are decoded into camera-frame MANO trajectories using the GRVQ-8K tokenizer, and linearly resampled to our 60-frame, 12 FPS evaluation protocol. A same-pose alignment test verifies the consistency between Being-H0 and our MANO joint conventions, with an average residual of approximately 5 mm caused by different fingertip definitions.

To compensate for the lack of initial hand state, we additionally evaluate a wrist-anchored variant by aligning the first-frame wrist position with the ground truth. This improves Being-H0-1B from 115.61 mm to 96.55 mm, Being-H0-8B from 85.39 mm to 81.22 mm, and Being-H0-14B from 84.90 mm to 81.21 mm. While larger Being-H0 models benefit from improved global wrist localization, our lower

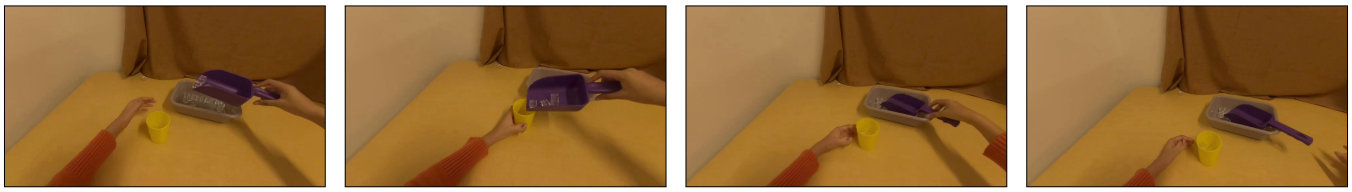
(a) Folding a cloth (Part 2; audit case #11).



Caption. Left hand: Place the folded cloth on the red surface. Right hand: Fold the cloth in half vertically.

Plan. 1. The person places a folded piece of cloth on a red surface. **2.** They fold the cloth in half vertically with both hands. **3.** They then fold the cloth in half horizontally, completing the folding process.

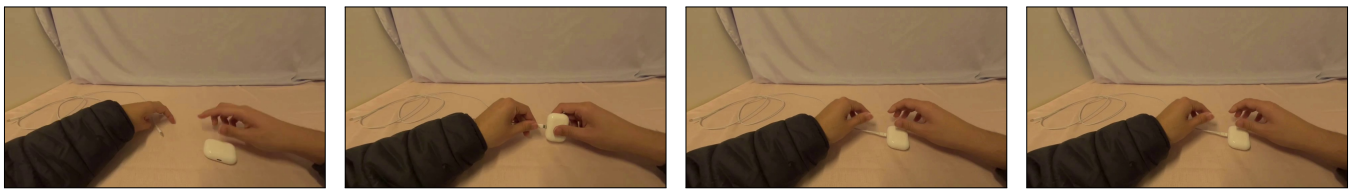
(b) Scooping ice into a cup (Part 4; audit case #43).



Caption. Left hand: Pick up the yellow cup. Right hand: Scoop ice cubes into the cup.

Plan. 1. The person reaches for a yellow cup with their left hand. **2.** They pick up a purple scoop with their right hand. **3.** The person scoops ice cubes from a container into the yellow cup. **4.** After filling the cup, they place the scoop back into the container. **5.** The person adjusts the position of the cup on the table.

(c) Connecting an AirPods charging cable (Part 1; audit case #62).



Caption. Left hand: Insert the USB cable into the white device. Right hand: Hold the white device steady.

Plan. 1. The left hand picks up the USB cable from the bed. **2.** The left hand inserts the USB cable into the white device held by the right hand. **3.** The right hand holds the white device steady while the left hand completes the insertion of the USB cable.

Figure 5: **Some annotation examples from the human audit.** Each row shows four uniformly spaced ground-truth frames from an audited five-second window, followed by its caption and temporally ordered motion-plan steps.

finger-relative error demonstrates the advantage of explicit per-hand plans for fine-grained long-horizon dexterous motion forecasting.

Training Cost

The proposed two-stage training strategy reduces optimization cost compared with the single-stage baseline. Table 10 reports the gradient-optimization time over all five EMPIRE-651K partitions. All runs use the same hardware setup (8 A40 GPUs) and batch size (64 windows per optimizer step). The offline generation of Stage-I predicted plans is treated as preprocessing and excluded.

The single-stage baseline jointly fine-tunes the 3B VLM and DiT for four epochs, requiring 40,684 steps at 10.26 s/step and 116 hours in total. In contrast, our method separates plan learning and motion generation. Stage-I updates only the 3B VLM for one epoch (10,171 steps, 34 hours), while Stage-II freezes the VLM and trains the 182M DiT for four epochs (40,684 steps, 37 hours). As a result, our complete training pipeline requires 71 hours, reducing the optimization cost by 39% compared with the baseline, despite using one additional training epoch.

The efficiency gain comes from decoupling semantic learning from motion generation: the expensive VLM backbone is optimized only during plan learning, while motion synthesis training updates only the lightweight DiT.

Inference Cost

Table 11 reports the end-to-end wall-clock time per test window under the same protocol used for the main-paper results. For each window, a single GPU generates the complete best-of-8 prediction, including preprocessing, model inference, and trajectory decoding. As shown in Table 11, our model requires only 1.0 s per window, substantially faster than Being-H0 variants (15.6–83.5 s).

The efficiency advantage comes from the different generation paradigms. Our model generates a compact manipulation plan with at most 96 autoregressive tokens, whose hidden states are reused to condition all 8 DiT samples. The VLM is therefore executed only once, while motion synthesis

Configuration	Stage I: Learn to Plan	Stage II: Learn to Act
Trainable modules	Gemma-2 decoder, multimodal projector, depth mapper, and FoV encoder; SigLIP and DA3 encoders are frozen	DiT-B and conditioning projectors; the Planner and all condition encoders are frozen
Training objective	Autoregressive plan-language cross-entropy	Flow matching with $t \sim \mathrm{Beta}(1.5, 1.0)$ on $[0, 1]$
Training length	1 epoch (10,171 optimizer steps)	4 epochs (40,684 optimizer steps)
Optimizer	AdamW, (β_1, β_2)=$(0.9, 0.999)$, weight decay 0.1	Same as Stage I
Learning rate	1×10^{-5}	1×10^{-4}
Schedule / clipping	Constant learning rate, no warmup; gradient-norm clipping at 1.0	Same as Stage I
Batch size	Global batch 64; per-device batch 2; 4 gradient-accumulation steps	Global batch 64; per-device batch 4; 2 gradient-accumulation steps
Compute	8×NVIDIA A40; bfloat16; FSDP full sharding with activation checkpointing	Same as Stage I

Table 9: **Training details.** Stage-specific settings from the final Stage I and Stage II checkpoint configurations.

Run	Trainable	Epoch	Steps	s/step	Hours
Baseline (single stage)	VLM + DiT	4	40,684	10.26	116
Ours Stage-I (plan SFT)	VLM	1	10,171	12.14	34
Ours Stage-II (DiT)	DiT	4	40,684	3.23	37
Ours total	—	5	50,855	—	**71**

Table 10: **Gradient-optimization cost on EMPIRE-651K.**

Model	AR tokens	s / window
Baseline (no plan)	0	0.4
EMPIRE (plan + depth)	≤96	**1.0**
Being-H0-1B	∼10,400	15.6
Being-H0-8B	∼10,400	28.3
Being-H0-14B	∼10,400	83.5

Table 11: **End-to-end inference cost.**

is performed through horizon-parallel flow matching with 4 Euler steps. In contrast, Being-H0 directly generates motion tokens autoregressively, requiring approximately 10,400 sequential token generations under the best-of-8 setting. Since each token depends on previous outputs, its inference cost scales with both the forecasting horizon and the number of samples.

The caption-only baseline requires only 0.4 s per window because it entirely removes autoregressive plan generation. Compared with this baseline, our additional cost is limited to generating a short manipulation plan, while avoiding the expensive autoregressive motion synthesis. These results demonstrate that explicit planning provides an effective balance between semantic guidance and inference efficiency.

Ablation Analysis

Stage II Training on Predicted Motion Plans

During deployment, Stage II is conditioned on motion plans generated by the frozen Stage I planner. Training the motion generator with ground-truth plans would introduce a train–test mismatch by providing cleaner conditioning signals than those available during inference. Table 12 compares training Stage II with ground-truth plans and with plans predicted by Stage I. For predicted-plan training, Stage I generates motion plans for training windows using the same inference setting, and Stage II is trained with these predicted plans as conditioning inputs. Since Stage I remains frozen, this only changes the conditioning data without introducing additional optimization cost. Table 12 shows that predicted-plan training consistently improves performance. On the Part 1 26-task subset, MPJPE decreases from 94.67 to 93.40 mm. When trained on all 111 tasks, the improvement increases from 90.22 to 84.53 mm. These results demonstrate that predicted-plan training better matches the inference-time conditioning distribution, and we therefore adopt Stage-I predicted plans for Stage II in the final model.

Stage-II plan condition	26-task subset	Train Part 1–5
Ground-truth plan	94.67	90.22
Stage-I predicted plan (ours)	**93.40**	**84.53**

Table 12: **Stage II training with predicted versus ground-truth plans.**

Detailed Analysis of When Motion Planning Helps?

Table 3 of the main paper compares the deployable model with self-generated-plan against the no-plan baseline on the EgoDex test set. As shown, explicit motion planning improves forecasting performance on average, but the benefit is not uniform across tasks.

Let Δ denote the per-task MPJPE difference between the planning model and the baseline (negative values indicate improvement). As shown in Table 3 of the main paper, the self-generated plan improves performance on $89/111$ tasks (80%), achieving a sample-weighted average gain of 22.3 mm and a median per-task improvement of 14.3 mm. Moreover, the benefit increases with task difficulty: baseline MPJPE and Δ exhibit a Pearson correlation of -0.567, indicating larger improvements for more challenging tasks. Consistent with the difficulty analysis in the main paper, planning provides limited gains on easy tasks ($\Delta = -4.6$ mm, improving $25/43$ tasks), but substantially larger improvements on medium (-26.2 mm, $47/49$ tasks) and hard tasks (-49.7 mm, $17/19$ tasks).

Planning also provides stronger benefits for unseen tasks. Compared with the Part 1 training setting, the self-generated plan improves $15/26$ seen tasks (58%, weighted $\Delta = -10.9$ mm) and $74/85$ unseen tasks (87%, weighted $\Delta = -24.8$ mm), yielding more than twice the improvement on unseen scenarios. The largest gains are observed on complex, long-horizon manipulation tasks, including `wash_fruit` ($343 \rightarrow 154$ mm), `stock_unstock_fridge` ($303 \rightarrow 185$ mm), `wash_put_away_dishes` ($234 \rightarrow 126$ mm), and `flip_pages` ($164 \rightarrow 58$ mm). Performance degradation is relatively rare and is primarily observed in two scenarios: simple tasks with low baseline errors, where additional planning constraints may limit motion flexibility, and highly challenging tasks where the predicted plans may still contain inaccuracies.

Qualitative Hand-Motion Forecasts

Figures 6 and 7 present qualitative comparisons of future hand-motion forecasts on two representative tasks. All predictions are rendered on the same five future RGB frames sampled from the 60-frame, 12 fps forecasting horizon. The ground-truth row uses the corresponding EgoDex MANO annotations, and all predictions are transformed into the corresponding future camera views before rendering. Blue and red denote the left and right hands, respectively.

As shown in the figures, EMPIRE better captures the temporal evolution of bimanual manipulation compared with existing baselines. In particular, it preserves more accurate hand-object interactions and finger articulations over long horizons, while VITRA and Being-H0 exhibit larger deviations in hand placement and coordination. These results demonstrate that explicit manipulation plans provide effective guidance for long-horizon hand-motion forecasting.

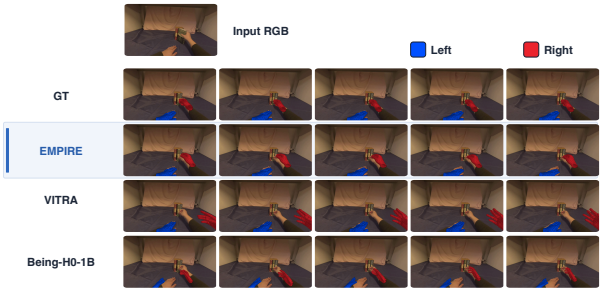


Figure 6: Qualitative comparison of motion predictions against baselines. `Task: insert_remove_bookshelf`.

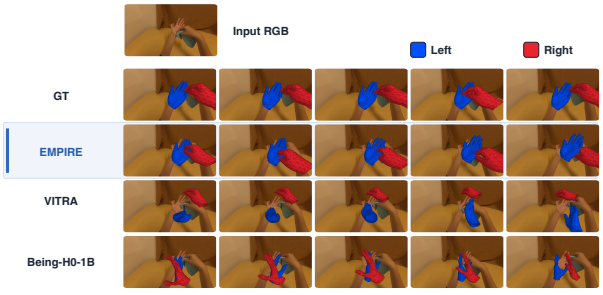


Figure 7: Qualitative comparison of motion predictions against baselines. `Task: dry_hands`.