SiMDex: Mining Similar Egocentric Videos for Cross-Embodiment Dexterous Manipulation

Anonymous authors
Paper under double-blind review

Abstract

Recent years have witnessed an explosive trend of scaling ego-centric human videos for robot manipulation, yet it remains unclear which data actually benefits dexterous manipulation. We present **SiMDex**, a **si**milarity-based data **m**ining framework that casts human data selection for VLA post-training in **dex**terous manipulation as a recommendation problem. For each robot demonstration, SiMDex runs a three-stage recall–ranking–re-ranking cascade over $\sim$32M egocentric human samples: recall and ranking select a task-relevant subset in a morphology-agnostic action space, and re-ranking verifies it on pixels, with no change to VLA architecture or training. Against a strong baseline trained with an equal amount of randomly sampled human data, SiMDex uses only $\sim$1.49M mined samples ($<$5% of the pool) yet improves the overall success rate from 47.7% to 61.7%—showing that selective curation outperforms indiscriminate data mixing.

1 Introduction

Vision-Language-Action (VLA) models have driven strong progress in robotic manipulation, with performance scaling predictably as training data grows (Brohan et al., 2023b;a; Black et al., 2025b;a; Kim et al., 2024). Yet scaling robot data is itself the bottleneck: the high cost, poor portability, and teleoperation expertise of robot hardware drive researchers to seek other ways to scale up manipulation data (Open X-Embodiment Collaboration, 2024; Khazatsky et al., 2024; Ohkawa et al., 2026). Among various approaches, ego-centric human manipulation data has emerged to be a promising data source for scaling up generalist robot policies. Large-scale egocentric datasets (Grauman et al., 2022; Damen et al., 2022; Li et al., 2026; Punamiya et al., 2026) already span thousands of hours and keep growing as wearable devices spread and embodied AI drives demand.

From thousands of hours (Grauman et al., 2022) to tens of thousands of hours (Zheng et al., 2026) to a million hours (Deng & Zhou, 2026), the vast expansion of human ego-centric manipulation data now casts new questions: **how can we use human ego-centric data more efficiently?** In practice, egocentric pools are extraordinarily heterogeneous—spanning cooking, socializing, sports, and countless other activities—of which only a small fraction is relevant to any given manipulation task. Naively training on all of it indiscriminately injects noise and dilutes task-relevant supervision, while discarding human data entirely wastes the cross-embodiment knowledge acquired during large-scale pre-training and the generalization it brings to real-world robot deployment.

In large-scale vision and language modeling, targeted data curation has proven essential to unlocking the potential of web-scale corpora (Gadre et al., 2023; Abbas et al., 2023; Penedo et al., 2024). Carefully selected subsets often match or surpass models trained on far larger unfiltered data; robot learning from human video now stands at the same crossroads. Once a VLA has been pre-trained on large-scale egocentric videos, *which* human data to revisit for the target task during post-training matters more than *how much*. The key is not to scale indiscriminately, but to select precisely.

Data curation has received growing attention in robot learning, yet existing methods operate at a different scope from ours. One line filters or reweights demonstrations within robot datasets—e.g., CUPID (Agia et al., 2025) and Re-Mix (Hejna et al., 2024)—or retrieves behaviors from offline robot data, all operating entirely within robot data rather than mining an external human pool. Closest to ours, recent work retrieves task-relevant human videos to aid policy learning (Zhu & Feng,

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.

2025; Papagiannis et al., 2025), but operates over small video banks and feeds retrieved cues as auxiliary inputs or in-context exemplars, requiring changes to the policy architecture. What remains unaddressed is task-aware, per-demonstration retrieval at the scale of tens of millions of cross-embodiment samples that, based on fine-grained action-level similarity, directly augments VLA training without modifying the policy architecture or the training procedure.

We present SiMDex, a similarity-based data mining framework that casts human data selection for VLA post-training as a recommendation problem: as in industrial recommenders that retrieve from billion-scale catalogs, the tens-of-millions-sample human pool makes exhaustive comparison infeasible. SiMDex adopts a three-stage recall–ranking–re-ranking cascade that progressively distills task-relevant subsets for each robot demonstration. Recall and ranking produce the mined subset, while re-ranking operates on pixels and audits that subset with a signal that shares no information with the two action-space stages. The cascade operates in a morphology-agnostic action space that requires no architectural change—shaping *what* a VLA learns from, not *how*. Notably, SiMDex re-mines the same egocentric corpus used for pre-training, revisiting it task-aware rather than collecting new data. Large-scale egocentric collection pays off twice, once in breadth and once in precision.

Across three real-world dexterous manipulation tasks, SiMDex improves the overall success rate from 47.7% to 61.7% using fewer than 5% of the available human samples, and it leads the baseline at every robot-data budget we test, from a quarter to twice the default amount.

In summary, our main contributions are threefold:

- **Recommendation view.** We reframe human data selection for VLA post-training as a recommendation problem, re-mining the *same* egocentric corpus used for pre-training—now task-aware—so that large-scale collection pays off twice: in breadth, then in precision.
- **Simple, scalable mining pipeline.** We instantiate this view as a three-stage cascade: recall and ranking mine a training subset from tens of millions of samples in a morphology-agnostic action space, and re-ranking verifies it on pixels, with no change to the VLA.
- **Empirical gains.** On real-world dexterous manipulation tasks, SiMDex substantially improves overall success over an identical baseline trained on equal randomly-sampled human data, and the gain holds at every robot-data budget from $0.25\times$ to $2\times$ the default.

2 Related Works

VLA Models for Robotic Manipulation. Recent VLA models achieve strong instruction-conditioned manipulation through large-scale pre-training (Brohan et al., 2023b;a; Octo Model Team et al., 2024; Kim et al., 2024; Chi et al., 2024; Li et al., 2024; Black et al., 2025b;a; Amin et al., 2026; Cheang et al., 2025), yet most remain limited to low-DoF grippers. Extending them to high-DoF dexterous hands is far harder due to the expanded action space and contact-rich dynamics (Wang et al., 2024; Handa et al., 2020). Recent efforts use hierarchical planning–control (Zhong et al., 2026; Yuan et al., 2025) or end-to-end prediction (Wen et al., 2025a; Luo et al., 2025), but both are bottlenecked by the scarcity of real dexterous data, as teleoperation (Wang et al., 2024; Wen et al., 2025b) yields datasets orders of magnitude smaller than gripper counterparts. Rather than scaling robot-data collection, SiMDex takes a complementary, model-agnostic approach: mining task-relevant human egocentric data to augment VLA training without architectural change.

Learning from Human Videos. Human videos have been leveraged for robot learning, from visual representation pre-training (Nair et al., 2022; Ma et al., 2023; Majumdar et al., 2023) to direct policy learning (Bahl et al., 2022; Wang et al., 2023; Kareer et al., 2025) and VLA pre-training from egocentric videos (Yang et al., 2025; Li et al., 2025), with another line bridging the embodiment gap via domain-adaptive imitation (Punamiya et al., 2025), data editing (Lepert et al., 2026), robot-data generation (Mu et al., 2026), or shared action-space design (Chen et al., 2026). Large-scale benchmarks such as Ego4D (Grauman et al., 2022), Ego-Exo4D (Grauman et al., 2024), and EgoDex (Hoque et al., 2026) provide rich foundations, and Zheng et al. (2026) show a log-linear scaling law between human data scale and validation loss—though their mid-training requires aligned human–robot play under the robot's task distribution, making data–task correspondence a manual prerequisite. More broadly, these methods consume human data at the dataset level without tailoring selection to the downstream task (Nair et al., 2022; Radosavovic et al., 2022; Zheng et al., 2026; Luo et al., 2025;

Kareer et al., 2025), inheriting noise from irrelevant activities. SiMDex instead performs task-aware hierarchical retrieval to identify the most relevant human demonstrations for VLA training.

Data Curation for Robot Learning. Data curation is central to foundation models: in vision and language, filtered or deduplicated subsets match or surpass models trained on larger corpora (Gadre et al., 2023; Abbas et al., 2023; Tirumala et al., 2023), and similar-sample mining benefits representation pre-training (Lin et al., 2025). Within robot learning, curation has so far stayed confined to robot data—estimating demonstration influence (Agia et al., 2025), optimizing mixing weights (Hejna et al., 2024), or retrieving behaviors from robot datasets (Du et al., 2023). A few methods do retrieve human videos (Zhu & Feng, 2025; Papagiannis et al., 2025), but use them as auxiliary or in-context inputs over limited collections. To our knowledge, no prior approach offers task-aware, per-demonstration retrieval that mines a massive cross-embodiment pool by fine-grained action-level similarity and feeds the result directly into VLA training without touching the architecture. SiMDex fills this gap by framing cross-embodiment selection as a retrieval problem: like industrial recommenders over billion-scale catalogs, we adapt a recall–ranking–re-ranking cascade to mine a task-relevant subset from tens of millions of samples. Re-ranking verifies retrieval quality on pixels via a motion signal shared by both embodiments, independent of the action space.

3 METHOD

We propose SiMDex, a similarity-based data mining framework that retrieves high-quality human data from large-scale egocentric video to enhance VLA training for dexterous manipulation. Operating over a pool of tens of millions of human samples, SiMDex deliberately adopts a lightweight, scalable similarity cascade rather than a heavy learned cross-embodiment alignment module: at this scale, a fast and reliable retrieval pipeline is what makes task-aware mining practical.

3.1 UNIFIED MORPHOLOGY-AGNOSTIC REPRESENTATION

A prerequisite for cross-embodiment mining is a representation that abstracts away the morphological gap between robot and human hands. We adopt a fingertip-centric geometric representation that captures task-relevant contact structure while remaining agnostic to joint-level kinematics, and map both sources onto it through one shared wrist-local retargeting procedure.

Wrist-local retargeting. For each source, we recover per-frame wrist poses and fingertip positions in 3D. Letting $\boldsymbol{T}_t \in \mathbb{R}^{4\times4}$ be the wrist pose at time t, with rotation $\boldsymbol{R}_t \in \mathbb{R}^{3\times3}$ and translation $\boldsymbol{o}_t \in \mathbb{R}^3$, and $\boldsymbol{q}_t \in \mathbb{R}^3$ a fingertip position in world coordinates, we re-express each fingertip relative to its wrist as $\boldsymbol{q}_t^{\text{loc}} = \boldsymbol{R}_t^\top(\boldsymbol{q}_t - \boldsymbol{o}_t) \in \mathbb{R}^3$, isolating intrinsic grasp geometry from workspace location. The five wrist-local fingertips of one hand form a fingertip configuration $\boldsymbol{p}_t \in \mathbb{R}^{15}$. The wrist action $\boldsymbol{d}_t \in \mathbb{R}^6$ gives a future wrist pose in the current wrist frame, with translational part $\boldsymbol{R}_t^\top(\boldsymbol{o}_{t+k} - \boldsymbol{o}_t) \in \mathbb{R}^3$ and rotational part the axis-angle vector of $\boldsymbol{R}_t^\top \boldsymbol{R}_{t+k}$ for the k-th future step. Together, $(\boldsymbol{d}_t, \boldsymbol{p}_t)$ form the shared descriptor onto which both embodiments are mapped. We write $\boldsymbol{p}_t$ for the observed configuration and $\boldsymbol{f}_t \coloneqq \boldsymbol{p}_{t+k}$ for the one predicted at the same future step.

Unified action. After preprocessing, both sources converge to the same per-timestep description. Its embodiment-shared part concatenates bimanual wrist actions $\boldsymbol{a}_t^{\text{wr}} = (\boldsymbol{d}_t^{\text{L}}, \boldsymbol{d}_t^{\text{R}}) \in \mathbb{R}^{12}$ and bimanual fingertip targets $\boldsymbol{a}_t^{\text{ft}} = (\boldsymbol{f}_t^{\text{L}}, \boldsymbol{f}_t^{\text{R}}) \in \mathbb{R}^{30}$, 42 embodiment-shared dimensions in total. Because both are expressed relative to the current wrist, these dimensions are invariant to workspace location and require no matching of kinematic structures across embodiments. They are the basis for both mining (Sec. 3.2) and supervision (Sec. 3.3). Its robot-specific part carries the mechanism-level commands that have no human counterpart, arm-joint actions $\boldsymbol{a}_t^{\text{ar}} \in \mathbb{R}^{14}$ and hand-joint commands $\boldsymbol{a}_t^{\text{hd}} \in \mathbb{R}^{32}$, giving $\boldsymbol{a}_t = (\boldsymbol{a}_t^{\text{wr}}, \boldsymbol{a}_t^{\text{ar}}, \boldsymbol{a}_t^{\text{hd}}, \boldsymbol{a}_t^{\text{ft}}) \in \mathbb{R}^{88}$. These sizes are those of our platform (App. D.3); SiMDex relies only on the split between shared and robot-specific blocks.

Sources. Robot demonstrations are collected by bimanual teleoperation of a two-arm, two-hand platform, from which wrist poses, fingertip positions, and joint readings are recovered via forward kinematics. Human demonstrations are drawn from EgoDex (Hoque et al., 2026), an egocentric video dataset with tracked wrist and fingertip transforms; we transform trajectories into a body-centered frame to remove ego-motion, filter by hand visibility and velocity, and apply temporal

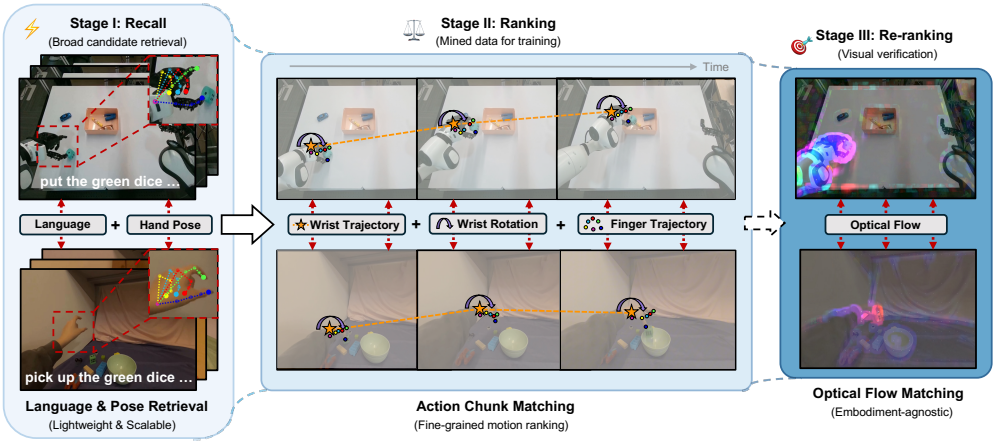

Figure 1: **Overview of the SiMDex pipeline.** Robot demonstrations (anchors, top) are matched against human candidates (bottom) through a three-stage cascade of increasing cost, denoted by deepening shades. The cascade **recalls** by language and hand pose, **ranks** by action-chunk similarity, and **re-ranks** by optical flow for verification. Orange path: wrist position; purple arc: wrist rotation; colored dots: fingertip trajectories; red arrows: anchor–candidate comparison.

smoothing before the same wrist-local retargeting (App. B.1). We then randomly mirror left and right hands to remove handedness bias. Each episode is sliced into frame-level samples, each pairing the current observation and a language instruction with a future target of $H = 30$ steps.

3.2 Similarity-Based Data Mining

Let $\mathcal{D}_r = \{\tau_n\}_{n=1}^{N}$ be the robot samples (anchors) and $\mathcal{D}_h = \{\tau_m\}_{m=1}^{M}$ the human pool (targets), with $M \gg N$. Each sample τ pairs a language instruction with the observation and future action chunk of Sec. 3.1, and retrieval runs per anchor. Our goal is to retrieve the subset $\mathcal{D}_h^* \subset \mathcal{D}_h$ whose members are most similar to the anchors in hand configuration, language semantics, and motion dynamics. Since exhaustive comparison over a pool of this size is infeasible, we adopt a three-stage cascade of increasing precision and cost (Fig. 1; App. A). Stage I scans the full pool with lightweight signals and Stage II re-scores only its candidates, together producing the mined subset used for training; Stage III operates on pixels and provides embodiment-agnostic verification.

Stage I: Recall. The recall stage retrieves a broad candidate set from two lightweight signals. For *pose*, we ℓ_2-normalize the initial fingertip configuration, removing hand scale, into a descriptor $\hat{\boldsymbol{p}} \in \mathbb{R}^{15}$ and retrieve by Euclidean distance, capturing similar initial grasp posture. For *language*, we encode each instruction into an embedding $\boldsymbol{e} \in \mathbb{R}^{384}$ with an off-the-shelf sentence encoder and retrieve by cosine similarity, capturing task semantics. The two rankings are fused by reciprocal-rank fusion into a candidate set per anchor. Both signals are cheap enough to index the full pool.

Stage II: Ranking. The ranking stage refines candidates by fine-grained motion similarity over the future action chunk, along four components derived from two trajectories: (i) the wrist trajectory $\boldsymbol{F}_{\mathrm{wr}} \in \mathbb{R}^{31\times 3}$ of one hand, the current position followed by the 30 future ones, which preserves spatial structure; (ii) a translation waveform $\boldsymbol{w}_{\mathrm{tr}} \in \mathbb{R}^{10}$ of the same hand, the per-step translation speed resampled from 30 steps to 10; (iii) a rotation waveform $\boldsymbol{w}_{\mathrm{rot}} \in \mathbb{R}^{10}$, the per-step rotation speed obtained the same way from the wrist rotations; and (iv) the fingertip trajectory $\boldsymbol{F}_{\mathrm{ft}} \in \mathbb{R}^{30\times 15}$, the 30 future fingertip targets of one hand. Each component is scored per hand and summed over the two hands, then sorted, giving four per-component ranks $r_{\mathrm{wr}}, r_{\mathrm{tr}}, r_{\mathrm{rot}}, r_{\mathrm{ft}}$; the fused score sums them with equal weight, $\rho = r_{\mathrm{wr}} + r_{\mathrm{tr}} + r_{\mathrm{rot}} + r_{\mathrm{ft}}$. We then deduplicate by source trajectory, keeping only the highest-ranked entry per source episode. This prevents any single episode from dominating the mined subset $\mathcal{D}_h^*$, and the deduplicated output is what we use for VLA training.

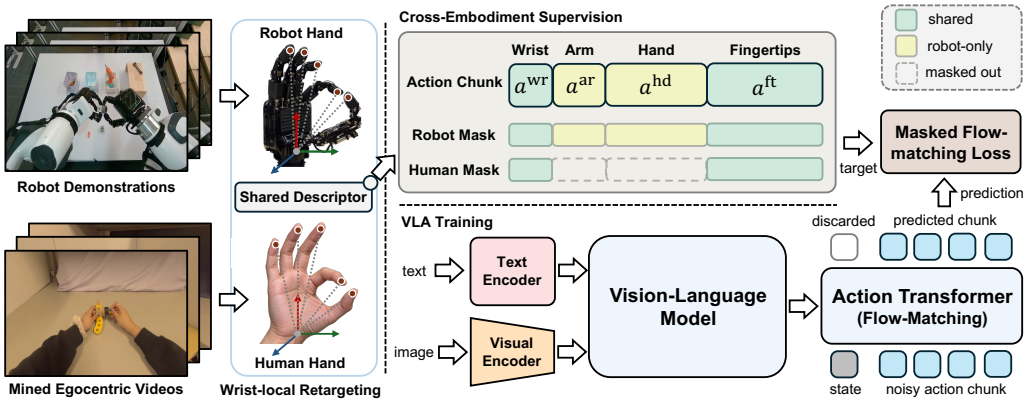

Figure 2: **Cross-embodiment training with mined human data.** Robot demonstrations (top left) and mined egocentric videos (bottom left) are retargeted onto a shared descriptor and laid out in one unified action chunk. Green marks the embodiment-shared segments, yellow the robot-only ones, and dashed cells are excluded by the mask. The **upper lane** carries the supervision target built from the retargeted descriptor. The **lower lane** carries the model prediction, and the two meet at the masked flow-matching loss, which is computed only on the unmasked entries of the chunk.

Stage III: Re-ranking. The re-ranking stage provides an embodiment-agnostic verification using optical flow. Unlike the preceding stages, which operate in the shared action space and therefore inherit any retargeting error, optical flow captures pixel-level motion patterns independent of hand morphology. For each anchor–target pair, we aggregate dense flow into a clip-level descriptor that summarises motion direction and energy, then re-score by descriptor similarity, yielding a signal that does not depend on retargeting accuracy. Because dense flow requires decoding and comparing video rather than cached action features, it is far costlier per pair than the action-space comparisons above, and we use it to analyse retrieval quality (Sec. 4.5) rather than to filter the training set.

3.3 Cross-Embodiment Training

Mined human data enters training as an additional data source alongside the robot data, with no other change to the model or training procedure, so that data selection remains the only variable (Fig. 2).

Data streams. The mined subset $\mathcal{D}_h^*$ and the robot demonstrations $\mathcal{D}_r$ are loaded as two independently weighted data streams and sampled at a 1:1 ratio (App. D.2), rather than concatenated into one dataset. Both are retargeted into the shared descriptor of Sec. 3.1, so a single policy consumes them through the same inputs and outputs, with no source-specific module or embodiment flag.

Cross-embodiment supervision. All cross-embodiment difference is confined to the supervision side, where it takes the form of a mask. Robot samples are supervised on all four segments of the action chunk, 88 dimensions. Human samples are supervised only on the embodiment-shared segments $\boldsymbol{a}^{\mathrm{wr}}$ and $\boldsymbol{a}^{\mathrm{ft}}$; the robot-specific segments $\boldsymbol{a}^{\mathrm{ar}}$ and $\boldsymbol{a}^{\mathrm{hd}}$ are filled with zero placeholders and excluded by a mask $\boldsymbol{m} \in \{0,1\}^{H\times 88}$. The same mask also removes future steps beyond the end of an episode, for both sources. The mask is applied per sample rather than per batch, so a single batch can mix both sources freely. Because the difference lives in the mask rather than in the network, human and robot data train a single model without any architectural change.

Architecture. The policy below the mask is an off-the-shelf π_0-like flow-matching VLA: a vision-language backbone encodes a single visual observation $\boldsymbol{I}_t$ and the language instruction l, a state adapter projects the proprioceptive state $\boldsymbol{s}_t \in \mathbb{R}^{86}$ into the backbone hidden space, and an action transformer denoises a noisy action chunk into the predicted chunk $\boldsymbol{a}_{t:t+H} \in \mathbb{R}^{H\times 88}$ over horizon H, laid out as in Sec. 3.1. The state pairs the robot's joint readings with the current fingertip configurations $\boldsymbol{p}_t$ of both hands; for human samples the joint slots are empty and zero-filled. This path is

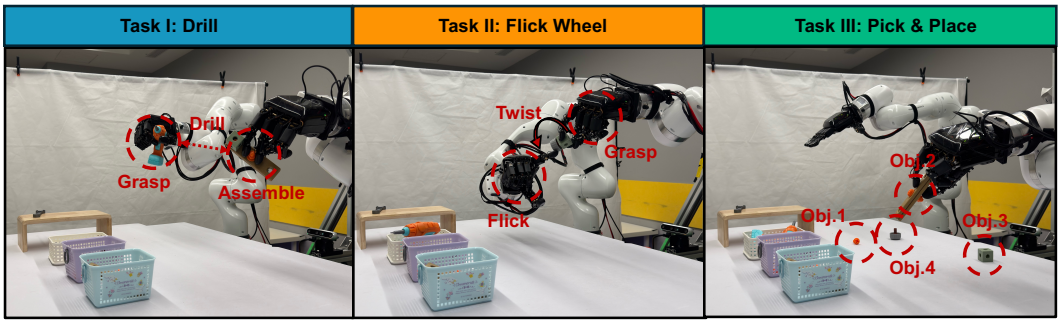

Figure 3: **Overview of SiMDex evaluation tasks.** We evaluate on three dexterous manipulation tasks: **Task I: Drill** (grasp, assemble, drill), **Task II: Flick Wheel** (grasp, twist, flick), and **Task III: Pick & Place** (grasp and place 4 objects of diverse shapes). Red marks show key stages and objects.

identical for robot and human samples: the same encoders, the same backbone, and the same input and output dimensions. SiMDex leaves the backbone unchanged; App. D.1 details ours.

Training objective. Supervision and prediction meet at the loss. We adopt a flow-matching objective. Given a ground-truth action chunk $\boldsymbol{a} \in \mathbb{R}^{H\times 88}$, we sample noise $\boldsymbol{\epsilon}$ with entries drawn i.i.d. from $\mathcal{N}(0,1)$ and a timestep $\sigma \in [0,1]$, form an interpolated state $\boldsymbol{x}_\sigma \in \mathbb{R}^{H\times 88}$ with flow target $\boldsymbol{u}_\sigma \in \mathbb{R}^{H\times 88}$, and regress the predicted velocity $\hat{\boldsymbol{u}}_\sigma = f_\theta(\boldsymbol{x}_\sigma, \boldsymbol{I}_t, l, \boldsymbol{s}_t)$. The training loss is a masked mean squared error:

$$\mathcal{L} = \frac{\sum_{h,d} m_{h,d} \left([\hat{\boldsymbol{u}}_\sigma]_{h,d} - [\boldsymbol{u}_\sigma]_{h,d}\right)^2}{\sum_{h,d} m_{h,d}} \tag{1}$$

where $h \in \{1,\dots,H\}$ indexes the prediction horizon, $d \in \{1,\dots,88\}$ indexes the action dimension, $[\hat{\boldsymbol{u}}_\sigma]_{h,d}$ and $[\boldsymbol{u}_\sigma]_{h,d}$ are the predicted and target velocities at entry (h,d), and $m_{h,d} \in \{0,1\}$ is the corresponding mask entry. The mask $\boldsymbol{m}$ ensures human samples contribute gradients only through the 42 wrist and fingertip dimensions, enabling training without architectural modification.

4 Experiments

We design experiments to answer the following research questions:

RQ1: Does SiMDex outperform a baseline trained on an equal amount of random human data?

RQ2: How does the benefit of SiMDex scale with the amount of available robot demonstrations?
RQ3: What changes in the retrieved samples at each stage of the SiMDex cascade?

4.1 Evaluation Tasks

As shown in Fig. 3, we design three dexterous manipulation tasks covering bimanual tool use, fine-grained finger dexterity, and precise grasping of multiple objects with diverse shapes.

Task I: Drill (max score 3). The robot sequentially grasps the drill, aligns and assembles it with a fixture, and presses the trigger—evaluating tool use and multi-step coordination.

Task II: Flick Wheel (max score 3). The robot grasps the assembly, twists the wheel with two fingers, and flicks it off with one finger. This is the most demanding in fine-grained finger dexterity, as the two-finger twist and one-finger flick stages require precise fingertip coordination.

Task III: Pick & Place (max score 4). Four objects of diverse geometries are scattered on the table; the robot picks and places each at a designated location, testing precise grasps across object shapes.

Stages are sequential except in Pick & Place, whose objects are scored independently. Sub-tasks earn partial credit per completed step, and success rate is total score over the maximum; each task is evaluated over 10 trials (2 rounds $\times$ 5 layouts) by an independent team blind to the method (App. E).

Table 1: **Real-world evaluation on dexterous manipulation tasks (RQ1).** Sub-task scores (0–1) are means over 10 trials (2 rounds $\times$ 5); $\pm$ is the standard deviation over the two round-level means ($n = 2$; App. E.5). Total = sum of sub-task scores; Success Rate (SR) = Total / max $\times 100\%$. Best in **bold**. Δ = SiMDex $-$ GR-Dexter (Wen et al., 2025a), the identical model trained with an equal amount of randomly sampled human data. Totals, rates, and Δ are computed from unrounded scores, so Δ may differ by at most one unit in the last digit from the difference of the rounded entries.

Task	Method	Sub-task Scores			Total	Success Rate(%)↑
		Grasp	Assem.	Drill		
Drill (max 3)	GR-Dexter	**0.80**$\pm$.00	**0.80**$\pm$.00	**0.33**$\pm$.00	**1.93**$\pm$.00	**64.5**
	SiMDex (Ours)	0.73$\pm$.10	0.63$\pm$.33	0.27$\pm$.28	1.63$\pm$.71	54.5
	Δ	-0.07	-0.17	-0.07	-0.30	-10.0
		Grasp	Twist	Flick		
Flick Wheel (max 3)	GR-Dexter	0.60$\pm$.14	0.13$\pm$.19	0.00$\pm$.00	0.73$\pm$.33	24.5
	SiMDex (Ours)	**0.85**$\pm$.07	**0.47**$\pm$.09	**0.10**$\pm$.00	**1.42**$\pm$.17	**47.2**
	Δ	$+0.25$	$+0.33$	$+0.10$	$+0.68$	$+22.8$
		Obj.1	Obj.2	Obj.3	Obj.4	
Pick & Place (max 4)	GR-Dexter	0.30$\pm$.05	0.67$\pm$.09	0.53$\pm$.09	0.67$\pm$.09	2.16$\pm$.05	54.0
	SiMDex (Ours)	**0.67**$\pm$.19	**0.87**$\pm$.00	**0.97**$\pm$.05	**0.83**$\pm$.05	**3.33**$\pm$.19	**83.3**
	Δ	$+0.37$	$+0.20$	$+0.44$	$+0.17$	$+1.17$	$+29.4$
Overall Success Rate(%)						47.7 $\rightarrow$ **61.7** $+14.0$	

4.2 Implementation Details

Base model. Our base model follows GR-Dexter (Wen et al., 2025a), a π_0-like flow-matching VLA for dexterous manipulation. Its pre-training data include EgoDex, the egocentric corpus that SiMDex re-mines during post-training. SiMDex modifies only the human-data source, leaving the architecture and training procedure unchanged; the baseline (GR-Dexter) uses the identical model and an equal amount of randomly sampled human data, so only the selection differs.

Data. Robot demonstrations are collected by bimanual teleoperation of a two-arm, two-hand platform. From the raw collection we curate a high-quality subset of $\sim$1.35M frame-level samples ($\sim$12.4 hours; App. B.4), which serves as the default robot-data budget ($1\times$) throughout the paper; all results in Table 1 use this setting. The human pool is derived from EgoDex (Hoque et al., 2026), an egocentric manipulation dataset of 829 hours across 194 tabletop tasks. After the preprocessing of Sec. 3.1, it yields 32,034,551 frame-level samples ($\sim$32M; App. B.3) over 164,959 episodes, sliced into sliding-window samples (each a $\sim$1 s, 30-step window). From this pool, SiMDex retrieves $\sim$1.49M samples (under 5% of the pool); the baseline randomly samples an equal amount.

Training. For the main results we train with a 1:1 mixture of robot and human samples for 40K steps, roughly two passes over each source. All hyperparameters are identical between SiMDex and the baseline, isolating data selection rather than model capacity or training budget (App. D.4).

4.3 Main Results

Table 1 reports real-world results under our standard setting. Mining task-relevant human data lifts the overall success rate from 47.7% to 61.7% (+14.0) with only $\sim$1.49M retrieved samples (under 5% of the pool), showing that targeted retrieval outperforms random sampling of equal size.

Per-task. SiMDex improves two of three complex tasks by large margins. On *Flick Wheel*, it nearly doubles the success rate (24.5 $\rightarrow$ 47.2, +22.8); notably the baseline almost entirely fails the Twist (0.13) and Flick (0.00) stages, indicating that mined human data conveys fine-grained finger skills that random data does not (App. F.2). On *Pick & Place*, it improves 54.0 $\rightarrow$ 83.3 (+29.4) with consistent gains on all four objects. On *Drill*, SiMDex is lower than the baseline (54.5 vs. 64.5). The gap arises at Grasp and Assembly, not at drilling (App. F.2), and varies widely between rounds ($\pm$.71 on the SiMDex total), so we do not read it as a systematic effect of mining.

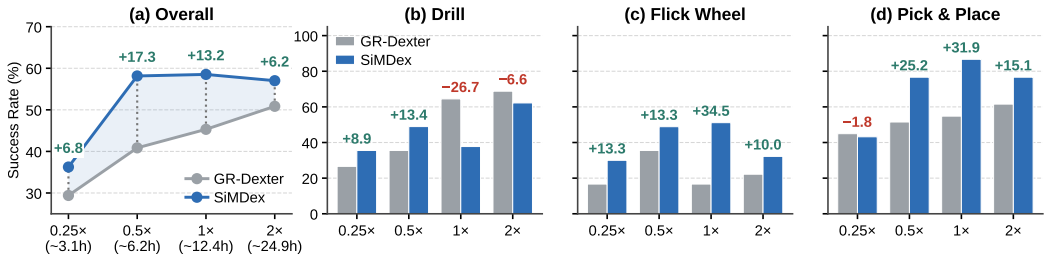

Figure 4: **Data scaling ablation (RQ2). (a)** Overall SR vs. robot-data scale: SiMDex stays at ∼57–59% from $0.5\times$ to $2\times$ while the baseline drops with less data (largest gain $+17.3$ at $0.5\times$). **(b–d)** Per-task SR (Δ above bars). All results use one round of five trials per task; the $1\times$ column is the first of the two rounds in Table 1. Full per-sub-task numbers are in App. F.1.

4.4 Data Scaling Ablation

Since robot data is costly, we vary the robot-data budget from $0.25\times$ (∼3.1h) to $2\times$ (∼24.9h) while keeping the mined human data fixed (Fig. 4). This subset is mined once with the $1\times$ anchors and reused at every budget, and the $2\times$ budget adds lower-quality early teleoperation data (App. D.5).

Gains across budgets. SiMDex outperforms the baseline at every scale (Fig. 4a) and stays at ∼57–59% from $0.5\times$ to $2\times$, while the baseline falls from 50.9% to 40.9%. The largest gain ($+17.3$) appears at $0.5\times$, and at $0.25\times$ SiMDex still improves by $+6.8$. Since the subset is reused, mining at smaller budgets also uses anchors they do not train on; re-mining per budget is future work.

Per-task behavior. Per task (Fig. 4b–d), Flick Wheel gains at every budget and Pick & Place at every budget except $0.25\times$, while the sign of the Drill gap varies. With a single round of five trials per point, we read per-task differences at a single budget as indicative only.

4.5 Visualization

For RQ3, Fig. 5 shows, for three anchors, the top retrieved sample after each stage; in these examples, the match to the anchor tightens step by step from Recall through Ranking to Re-ranking.

Recall matches coarse cues only: it retrieves samples with a similar initial pose or language-relevant objects (a "wheel", a "basket"), but the actions are often unrelated to the target—simple planar grasping rather than tool use or twisting. **Ranking** aligns fine-grained motion: pinching and twisting patterns, finger curvature, and rotation direction come closer to the anchor. Yet samples with distant hand positions can still rank high when their kinematic values happen to match. **Re-ranking** reorders these candidates by optical flow; its top samples, already in the training subset, match the anchor's motion: a left hand stabilizing while the right aligns a tool (Drill), a coordinated stabilize-and-twist (Flick Wheel), and sequential precision grasping of small items (Pick & Place). In short, recall brings in clips with related objects or initial poses, ranking aligns their motion with the anchor, and re-ranking checks that motion on pixels. More examples, including failure cases, are in App. C.

5 Conclusion

We presented SiMDex, a similarity-based data mining framework that retrieves task-relevant human demonstrations from large-scale egocentric video to improve VLA training for dexterous manipulation. Cast as a retrieval problem, SiMDex uses recall and ranking to mine a small, high-quality subset from a pool of tens of millions of human samples and re-ranking to verify it, modifying only the data source while leaving the VLA architecture and training unchanged. Across three real-world tasks, mining lifts the overall success rate from 47.7% to 61.7% using under 5% of the pool, and the gain holds at every robot-data budget from $0.25\times$ to $2\times$, pointing to selective human-data mining as a practical complement to the costly collection of more robot demonstrations.

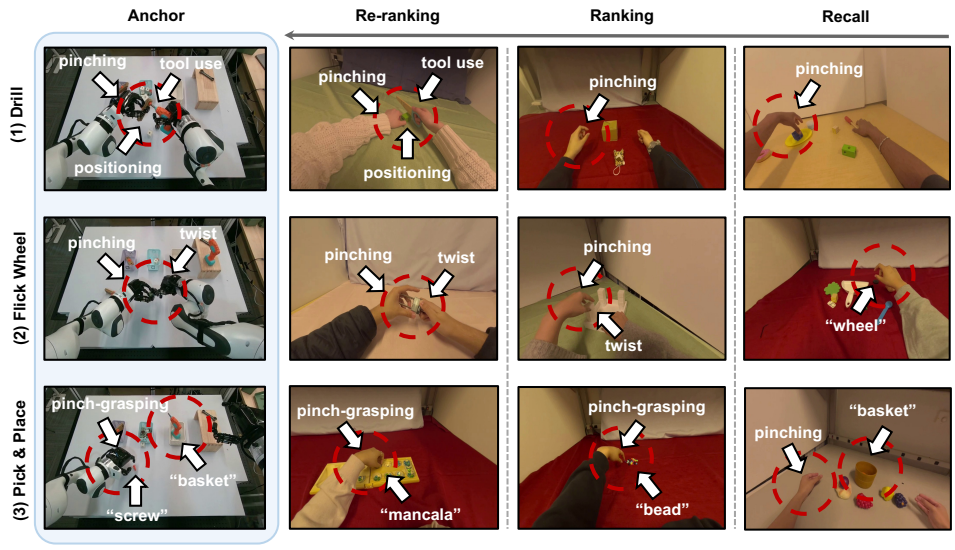

Figure 5: **Qualitative visualization of the SiMDex three-stage retrieval pipeline (RQ3).** Each row shows a robot anchor (left) and the top retrieved human sample after each stage, from Recall (right) to Re-ranking. Plain labels name hand actions, quoted labels name objects, and red circles mark interaction regions. Re-ranking then adds a quality check on pixels via optical flow, flagging samples that match the anchor in the action space but move differently in the video.

6 Limitations

Scope. Our robot data covers one scenario (industrial assembly), three tasks, and ∼12.4 hours of demonstrations; generality across scenes, tasks, and embodiments remains untested.

What the comparison isolates. Our control holds the human-data budget fixed and varies only how that data is selected, since the value of human data at scale is documented in prior work (Zheng et al., 2026). Our claims therefore concern selection—targeted retrieval beats random sampling at equal budget—not the value of human data in general, which would require a robot-only control.

Dependence on pool coverage. SiMDex depends on pool coverage: the drilling stage of Drill has no counterpart among the retrieved clips (App. C.3), and mining brings no gain there.

Kinematic-only similarity. Our similarity is purely kinematic (wrist 6D pose and fingertip positions): it captures motion but not contact forces, object state, or interaction semantics; the ranking stage can therefore mismatch—e.g., pairing the robot's nearly static drilling with oscillating tool-use clips—an error that re-ranking can flag on pixels via optical flow (App. C.3).

7 Future Work

These limitations point to a natural next step: moving from one-shot, offline retrieval toward closed-loop, adaptive mining. In the current pipeline, *what* to retrieve is fixed before training, yet our scaling analysis suggests that the most useful human data may depend on the robot-data budget and on which skills the policy has yet to master. The policy's failure modes could steer subsequent retrieval, the similarity signal could capture contact and object state, and re-ranking could move from checking the mined subset to filtering it once its flow descriptors are cached per clip. Closing this loop would let egocentric collection be exploited not once, but continually throughout policy learning. Ultimately we see SiMDex as a step toward a recommender for robot skills, where learning a new skill is not a fresh data-collection burden but a query against experience that already exists.

AI USE STATEMENT

In this work, we used generative AI tools to assist with translation, to receive feedback on our research methodology, experiments, and their presentation, to help phrase the interpretation of our experimental results, and to transcribe the evaluation score sheets and check them against the scoring rubric. We have not used generative AI tools to generate synthetic datasets, to develop theoretical models or conceptual frameworks, to formulate mathematical claims, to provide critical ingredients for proving mathematical claims, to assist in the writing of proofs, to propose or refine hypotheses, or to implement methods; qualitative and thematic data analysis is not applicable to this work.

Additionally, we used generative AI tools to draft and edit parts of the manuscript for readability, to suggest a paper structure, and to find and format references. We have reviewed all AI-assisted work. Every bibliographic entry was manually verified by the authors against the original publication—checking title, author list, venue, and year—and every reported number, table, and figure against the original experimental records. All interpretations of the experimental results were verified by the authors and are based on evaluations conducted on real robot hardware; no reported result was generated, simulated, or extrapolated by a generative AI tool. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve new human-subject data collection. The human egocentric data used in this paper comes entirely from EgoDex (Hoque et al., 2026), a publicly released dataset, and is used in accordance with its license and intended research use; consent and collection procedures were handled by the original dataset authors. Our pipeline uses this data only for retrieval, and we release no imagery or personally identifiable information beyond the public source dataset.

Robot demonstrations were collected by trained operators through teleoperation in a controlled laboratory setting. During autonomous evaluation, no person entered the robot workspace; trials were observed and scored from outside it by trained personnel who could halt the robot at any time. The authors declare no conflicts of interest and took no part in running or scoring the evaluation trials.

REPRODUCIBILITY STATEMENT

Sec. 3.1 gives the wrist-local retargeting and the unified action layout, and lists the preprocessing steps applied to the human data before retargeting; Sec. 3.2 gives the three cascade stages with their descriptors, similarity measures, and rank-fusion rules; and Sec. 3.3 gives the cross-embodiment mask, the data-stream mixing, and the masked flow-matching objective (Eq. 1).

Sec. 4.1 defines the evaluation tasks and App. E gives the full trial protocol and scoring rubric; the captions of Table 1 and Fig. 4, together with App. E.5, state how the results are aggregated. Sec. 4.2 reports the base model, the human pool and mined-subset sizes, the default robot-data budget, the sampling ratio, and the training schedule; the robot-data budgets used in the scaling ablation are given in Sec. 4.4 and App. D.5. The appendix gives the remaining details: per-stage formulas and retrieval hyperparameters (App. A), the preprocessing and statistics of the human pool (App. B), the base model and training hyperparameters (App. D), and all per-sub-task scaling results (App. F).

EgoDex (Hoque et al., 2026) is publicly available, so the retrieval stages can be reproduced on the same pool; for the real-robot experiments, App. E specifies the tasks, language instructions, scoring rubric, and trial protocol in enough detail to repeat them on comparable hardware.

REFERENCES

Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S. Morcos. SemDeDup: Data-efficient learning at web-scale through semantic deduplication. *arXiv preprint arXiv:2303.09540*, 2023.

Christopher Agia, Rohan Sinha, Jingyun Yang, Rika Antonova, Marco Pavone, Haruki Nishimura, Masha Itkina, and Jeannette Bohg. CUPID: Curating data your robot loves with influence functions. In *Conference on Robot Learning (CoRL)*, pp. 2907–2932, 2025.

Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, et al. $\pi^*_{0.6}$: A VLA that learns from experience. In *Proceedings of Robotics: Science and Systems (RSS)*, 2026.

Shikhar Bahl, Abhinav Gupta, and Deepak Pathak. Human-to-robot imitation in the wild. In *Proceedings of Robotics: Science and Systems (RSS)*, 2022.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. In *Conference on Robot Learning (CoRL)*, pp. 17–40, 2025a.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, et al. π_0: A vision-language-action flow model for general robot control. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025b.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023a.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, et al. RT-1: Robotics transformer for real-world control at scale. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023b.

Chilam Cheang, Sijin Chen, Zhongren Cui, Yingdong Hu, Liqun Huang, et al. GR-3 technical report. *arXiv preprint arXiv:2507.15493*, 2025.

Sijin Chen, Kaixuan Jiang, Haixin Shi, Yanhui Wang, Weiheng Zhong, et al. Translation as a Bridging Action: Transferring manipulation skills from humans to robots. *arXiv preprint arXiv:2606.28133*, 2026.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion Policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, pp. 1684–1704, 2024.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision*, 130(1):33–55, 2022.

Yufan Deng and Daquan Zhou. HumanNet: Scaling human-centric video learning to one million hours. *arXiv preprint arXiv:2605.06747*, 2026.

Maximilian Du, Suraj Nair, Dorsa Sadigh, and Chelsea Finn. Behavior retrieval: Few-shot imitation learning by querying unlabeled datasets. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, et al. DataComp: In search of the next generation of multimodal datasets. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 27092–27112, 2023.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, et al. Ego4D: Around the world in 3,000 hours of egocentric video. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18995–19012, 2022.

Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, et al. Ego-Exo4D: Understanding skilled human activity from first- and third-person perspectives. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19383–19400, 2024.

Ankur Handa, Karl Van Wyk, Wei Yang, Jacky Liang, Yu-Wei Chao, et al. DexPilot: Vision-based teleoperation of dexterous robotic hand-arm system. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9164–9170, 2020.

Joey Hejna, Chethan Anand Bhateja, Yichen Jiang, Karl Pertsch, and Dorsa Sadigh. ReMix: Optimizing data mixtures for large scale imitation learning. In *Conference on Robot Learning (CoRL)*, pp. 145–164, 2024.

Ryan Hoque, Peide Huang, David J. Yoon, Mouli Sivapurapu, and Jian Zhang. EgoDex: Learning dexterous manipulation from large-scale egocentric video. In *International Conference on Learning Representations (ICLR)*, 2026.

Simar Kareer, Dhruv Patel, Ryan Punamiya, Pranay Mathur, Shuo Cheng, Chen Wang, Judy Hoffman, and Danfei Xu. EgoMimic: Scaling imitation learning via egocentric video. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13226–13233, 2025.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, et al. DROID: A large-scale in-the-wild robot manipulation dataset. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, et al. OpenVLA: An open-source vision-language-action model. In *Conference on Robot Learning (CoRL)*, pp. 2679–2713, 2024.

Marion Lepert, Jiaying Fang, and Jeannette Bohg. Masquerade: Learning from in-the-wild human videos using data-editing. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, et al. CogACT: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024.

Qixiu Li, Yu Deng, Yaobo Liang, Lin Luo, Lei Zhou, Chengtang Yao, Lingqi Zeng, Zhiyuan Feng, Huizhi Liang, Sicheng Xu, et al. Scalable vision-language-action model pretraining for robotic manipulation with real-life human activity videos. *arXiv preprint arXiv:2510.21571*, 2025.

Yihang Li, Xuelong Wei, Jingzhou Luo, Yingjing Xiao, Yibo Bai, Guangyuan Zhou, Teng Zou, Chenguang Gui, Jiajun Wen, He Zhang, et al. EgoLive: A large-scale egocentric dataset from real-world human tasks. *arXiv preprint arXiv:2604.23570*, 2026.

Nie Lin, Takehiko Ohkawa, Yifei Huang, Mingfang Zhang, Minjie Cai, Ming Li, Ryosuke Furuta, and Yoichi Sato. SiMHand: Mining similar hands for large-scale 3D hand pose pre-training. In *International Conference on Learning Representations (ICLR)*, 2025.

Hao Luo, Yicheng Feng, Wanpeng Zhang, Sipeng Zheng, Ye Wang, et al. Being-H0: Vision-language-action pretraining from large-scale human videos. *arXiv preprint arXiv:2507.15597*, 2025.

Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. VIP: Towards universal visual reward and representation via value-implicit pre-training. In *International Conference on Learning Representations (ICLR)*, 2023.

Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Yecheng Jason Ma, Claire Chen, et al. Where are we in the search for an artificial visual cortex for embodied intelligence? In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 655–677, 2023.

Juncheng Mu, Sizhe Yang, Yiming Bao, Hojin Bae, Tianming Wei, Linning Xu, Boyi Li, Huazhe Xu, and Jiangmiao Pang. DexImit: Learning bimanual dexterous manipulation from monocular human videos. In *Proceedings of Robotics: Science and Systems (RSS)*, 2026.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3M: A universal visual representation for robot manipulation. In *Conference on Robot Learning (CoRL)*, pp. 892–909, 2022.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, et al. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

Takehiko Ohkawa, Jumpei Arima, Yuki Noguchi, Masatoshi Tateno, Makoto Sugiura, et al. YUBI: Yielding universal bidigital interface for bimanual dexterous manipulation at scale. *arXiv preprint arXiv:2606.10244*, 2026.

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

Georgios Papagiannis, Norman Di Palo, Pietro Vitiello, and Edward Johns. R+X: Retrieval and execution from everyday human videos. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8284–8290, 2025.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, et al. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 30811–30849, 2024.

Ryan Punamiya, Dhruv Patel, Patcharapong Aphiwetsa, Pranav Kuppili, Lawrence Zhu, Simar Kareer, Judy Hoffman, and Danfei Xu. EgoBridge: Domain adaptation for generalizable imitation from egocentric human data. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 47092–47122, 2025.

Ryan Punamiya, Simar Kareer, Zeyi Liu, Joshua Citron, Ri-Zhao Qiu, Xiongyi Cai, Alexey Gavryushin, Jiaqi Chen, Davide Liconti, Lawrence Y. Zhu, et al. EgoVerse: An egocentric human dataset for robot learning from around the world. In *Proceedings of Robotics: Science and Systems (RSS)*, 2026.

Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training. In *Conference on Robot Learning (CoRL)*, pp. 416–426, 2022.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.

Kushal Tirumala, Daniel Simig, Armen Aghajanyan, and Ari Morcos. D4: Improving LLM pretraining via document de-duplication and diversification. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 53983–53995, 2023.

Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. MimicPlay: Long-horizon imitation learning by watching human play. In *Conference on Robot Learning (CoRL)*, pp. 201–221, 2023.

Chen Wang, Haochen Shi, Weizhuo Wang, Ruohan Zhang, Li Fei-Fei, and Karen Liu. DexCap: Scalable and portable mocap data collection system for dexterous manipulation. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 5776–5788, 2020.

Ruoshi Wen, Guangzeng Chen, Zhongren Cui, Min Du, Yang Gou, et al. GR-Dexter technical report. *arXiv preprint arXiv:2512.24210*, 2025a.

Ruoshi Wen, Jiajun Zhang, Guangzeng Chen, Zhongren Cui, Min Du, et al. Dexterous teleoperation of 20-DoF ByteDexter hand via human motion retargeting. *arXiv preprint arXiv:2507.03227*, 2025b.

Ruihan Yang, Qinxi Yu, Yecheng Wu, Rui Yan, Borui Li, An-Chieh Cheng, Xueyan Zou, Yunhao Fang, Xuxin Cheng, Ri-Zhao Qiu, et al. EgoVLA: Learning vision-language-action models from egocentric human videos. *arXiv preprint arXiv:2507.12440*, 2025.

Haoqi Yuan, Yu Bai, Yuhui Fu, Bohan Zhou, Yicheng Feng, et al. Being-0: A humanoid robotic agent with vision-language models and modular skills. *arXiv preprint arXiv:2503.12533*, 2025.

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.

Yifan Zhong, Xuchuan Huang, Ruochong Li, Ceyao Zhang, Zhang Chen, et al. DexGraspVLA: A vision-language-action framework towards general dexterous grasping. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 18836–18844, 2026.

Yichen Zhu and Feifei Feng. Let me show you: Learning by retrieving from egocentric video for robotic manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9824–9829, 2025.

APPENDIX



Figure 6: **Robot anchors and the human samples mined by SiMDex.** Each robot tile is paired with the adjacent human tile mined for it from EgoDex (Hoque et al., 2026).

Each pair in Figure 6 reflects three ideas behind SiMDex:

- **Relevance lies in motion, not labels.** The two tiles of a pair differ in embodiment, scene, and activity, yet share the motion itself: a pinch, a wrist twist, or a hand-over.
- **Labels can mislead.** Grasps mined for a sub-task come from recordings of other activities, so a pool that looks unrelated by its labels can still hold the motion the robot needs (App. C).
- **Retrieval must fit a budget.** Like a recommender, SiMDex surfaces useful samples from tens of millions under a strict compute budget, so it uses a cascade of cheap signals and keeps the costly pixel-level signal (optical flow) for verification only (Stage III, Sec. 3.2).

The rest of the appendix covers method details (App. A), data and the human pool (App. B), what gets mined (App. C), experimental setup (App. D), evaluation protocol (App. E), and additional results (App. F), with per-sub-task scaling scores, conditional pass rates, and qualitative rollouts.

A METHOD DETAILS

This appendix expands the method of Sec. 3 with the full set of definitions, hyperparameters, and per-stage formulas that were abbreviated in the main paper. We keep the notation of the main text throughout, introducing a symbol only where the appendix needs to name something that the main text refers to in words. Nothing below modifies the method of Sec. 3: every added quantity is a concrete realization of a component already described there.

A.1 UNIFIED MORPHOLOGY-AGNOSTIC REPRESENTATION

From single-hand to bimanual action. The main text defines, for a single hand, the wrist-local fingertip configuration $\boldsymbol{p}_t \in \mathbb{R}^{15}$, the wrist action $\boldsymbol{d}_t \in \mathbb{R}^6$ for the k-th future step, and the fingertip target $\boldsymbol{f}_t := \boldsymbol{p}_{t+k} \in \mathbb{R}^{15}$ predicted at that same future step. We make the single-hand $\rightarrow$ bimanual composition explicit here and give an explicit symbol to the 42 embodiment-shared dimensions, which the main text refers to only by their count. Writing the left and right hands with superscripts L and R, the two per-hand quantities are

$$\boldsymbol{d}_t^{\mathrm{L}},\ \boldsymbol{d}_t^{\mathrm{R}} \in \mathbb{R}^6, \qquad \boldsymbol{f}_t^{\mathrm{L}},\ \boldsymbol{f}_t^{\mathrm{R}} \in \mathbb{R}^{15}, \tag{2}$$

and the two embodiment-shared segments of the main text, $\boldsymbol{a}_t^{\mathrm{wr}} \in \mathbb{R}^{12}$ and $\boldsymbol{a}_t^{\mathrm{ft}} \in \mathbb{R}^{30}$, are exactly their per-hand concatenations. We write the two segments together, in the order in which they appear in $\boldsymbol{a}_t$, as the shared action

$$\boldsymbol{a}_t^{\mathrm{sh}} = \left(\boldsymbol{a}_t^{\mathrm{wr}},\ \boldsymbol{a}_t^{\mathrm{ft}}\right) \in \mathbb{R}^{42}. \tag{3}$$

Its 42 dimensions decompose as $2 \times 6 = 12$ for the two wrists and $2 \times 15 = 30$ for the ten fingertips; they are the 42 wrist and fingertip dimensions through which human samples contribute gradients in Sec. 3.3. The symbol $\boldsymbol{a}_t^{\mathrm{sh}}$ is a notational convenience only: it does not change the action layout of the main text, but names those components of $\boldsymbol{a}_t$ that are embodiment-shared, because the action-space comparison of mining operates on them alone. The two hands are handled symmetrically and combined as described in App. A.5.

Anchor and target sets. Following the main text, let $\mathcal{D}_r = \{\tau_n\}_{n=1}^{N}$ be the robot samples (anchors) and $\mathcal{D}_h = \{\tau_m\}_{m=1}^{M}$ the human target pool, with $M \gg N$. Each sample τ pairs a language instruction l with an observation and a future action chunk; mining uses only l, the observed fingertip configuration $\boldsymbol{p}_t$, and the shared part $\boldsymbol{a}_{t:t+H}^{\mathrm{sh}} \in \mathbb{R}^{H \times 42}$ of that chunk. Anchors and targets are mapped onto the identical representation of Sec. 3.1 and all features are extracted by the *same* operators, so that similarities are computed in one common space; this symmetry is what makes cross-embodiment comparison well defined and is assumed throughout. Throughout, we write $s(\boldsymbol{z}, \boldsymbol{z}')$ for the cosine similarity between two real vectors $\boldsymbol{z}$ and $\boldsymbol{z}'$ of the same length, so that $s(\boldsymbol{z}, \boldsymbol{z}') \in [-1, 1]$.

A.2 STAGE I: RECALL

Recall combines two lightweight per-anchor rankings, one from initial hand pose and one from language, into a single candidate set. Fix an anchor $\tau \in \mathcal{D}_r$; all ranks below are taken over targets $\tau' \in \mathcal{D}_h$ with respect to this anchor, and every anchor is processed independently.

Pose ranking. As in the main text, for a single hand we ℓ_2-normalize the observed fingertip configuration $\boldsymbol{p}_t$, removing hand scale, to obtain the pose descriptor

$$\hat{\boldsymbol{p}} = \frac{\boldsymbol{p}_t}{\|\boldsymbol{p}_t\|_2} \in \mathbb{R}^{15}. \tag{4}$$

Targets are retrieved by Euclidean distance, summed over the two hands,

$$\delta_{\mathrm{pose}} = \left\|\hat{\boldsymbol{p}}^{\mathrm{L}} - \hat{\boldsymbol{p}}'^{\mathrm{L}}\right\|_2 + \left\|\hat{\boldsymbol{p}}^{\mathrm{R}} - \hat{\boldsymbol{p}}'^{\mathrm{R}}\right\|_2, \tag{5}$$

where $\hat{\boldsymbol{p}}$ and $\hat{\boldsymbol{p}}'$ are the anchor and target pose descriptors. For a single hand, both descriptors are unit vectors, so $\|\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}'\|_2^2 = 2 - 2\, s(\hat{\boldsymbol{p}}, \hat{\boldsymbol{p}}')$ and ordering by Euclidean distance is equivalent to ordering by cosine similarity; we use the Euclidean form to match the main text. Ordering targets by increasing δ_{pose} yields a pose rank $r_{\mathrm{pose}}(\tau') \in \{1, \dots, M\}$ for each target τ' (smaller is more similar). The low-dimensional descriptor keeps this ranking cheap even over the full pool.

Language ranking. Each instruction l is encoded into a sentence embedding $\boldsymbol{e} \in \mathbb{R}^{384}$ by a pre-trained sentence encoder.[1] Ordering targets by decreasing $s(\boldsymbol{e}, \boldsymbol{e}')$, where $\boldsymbol{e}'$ is the target embedding, yields a language rank $r_{\mathrm{lang}}(\tau') \in \{1, \dots, M\}$ (smaller is more similar).

Rank fusion. The two rankings are merged by reciprocal-rank fusion, giving the fused recall score

$$s_{\mathrm{rrf}}(\tau') = \frac{1}{2}\,\frac{1}{\kappa + r_{\mathrm{pose}}(\tau')} + \frac{1}{2}\,\frac{1}{\kappa + r_{\mathrm{lang}}(\tau')}, \qquad \kappa = 60, \tag{6}$$

where $\kappa \in \mathbb{N}$ is the rank-fusion constant that controls how sharply the contribution of a ranking decays with rank: a larger κ flattens the difference between top ranks, so that a target favored by *both* the pose and language rankings is rewarded over one ranked first by a single signal alone. We set $\kappa = 60$, the standard default for reciprocal-rank fusion. We fuse ranks rather than scores because the two signals are not commensurable: pose is a Euclidean distance in $\mathbb{R}^{15}$ and language a cosine similarity in $\mathbb{R}^{384}$, so their scores cannot be added directly. The reciprocal transform further bounds the contribution of each ranking by $\frac{1}{2(\kappa+1)}$, so that neither signal can dominate the fused ordering through the long tail of its own rank distribution.

Candidate cutoff. The fused list is narrowed to the top $K_{\mathrm{I}} = 10{,}000$ targets. The cutoff is set by compute: it keeps the costlier framewise matching of Stage II, which compares each candidate over the whole chunk, affordable. Writing $r_{\mathrm{recall}}(\tau')$ for the rank of τ' under decreasing s_{rrf}, the retained candidate set is

$$\mathcal{C}_{\mathrm{I}}(\tau) = \big\{\, \tau' \in \mathcal{D}_h \;\big|\; r_{\mathrm{recall}}(\tau') \le K_{\mathrm{I}} \,\big\}, \qquad \big|\mathcal{C}_{\mathrm{I}}(\tau)\big| = K_{\mathrm{I}}. \tag{7}$$

A.3 Stage II: Ranking

Ranking refines $\mathcal{C}_{\mathrm{I}}(\tau)$ by fine-grained motion similarity along the four components named in the main text. We give the concrete form of each component and of their fusion.

Motion components. The four components of the main text are two framewise trajectories, which preserve spatial structure, and two fixed-length waveforms, which summarize motion rhythm; all are computed per hand. The wrist trajectory $\boldsymbol{F}_{\mathrm{wr}} \in \mathbb{R}^{31\times 3}$ is the current wrist position followed by the 30 future ones; we index its rows from 0, so that row 0 is the current position, which is the origin in wrist-local coordinates. The fingertip trajectory $\boldsymbol{F}_{\mathrm{ft}} \in \mathbb{R}^{30\times 15}$ holds the fingertip targets of one hand over the $H = 30$ future steps, with the five fingertips stacked along the feature dimension ($5 \times 3 = 15$). The translation waveform $\boldsymbol{w}_{\mathrm{tr}} \in \mathbb{R}^{10}$ is the per-step translation speed resampled from 30 steps to 10 samples. Concretely, we prepend a zero frame to the future wrist translation sequence and take consecutive differences to obtain per-step increments. The norm of each increment gives a scalar speed sequence of length 30. We then normalize that sequence in magnitude, build a phase variable from its cumulative path length, and resample along that phase to the fixed length of 10. The rotation waveform $\boldsymbol{w}_{\mathrm{rot}} \in \mathbb{R}^{10}$ is obtained by applying the same operator to the per-step wrist rotation increments. Normalization and path-length resampling make a waveform describe how speed varies along the path, not how fast or how long the motion was.

Per-component similarity. The two waveforms are scored by cosine similarity on the 10-dimensional vectors,

[1] We use the public `all-MiniLM-L6-v2` encoder from Sentence-Transformers (Reimers & Gurevych, 2019), built on MiniLM (Wang et al., 2020); it produces 384-dimensional embeddings and is not fine-tuned.

$$s_{\mathrm{tr}} = s(\boldsymbol{w}_{\mathrm{tr}}, \boldsymbol{w}'_{\mathrm{tr}}), \qquad s_{\mathrm{rot}} = s(\boldsymbol{w}_{\mathrm{rot}}, \boldsymbol{w}'_{\mathrm{rot}}). \tag{8}$$

The two trajectories are scored by a framewise similarity averaged over the $H = 30$ future steps. Writing $\boldsymbol{F}_{i:}$ for the row of a trajectory matrix $\boldsymbol{F}$ at future step i, the similarity between an anchor trajectory $\boldsymbol{F}$ and a target trajectory $\boldsymbol{F}'$ is

$$s_{\mathrm{traj}}(\boldsymbol{F}, \boldsymbol{F}') = \frac{1}{H} \sum_{i=1}^{H} s(\boldsymbol{F}_{i:},\ \boldsymbol{F}'_{i:}), \tag{9}$$

applied to the wrist and the fingertip trajectory to give s_{wr} and s_{ft}. Row 0 of the wrist trajectory is the origin for every sample; it carries no information and its cosine similarity is undefined, so it is left out of the average. Including it under any fixed convention would rescale and shift every score in the same way and leave the ranking unchanged. As stated in the main text, every component is scored per hand and then summed over the two hands to give a single bimanual component score.

Rank-sum fusion and deduplication. Each of the four components induces a ranking of the candidates in $\mathcal{C}_{\mathrm{I}}(\tau)$, giving per-component ranks $r_{\mathrm{wr}}, r_{\mathrm{tr}}, r_{\mathrm{rot}}, r_{\mathrm{ft}} \in \{1, \ldots, K_{\mathrm{I}}\}$, and the fused score ρ of the main text sums them with equal weight,

$$\rho = r_{\mathrm{wr}} + r_{\mathrm{tr}} + r_{\mathrm{rot}} + r_{\mathrm{ft}}, \tag{10}$$

and targets are ordered by increasing ρ (smaller is better). We then deduplicate the sorted list *by source trajectory*: scanning from the best-ranked target, we keep the first candidate of each source episode and discard its later duplicates, so that no source contributes more than once. The first $K_{\mathrm{II}} = 100$ entries of the deduplicated list form the mined subset contributed by τ,

$$\mathcal{C}_{\mathrm{II}}(\tau) \subseteq \mathcal{C}_{\mathrm{I}}(\tau), \qquad \left|\mathcal{C}_{\mathrm{II}}(\tau)\right| \leq K_{\mathrm{II}}. \tag{11}$$

The set $\mathcal{C}_{\mathrm{II}}(\tau)$ has fewer than K_{II} entries only when $\mathcal{C}_{\mathrm{I}}(\tau)$ spans fewer than K_{II} source episodes. The global mined subset $\mathcal{D}_h^*$ of the main text is the union of these per-anchor lists; this global deduplication keeps each target only once, even when several anchors retrieve it,

$$\mathcal{D}_h^* = \bigcup_{\tau \in \mathcal{D}_r} \mathcal{C}_{\mathrm{II}}(\tau). \tag{12}$$

The mined subset used for training is thus the output of the full cascade: per-anchor recall, ranking, and source deduplication, followed by the global deduplication of Eq. (12).

A.4 Stage III: Re-ranking

Re-ranking provides an embodiment-agnostic check based on optical flow. Crucially, and consistent with the main text, this stage produces a *verification and analysis signal only*: it does not alter the mined subset $\mathcal{D}_h^*$ of Eq. (12) that is used for training. It is applied to the top K_{II} candidates of each anchor rather than to the full pool, since dense flow is costly to compute.

Flow descriptor and re-ranking score. For an anchor–target pair we extract dense optical flow over a short clip around the matched frame.[2] The flow is aggregated into a fixed-length, embodiment-agnostic clip-level descriptor $\boldsymbol{g} \in \mathbb{R}^D$. The descriptor summarizes where in the frame motion occurs, in which directions it points, and how its magnitude evolves over the clip. Because it is built from pixel-level motion, this descriptor is independent of hand morphology and of retargeting accuracy. Writing $\boldsymbol{g}_c$ for the descriptor block of cue c, the re-ranking score between an anchor descriptor $\boldsymbol{g}$ and a target descriptor $\boldsymbol{g}'$ is a weighted sum of per-cue similarities,

[2] Our implementation uses Farnebäck's algorithm, but the stage does not depend on this choice: any dense flow field computed identically for anchors and targets serves the same purpose.

$$s_{\text{flow}}(\boldsymbol{g}, \boldsymbol{g}') = \sum_c \lambda_c \, s(\boldsymbol{g}_c, \boldsymbol{g}'_c), \qquad \lambda_c \geq 0, \quad \sum_c \lambda_c = 1, \tag{13}$$

where c indexes the cues that make up the descriptor and λ_c are their mixing weights. We use one fixed setting of λ_c that slightly favors spatial and directional agreement. Since Stage III is a training-free verification signal that does not enter the mined subset $\mathcal{D}_h^*$, neither the training set nor any success rate we report depends on this choice. Re-ranking the top-K_{II} candidates of each anchor by s_{flow} yields the verification ordering that the main text visualizes for three anchors (Fig. 5).

A.5 Bimanual Handling

We collect here the design choices for the two-hand setting, which the main text does not spell out beyond the two shared segments $\boldsymbol{a}_t^{\text{wr}}$ and $\boldsymbol{a}_t^{\text{ft}}$ of Eq. (3).

Per-hand scoring with symmetric fusion. In Stages I and II, every score that depends on the hands is computed *independently* for the left and right hands in the shared space, and the two per-hand values are combined symmetrically into a single bimanual score,

$$s_{\text{bi}} = s^{\text{L}} + s^{\text{R}}, \tag{14}$$

where s^{L} and s^{R} are the same per-hand similarity evaluated on the left and the right hand. This equal-weight combination treats the two hands symmetrically and introduces no hand-specific parameters; since only the ranking induced by the score is used, summing and averaging over the two hands give identical rankings. Eq. (5) is the same rule applied to a distance, where smaller is better.

Mirror augmentation. Human demonstrations carry a handedness bias (e.g. a predominance of right-handed execution), which would leave the left-handed portion of the shared space under-populated. To remove this bias, the preprocessing of the human data includes a random left–right mirror: a geometric reflection that swaps the left and right hands, reflects the wrist-local fingertip and wrist quantities accordingly, and flips the image horizontally. Its only role is to make the handedness distribution of the shared space symmetric; the mirror replaces a demonstration rather than adding a second copy, so it does not change the size of the human pool. The operation is purely geometric and preserves the shared representation of App. A.1, so mirrored targets remain directly comparable to anchors under all similarities above. One consequence is that a mined sample may be the mirrored realization of its source clip. This mirror is applied when the human pool is built and is independent of any data augmentation used during VLA training.

A.6 Retrieval Hyperparameters

Table 2 collects every hyperparameter of the cascade in one place. These values were fixed once and used unchanged for all three tasks and all robot-data scales; we did not tune them against downstream success rate. The cascade therefore has no task- or scale-specific configuration.

B Data and the Human Pool

This appendix specifies the two data sources of Sec. 4.2: the human pool that SiMDex mines and the robot demonstrations that serve as anchors. We state how each corpus is turned into the shared representation of Sec. 3.1, how large each is, and how the human pool is filtered before mining.

B.1 From Egocentric Video to the Shared Descriptor

Source. The human pool is derived from EgoDex (Hoque et al., 2026), a publicly released egocentric manipulation dataset of 829 hours over 194 tabletop tasks, distributed as 338,000 episodes and roughly 90M frames recorded at 30 fps. Each frame carries the 3D position and orientation of the upper-body joints, including 25 joints per hand, together with a per-frame tracking confidence; each episode carries a natural-language description. Two properties make the corpus usable as a retrieval

Table 2: Retrieval hyperparameters of the SiMDex cascade. One configuration is used for all tasks and all robot-data budgets; no value is tuned against downstream success rate.

Stage	Setting	Value
I: Recall	Pose descriptor $\hat{\boldsymbol{p}}$	ℓ_2-normalized, $\mathbb{R}^{15}$ per hand
	Pose retrieval	Euclidean distance
	Sentence embedding $\boldsymbol{e}$	$\mathbb{R}^{384}$, cosine similarity
	Rank-fusion constant κ	60
	Fused candidate set K_{I}	10,000
II: Ranking	Wrist trajectory $\boldsymbol{F}_{\mathrm{wr}}$	31×3
	Fingertip trajectory $\boldsymbol{F}_{\mathrm{ft}}$	30×15
	Waveforms $\boldsymbol{w}_{\mathrm{tr}}, \boldsymbol{w}_{\mathrm{rot}}$	$\mathbb{R}^{10}$ each
	Component fusion ρ	equal-weight rank sum
	Source deduplication	one sample per source episode
	Per-anchor shortlist K_{II}	100
III: Re-ranking	Applied to	top K_{II} candidates per anchor
	Effect on training set	none (verification only)

target for our purpose. First, the hand transforms are given per frame rather than per clip, which is what makes fine-grained motion comparison possible. Second, every episode already comes with language, so the recall stage of Sec. 3.2 needs no additional annotation.

Preprocessing. We apply five operators, in this order, before any sample enters the pool. *(i) Ego-motion removal.* Wrist and fingertip transforms are re-expressed in a body-centered frame. *(ii) Validity filtering.* Frames with absent or low-confidence hand tracking are discarded, as are frames whose wrist speed is out of range. *(iii) Temporal smoothing.* The retained trajectories are smoothed along time. *(iv) Wrist-local retargeting.* They are then mapped onto the shared descriptor of Sec. 3.1. *(v) Handedness mirroring.* A random left–right mirror is applied as specified in App. A.5.

Slicing. Each surviving episode is sliced into frame-level samples at unit stride. A sample pairs the observation and the language instruction at the current frame with a future target of $H = 30$ steps in the shared action space—that is, one sample covers approximately one second of motion at 30 fps. This is the unit of every sample count in this paper, for both sources (App. B.5).

B.2 Quality Filtering for Post-Training

Why the bar is higher here than in pre-training. Human video plays different roles in the two regimes. In large-scale pre-training it shapes the backbone, and its sheer volume makes the result insensitive to any individual sample. In post-training it is mixed with robot demonstrations at a 1:1 ratio and supervises the same action head through the same loss, so a noisy human sample competes directly with a real robot demonstration for gradient. We therefore hold the human data to a stricter standard when building the post-training pool, and it is this filtered pool—not the raw corpus—that both SiMDex and the baseline draw from, so the stricter bar applies to both alike. The filter uses only tracking quality and motion plausibility, not activity category; Table 3 lists the retained pool.

B.3 Pool Statistics

Table 3 reports the provenance and size of the pool. This pool is the human target set $\mathcal{D}_h$ over which the cascade of App. A operates. Both methods see it in full: SiMDex mines $\mathcal{D}_h$, and the baseline draws its matched random subset from the same $\mathcal{D}_h$. The comparison of Sec. 4.3 therefore isolates selection and nothing else: the model, schedule and data budget are identical.

Scale relative to the robot data. The human pool $\mathcal{D}_h$ is $23.7\times$ the size of the default robot-data budget: 32.0M frame-level samples against 1.35M (App. B.4). The subset SiMDex retrieves from it, $\sim$1.49M samples, is 4.65% of the pool and $\sim 1.10\times$ the robot budget. Its size is an outcome of

Table 3: Provenance and size of the human pool. The left column reports the public EgoDex release (Hoque et al., 2026); the right column reports the pool obtained after the preprocessing and quality filtering of Apps. B.1–B.2, which SiMDex mines and from which the baseline draws its matched random subset. Pool episodes are counted if they contribute at least one sample.

	EgoDex release	Post-training pool
Duration	829 h	$\sim$297 h
Episodes	338,000	164,959 (48.8%)
Frame-level samples	$\sim$90M	32,034,551 ($\sim$36%)
Samples per episode	$\sim$265	$\sim$194
Tabletop tasks	194	—

the cascade: the per-anchor top-K_{II} shortlist followed by global deduplication. Those parameters were fixed once and never tuned against downstream success (App. A.6), so this size was not chosen to hit a target fraction of the pool, and the baseline draws the same number at random.

B.4 Robot Demonstrations

Collection. Robot demonstrations are collected by bimanual teleoperation of a two-arm, two-hand platform by trained operators in a controlled laboratory setting, within a single application scenario (industrial assembly) covering the three tasks of Sec. 4.1. Episodes are recorded at 30 fps, and the policy takes a single egocentric view as its visual input, from the platform's head-mounted camera for robot samples and from the wearer's headset for human samples. Keeping the observation single-view and egocentric on both sides is what makes the two sources interchangeable at the model input. Mixing the two therefore needs no view selection and no cross-view alignment.

Scale. Wrist poses, fingertip positions and joint readings are recovered by forward kinematics. They pass through the same descriptor and the same unit-stride, $H = 30$ slicing as the human data. From the raw collection we curate a high-quality subset of $\sim$1.35M frame-level samples, or $\sim$12.4 hours. The raw collection is about twice the size of this curated subset. The additional material comes largely from early teleoperation sessions, when the operators were still building proficiency with the platform. Execution in those sessions is less fluent, and mistakes and interruptions are more frequent. This material is what the $2\times$ budget of Sec. 4.4 draws on; the $1\times$ budget uses the curated subset alone.

B.5 Sample Unit and Normalization

A sample is a single frame paired with its language instruction and a 30-step future target in the shared action space of Sec. 3.1. Both corpora are recorded at 30 fps, sliced at unit stride with the same horizon, and mapped through the same retargeting operator, so one human sample and one robot sample cover the same amount of time. In format, they differ only in which segments of the action vector carry supervision and in the joint slots of the state, which are zero-filled for human samples (Sec. 3.3). All data budgets, including the 1:1 mixture and the baseline's equal human-data budget, are counted in samples; our hour figures are 30 fps conversions, for intuition only.

Normalization. Action targets are normalized per source before training, using precomputed mean and standard deviation statistics. The same statistics are used for SiMDex, the baseline and every scaling configuration. The relative contribution of the two streams is therefore set by the mixture ratio, not by the numerical scale of either source, since normalization removes that scale.

C What Gets Mined

The main paper reports how much the mined data helps; this appendix reports what it *is*. The examples in the figures of this appendix are drawn from the mined subset used for training. Stage III does not alter that subset (Sec. 3.2) and serves only as a check on it.

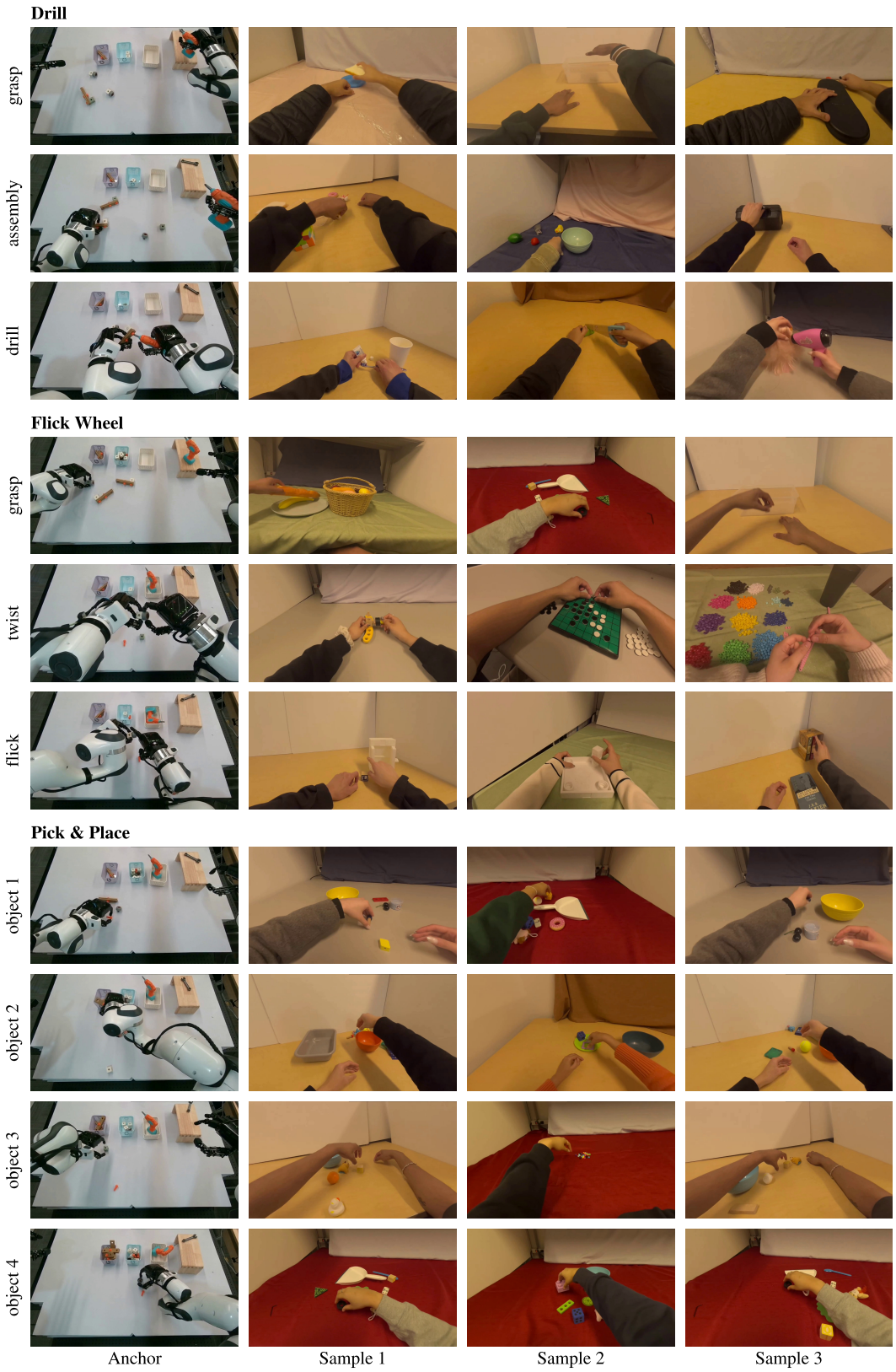

Figure 7: **Every sub-task, and what is mined for it.** Rows follow the three tasks and their stages; the anchor comes first, then three mined human samples, each from a different source episode.

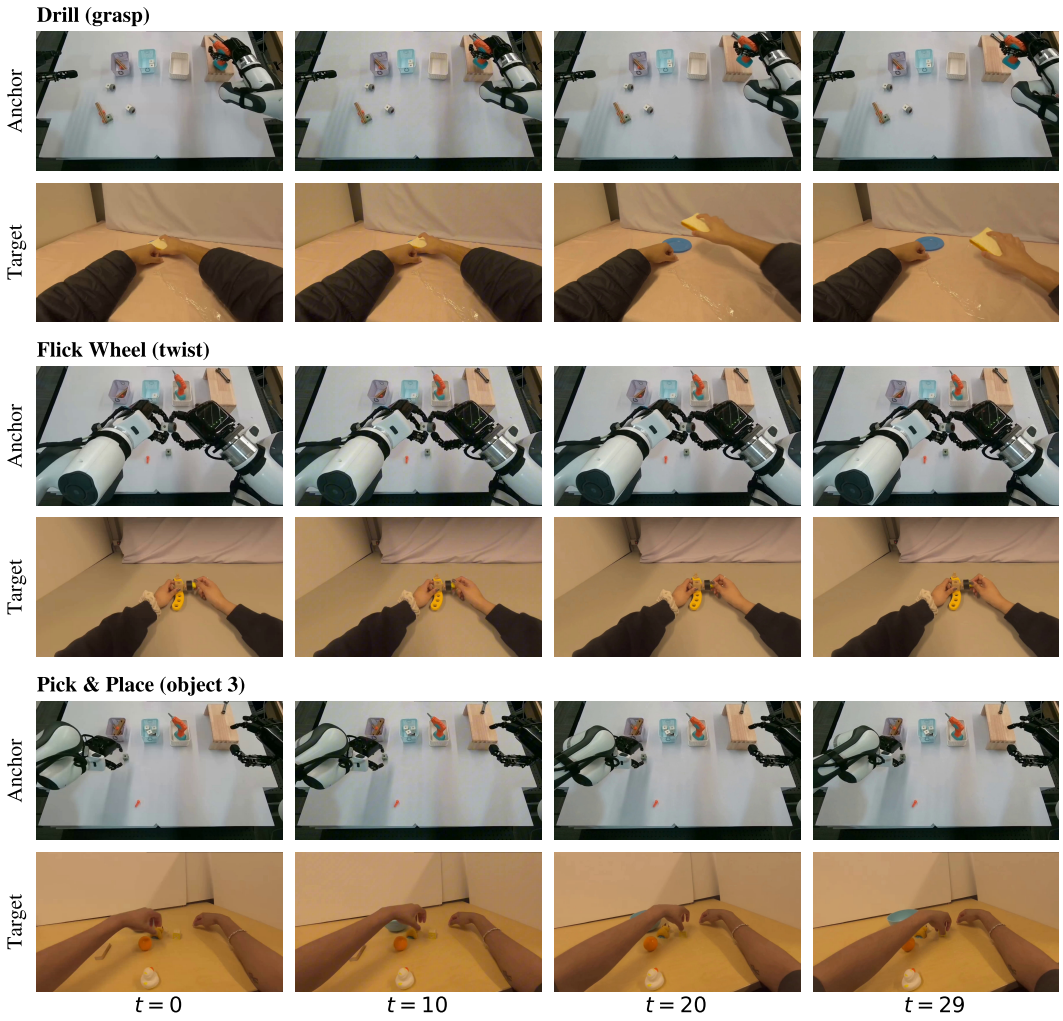

Figure 8: **Anchors and the human samples mined for them.** One sub-task per task, at four positions of the $H = 30$ horizon; within each block, anchor and target use the same frame indices.

C.1 What the Mined Set Covers

Fig. 7 shows all ten robot sub-tasks, covering every stage of every task. The first cell of each row is the robot anchor; the three that follow are human samples mined for it. No sub-task is left out.

C.2 What the Match Is Based On

Fig. 8 takes three sub-tasks, one per task, and shows the anchor and its mined sample as four frames each. The four frames are four positions of the $H = 30$ horizon, and the anchor and the sample use the same frame indices. The useful thing to notice is what the pairs do *not* share: environment, objects and hands all differ, and none of the clips is a recording of the robot's task. What they share is the part the policy is supervised on: the wrist motion and the fingertip configuration over the horizon. This is the intended behavior of a morphology-agnostic action space: matches are judged on motion structure, not on appearance. It is also why visual verification (Stage III) is informative rather than redundant: pixel-level motion carries information that the action-space stages never see.

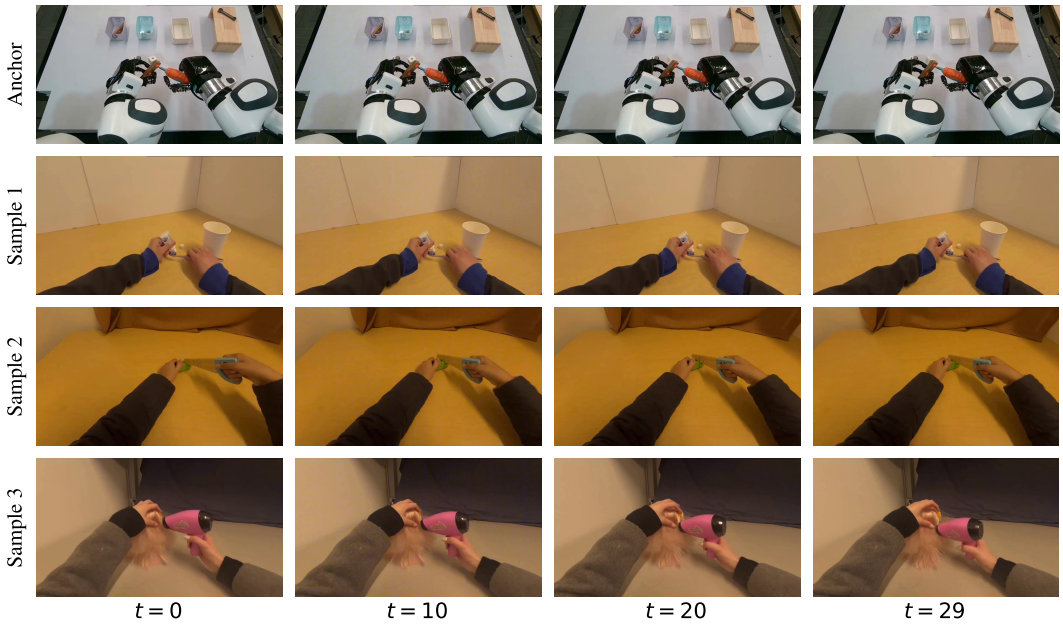

Figure 9: **What is mined for the drilling stage of *Drill*.** Below the anchor: a tube, a saw and a hair dryer, shown at the same four positions of the $H = 30$ horizon. None of them is drilling.

C.3 How Retrieval Fails

Reporting what the cascade retrieves would be incomplete without the cases where it retrieves the wrong thing. Our similarity operates in a purely kinematic action space: wrist 6D pose and fingertip positions. It does not capture contact forces, object state or interaction semantics, and the errors below follow from this. Three types of error recur; we give an example of each.

Language-relevant but action-irrelevant. Recall combines initial hand pose with instruction semantics, and the language side can carry a candidate on an object noun alone. In the *Flick Wheel* row of Fig. 5, the Recall result contains a wheel-like object, but the hand merely grasps it on a plane. The word matches and the motion does not; Ranking addresses this by comparing motion.

Similar motion, different interaction. A candidate can match the anchor's translation and rotation waveforms while the hand is doing something else entirely. The same pinch-and-turn rhythm appears in unscrewing, in wiping, and in rotating an object to inspect it. Ranking narrows this considerably, since the fingertip trajectories have to agree as well. But hands in distant positions can still score highly when their per-frame values happen to align, as the Ranking column of Fig. 5 shows.

The right tool in hand, the wrong interaction. For the drilling stage of *Drill*, the mined clips show a tube being squeezed, a saw, and a hair dryer (Fig. 9), and none of the EgoDex task labels of the clips mined for *Drill* corresponds to drilling. Holding a hand-held appliance and holding a drill are nearly identical in wrist pose and fingertip configuration. The difference lies in contact, which is outside the action space, and in motion amplitude: the mined clips are oscillatory while the robot's drilling motion is nearly static, a difference the action-space scores largely discount, since they are built on cosine similarities and magnitude-normalized waveforms. Nothing in the ranking marks such candidates as wrong, because on every signal it observes they look like correct matches.

The re-ranking stage of Sec. 3.2 targets exactly these cases: optical flow is computed on pixels, independent of the retargeting that both preceding stages depend on, so a candidate that ranks first in the action space can be demoted by where and how its motion actually occurs in the frame. It reorders the same candidates that form the mined subset, so the samples it ranks first are already in the training data. For the cost reason given in Sec. 3.2, re-ranking in this work checks the mined subset rather than filtering it; since its descriptor summarizes a single clip, it can be cached per clip, which would make it affordable as a filter during mining, a step we leave to future work (Sec. 7).

D EXPERIMENTAL SETUP

This appendix expands the implementation details of Sec. 4.2. SiMDex changes only the *source* of the human data; the base model, the training procedure, and every hyperparameter are left untouched. At every robot-data budget, SiMDex and the baseline are therefore trained under identical settings. The only variable in any comparison is thus which human samples are used.

D.1 BASE MODEL

Our base model is the π_0-like flow-matching VLA for dexterous manipulation introduced in Sec. 4.2; the baseline uses the identical model. We summarize here only the components needed to interpret our experiments and to make the cross-embodiment supervision of Sec. 3.3 precise; the remaining components are those of the base model, and SiMDex modifies none of them.

Inputs and modules. At each timestep the model receives a single camera view $\boldsymbol{I}_t$, a language instruction l, and a proprioceptive state $\boldsymbol{s}_t \in \mathbb{R}^{86}$. A vision-language backbone encodes the observation and the instruction. A state adapter projects $\boldsymbol{s}_t$ into the backbone hidden space. An action transformer, conditioned on the backbone output, then denoises a noisy action chunk into the predicted chunk $\boldsymbol{a}_{t:t+H} \in \mathbb{R}^{H\times 88}$ over the horizon $H = 30$. The exact composition of the 88 action dimensions and the 86 state dimensions is given in App. D.3. As stated in Sec. 3.3, this path is shared by robot and human samples; it is not conditioned on the source of a sample. Human samples differ only in which state entries are populated and which action entries are supervised.

Training objective. Training uses the masked flow-matching objective of Eq. (1) in the main text. Given a ground-truth action chunk $\boldsymbol{a} \in \mathbb{R}^{H\times 88}$, we sample noise $\boldsymbol{\epsilon} \in \mathbb{R}^{H\times 88}$ with entries drawn i.i.d. from $\mathcal{N}(0,1)$ and a timestep $\sigma \in [0,1]$, form the interpolated state $\boldsymbol{x}_\sigma$ together with its flow target $\boldsymbol{u}_\sigma$, and regress the predicted velocity $\hat{\boldsymbol{u}}_\sigma = f_\theta(\boldsymbol{x}_\sigma, \boldsymbol{I}_t, l, \boldsymbol{s}_t)$ under the masked mean squared error of Eq. (1). The timestep σ is not drawn uniformly from $[0,1]$; instead, we sample

$$v \sim \mathrm{Beta}(1.5,\, 1.0), \qquad \sigma = 0.999\,(1 - v), \tag{15}$$

where $v \in [0,1]$ is an auxiliary scalar used only to generate the timestep $\sigma \in [0,1]$. The model regresses the flow target rather than the clean action or the noise itself. The action loss is the only objective active in all runs reported in this paper. The backbone's language-modeling loss and the optional auxiliary losses are disabled throughout, for SiMDex and the baseline alike.

What is held fixed. The architecture, the model capacity, the input and output dimensions, the optimizer, the schedule, and the number of training steps are identical across all compared methods. SiMDex requires only a backbone that reads a shared action layout with a per-dimension mask.

D.2 TRAINING HYPERPARAMETERS

All models—SiMDex, the baseline, and every scaling configuration—are trained with the hyperparameters of Table 4. The sole difference between SiMDex and the baseline is which human samples are used. We emphasize that none of these values were tuned per method: a single configuration, including the step count at $0.25\times$, is used for every run reported in this paper.

Two independently weighted data streams. The robot demonstrations and the mined human subset are not merged offline into a single dataset. They are registered as two separate sources in the training pipeline and multiplexed by source weight during loading, so that the two contribute in a 1:1 ratio in expectation rather than through a hard per-batch quota. Because the mixture is governed by source weights and not by dataset size, the ratio is unaffected by how many samples each source happens to contain. Action targets are normalized per source before training (App. B.5).

Preprocessing and augmentation. The mined subset and the randomly sampled subset of the baseline are consumed by the same pipeline: the mined data receives no additional filtering, reweighting or preprocessing of its own. All data-augmentation settings are likewise identical for every method, so augmentation cannot explain any difference between SiMDex and the baseline.

Table 4: Training hyperparameters, shared by SiMDex, the baseline, and all scaling configurations. Unlisted model and optimizer settings are also shared; no value was tuned per method.

Setting	Value
Optimizer	AdamW
Weight decay	0.1
Peak learning rate	3×10^{-5}
LR schedule	cosine, 5% linear warmup
Minimum learning rate	1×10^{-6}
Gradient clipping	enabled
Mixed precision	bf16
Training objective	masked MSE (flow matching)
Noise distribution	entries i.i.d. $\mathcal{N}(0, 1)$
Timestep sampling	$\sigma = 0.999\,(1 - v)$, $v \sim \mathrm{Beta}(1.5, 1.0)$
Regression target	flow target (velocity)
Supervised dims (robot / human)	$88/42$
Action horizon H	30
Training steps	40K (20K for the $0.25\times$ scale)
Robot : human mixture	1:1 (equal-weight sampling)
Random seed	42

Table 5: Composition of the 88-dimensional action chunk. For human samples, the "shared" blocks are supervised, while the "robot-only" blocks are zero-filled and masked out.

Segment	Symbol	Contents	Dim.	Human supervision
Bimanual wrist actions	$\boldsymbol{a}_t^{\mathrm{wr}}$	$(\boldsymbol{d}_t^{\mathrm{L}}, \boldsymbol{d}_t^{\mathrm{R}})$, 6 per hand	12	shared
Arm-joint actions	$\boldsymbol{a}_t^{\mathrm{ar}}$	joint commands, 7 per arm	14	robot-only
Hand-joint commands	$\boldsymbol{a}_t^{\mathrm{hd}}$	joint commands, 16 per hand	32	robot-only
Bimanual fingertip targets	$\boldsymbol{a}_t^{\mathrm{ft}}$	$(\boldsymbol{f}_t^{\mathrm{L}}, \boldsymbol{f}_t^{\mathrm{R}})$, 15 per hand	30	shared
Total ($\boldsymbol{a}_t$)			88	42 supervised

D.3 Unified Action and State Layout

The policy predicts an 88-dimensional action chunk and consumes an 86-dimensional state. We give the exact composition, since it fixes which dimensions carry gradient for a human sample.

Action layout. Table 5 lists the four blocks of the action chunk. Two of them—the bimanual wrist actions $\boldsymbol{a}_t^{\mathrm{wr}}$ and the bimanual fingertip targets $\boldsymbol{a}_t^{\mathrm{ft}}$—are expressed relative to the current wrist and are therefore defined for both embodiments; together they form the shared action $\boldsymbol{a}_t^{\mathrm{sh}} \in \mathbb{R}^{42}$ of App. A.1. The other two carry mechanism-level commands that have no human counterpart.

State layout. The state $\boldsymbol{s}_t \in \mathbb{R}^{86}$ concatenates the arm joint readings of both arms (7 per arm, 14 in total), the hand joint readings of both hands (21 per hand, 42 in total), and the current fingertip configurations $\boldsymbol{p}_t \in \mathbb{R}^{15}$ of both hands (15 per hand, 30 in total). Note that the hand joint *readings* (21 per hand) and the hand joint *commands* in the action chunk (16 per hand) live on different parameterizations, so the two counts need not match. For a human sample the 56 robot-specific state entries—arm and hand joints—have no counterpart and are zero-filled, while the 30 fingertip entries are populated; the state adapter consumes the same 86-dimensional vector in both cases.

Effective supervision for human samples. Under the mask $\boldsymbol{m}$ of Sec. 3.3, a human sample contributes gradient through the 12 wrist dimensions and the 30 fingertip dimensions, i.e. 42 of the 88 action dimensions. The mask is applied per sample rather than per batch, so a single batch may freely mix the two sources. Robot samples are supervised on all 88 dimensions.

D.4 Equal-Budget Comparison

The comparison of Table 1 is designed so that *selection* is the only variable.

Matched human-data budget. SiMDex trains on its $\sim$1.49M mined samples. The baseline draws an equal number of samples at random from the same $\sim$32M human pool—the same pool, the same preprocessing, the same quality filtering. Neither method sees human data that the other could not have seen; they differ only in which members of the pool end up in the training set.

Everything else held fixed. The robot demonstrations, the model, the random seed, and every other entry of Table 4 are identical between the two runs. Both human subsets enter training through the same two-source pipeline described in App. D.2, with the same source weight. Apart from evaluation noise, any difference in success rate is therefore attributable to which human samples are used, and not to model capacity, training budget, data volume, or mixture composition.

D.5 Data Scaling Ablation

The scaling study of Sec. 4.4 varies the robot-data budget while holding the human data fixed. We describe how the budgets are built and the controls that make the four scales comparable.

The four robot-data budgets. The default budget ($1\times$, $\sim$1.35M frame-level samples, $\sim$12.4 hours) is a high-quality subset curated from the raw teleoperation collection, and is used everywhere else in the paper. The $0.5\times$ ($\sim$6.2h) and $0.25\times$ ($\sim$3.1h) budgets are nested random subsets of it. The $2\times$ budget ($\sim$24.9h) is obtained by relaxing the quality filter over the *same* raw collection—the same platform, the same tasks, the same operators—rather than by collecting new demonstrations. We note the implication plainly: the largest budget trades some per-sample quality for quantity, so the $2\times$ column should be read as "more robot data of mixed quality" rather than as "twice as much data of the $1\times$ standard". This does not affect the comparison at any given scale, since SiMDex and the baseline consume exactly the same robot data at every budget. It does, however, bear on comparisons across scales, such as SiMDex at $0.5\times$ against the baseline at $2\times$, where about 6 hours of curated robot data are set against about 25 hours of mixed-quality robot data.

Nested subsampling. The smaller budgets are drawn from the $1\times$ set with a fixed random seed (53), so the scales are nested: every demonstration at $0.25\times$ is also present at $0.5\times$ and $1\times$. The same subsets are used for SiMDex and for the baseline, so both see identical robot data.

Fixed human data. The human data is identical across all scales. SiMDex reuses the same mined subset at every robot-data budget, and the baseline reuses the same randomly sampled subset, so the scaling curves isolate the effect of robot-data quantity and do not confound it with a change in the human data. In particular, the mined subset is *not* re-mined at each scale; it is mined once against the $1\times$ anchors and reused unchanged. At $0.5\times$ and $0.25\times$, the mined subset is thus retrieved with anchors from the full $1\times$ set, which includes demonstrations these scales do not train on.

Training-step control. All scales train for the 40K steps of Table 4, with one deliberate exception: the $0.25\times$ scale is trained for 20K steps to avoid overfitting on its very small robot-data budget. This is a control against overfitting, not a tuned advantage, and it applies equally to both methods.

Evaluation rounds. Unlike the main results, which average over two rollout rounds ($2 \times 5 = 10$ trials), each scaling configuration is evaluated with a single rollout round. The per-task values in the $1\times$ column of Fig. 4 therefore differ from Table 1, as noted in the caption of Fig. 4.

E Evaluation Protocol

This section specifies how the real-robot rollouts of Sec. 4.1 are run and scored. Our goal is an evaluation fine-grained enough to credit partial progress, identical across all compared methods, and reproducible from the description alone. Every element below—the task decomposition, the instruction set, the scoring rubric, the initialization, and the scoring personnel—is shared by SiMDex and the baseline at every data budget, so the only variable across methods is the human-data source.

E.1 Tasks, Language Instructions, and Scoring Rubric

The three evaluation tasks of Sec. 4.1 are decomposed into ordered sub-tasks: three for Drill, three for Flick Wheel, and four for Pick & Place, ten in total. These sub-tasks are the unit of scoring in Table 1, and they are also the unit of language conditioning: each sub-task has one fixed instruction, and the policy is conditioned on the instruction of the sub-task currently in progress. Table 6 lists all ten instructions verbatim with the observable steps on which each sub-task is scored.

Table 6: **Evaluation tasks, language instructions, and scoring rubric.** The sub-tasks match the columns of Table 1; instructions are given verbatim as fed to the policy. Each sub-task is worth one point, split equally over the listed steps, and is complete when all of its steps are achieved.

Task	Sub-task	Language instruction	Scored steps (points each)
I: Drill (max 3)	Grasp	`pick up the electric drill`	reach the drill; lift it and hold it stably in one hand; press the trigger ($\frac{1}{3}$)
	Assem.	`pick up the bar-to-cube assembly`	reach the assembly; grasp it; place it in alignment with the held drill ($\frac{1}{3}$)
	Drill	`unscrew the cube with screw-holes from the bar-to-cube assembly`	bring the drill to the assembly; make contact; drive the screw-hole cube off the bar ($\frac{1}{3}$)
II: Flick Wheel (max 3)	Grasp	`pick up the wheel-to-cube assembly`	reach the assembly; lift it and hold it stably ($\frac{1}{2}$)
	Twist	`unscrew the bolted wheel from the wheel-to-cube assembly`	bring two fingers to the wheel; make contact; turn the wheel until the bolt is fully released ($\frac{1}{3}$)
	Flick	`flick the wheel-to-cube assembly with index finger`	contact and flick the wheel with the index finger; the wheel comes off the assembly ($\frac{1}{2}$)
III: Pick & Place (max 4)	Obj.1	`put the plastic screw into the blue basket`	reach the named object; grasp it; place it inside the blue basket ($\frac{1}{3}$)
	Obj.2	`put the bolted double-strip bar into the blue basket`	
	Obj.3	`put the cube with screw-holes into the blue basket`	
	Obj.4	`put the bolted wheel into the blue basket`	

Bimanual structure of Drill. Task I is bimanual by construction: one hand acquires and holds the powered drill while the other acquires the workpiece and presents it to the tool, and only then is the tool actuated. The three sub-tasks therefore test, in order, single-hand tool acquisition, second-hand workpiece acquisition with inter-hand alignment, and powered tool use under that alignment. This is the property that makes Drill the most coordination-demanding of the three tasks, and it is why a failure to establish the alignment in the second sub-task leaves the third unreachable.

Instruction switching. Within a trial the policy is conditioned on one sub-task instruction at a time. In Drill and Flick Wheel, the instruction advances to the next sub-task when the current one is complete and does not advance otherwise, so a trial that stalls remains conditioned on the sub-task it failed. The one exception is the Drill Grasp: once the drill is held stably, a waiting period is allowed for the trigger press, and the instruction advances even if the trigger is not pressed within it. In Pick & Place, the instruction advances to the next object when the current one is placed or its attempt is terminated under the rules of Sec. E.3. The switching decision is made by the evaluation team (Sec. E.4) under the same rules for every method, and the policy receives no other signal about task state. This protocol is part of the harness and identical for both methods at every budget.

E.2 Step-Based Sub-Task Scoring

Each sub-task is worth one point, divided equally among the observable steps of Table 6: three steps of $\frac{1}{3}$ for most sub-tasks, and two steps of $\frac{1}{2}$ for the Grasp and Flick sub-tasks of Flick Wheel. A sub-task's score is the sum of the steps achieved, taking values in $\{0, \frac{1}{3}, \frac{2}{3}, 1\}$ or $\{0, \frac{1}{2}, 1\}$, so that a rollout that gets most of the way through a contact-rich stage is distinguished from one that never engages the object. Steps are credited only during their own sub-task: a Drill trigger press that occurs during a later sub-task, for example, does not retroactively earn the Grasp step.

Sequential dependence. Drill and Flick Wheel are sequential: their sub-tasks form a chain, and when a stage is not completed the subsequent stages cannot be reached and receive 0. The one step outside this chain is the trigger press of the Drill Grasp, which is scored but is not a precondition for assembly (Sec. E.1). Pick & Place is different in kind: its four objects are attempted in the fixed order of Table 6, but a failure on one object does not prevent the robot from attempting the next, and each object is therefore scored independently. This distinction is visible in Table 1: the sub-task scores of Drill and Flick Wheel do not increase along the chain for either method, whereas those of Pick & Place do not follow this pattern—the smallest object, the plastic screw (Obj.1), is the hardest object for both methods and scores below all three objects attempted after it.

From sub-task scores to reported numbers. A trial's score is the sum of its sub-task scores, so the per-trial maximum is 3 for Drill, 3 for Flick Wheel, and 4 for Pick & Place. The Total column of Table 1 is the mean of these per-trial sums over all trials, and the Success Rate is $\mathrm{Total}/\mathrm{max}\times 100\%$. The Overall Success Rate is the unweighted mean of the three task success rates.

E.3 Trial Structure, Initialization, and Reset

Trials and rounds. Each task in Table 1 is evaluated over 10 trials in two rounds of five. The five trials of a round use five distinct object layouts, indexed 1–5; the second round repeats the same five layouts in the same order. Each layout is therefore evaluated twice per method, and the two rounds differ only by repetition, not by initial conditions. The two rounds are run as separate evaluation sessions, over which the reported standard deviations are computed (Sec. E.5). The scaling study of Sec. 4.4 uses a single round of five trials per configuration instead (App. D.5).

Controlled initialization. Initialization is governed by a fixed table-level mask: each of the five trial indices has one prescribed object layout, and before every trial the workspace is restored to that layout. Restoration is verified against a third-person camera view rather than by eye, so a layout is reproduced from trial to trial and from method to method to within camera-verified placement error. Initialization is controlled along two axes. Across methods, trial index i always uses layout i, so SiMDex and the baseline start from identical initial conditions at every budget and are directly comparable trial by trial. Across trial indices, the five layouts differ, so the five trials of a round probe five distinct object configurations rather than a single one. The five layouts were chosen at random, and none of them reproduces a configuration that occurs in the training demonstrations: every evaluation trial therefore starts from an object configuration the policy has not seen during training. The comparison is thus fair within each trial while still testing robustness to spatial variation—neither method can benefit from an easier draw, and neither is evaluated on only one layout.

Termination, retries, and reset. A trial ends when the task's final sub-task is completed. When it is not completed, termination is decided by two rules. The first is a stall allowance: a rollout that is still moving but is evidently not converging on the criterion—repeating the same approach, or advancing very slowly—is given roughly two further minutes, and if nothing changes it is terminated and scored at the progress reached. The second is an attempt limit for sub-tasks that require picking an object off the table: the hand is allowed at most three attempts, and the sub-task is terminated after the third failure. One attempt is one approach–close–lift cycle, counted from the moment the hand begins to descend toward the object until it either lifts the object clear of the table or ends up without it; a grasp that is established and then dropped consumes an attempt. Retries are given by continued execution rather than by re-prompting: the instruction is not re-sent, and the policy is not reset or re-initialized within a trial. No human intervention is permitted while a trial is running; the operators observe and score from outside the workspace and act only to halt the robot for safety.

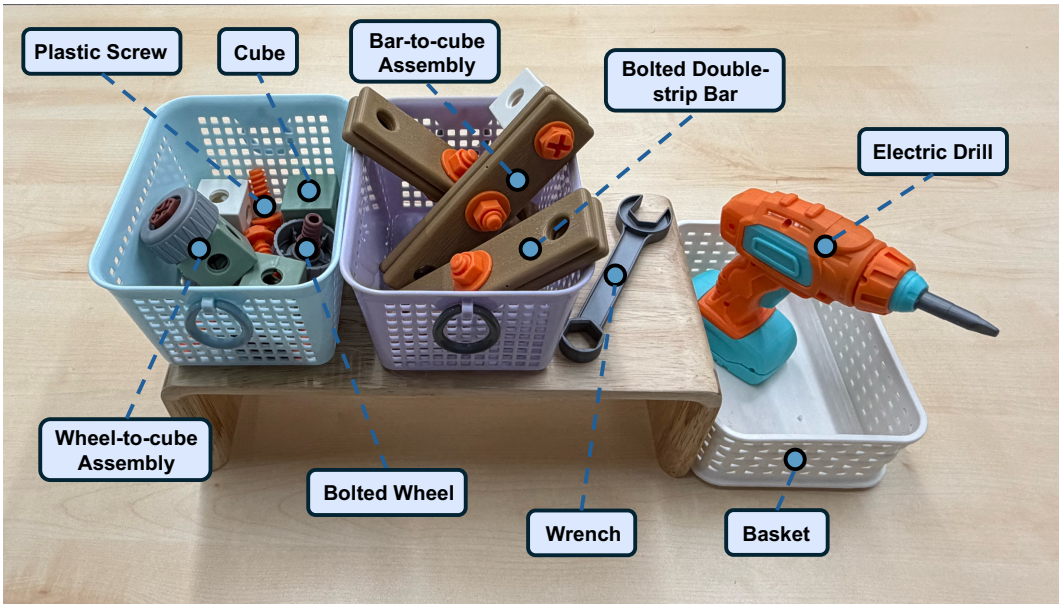

Figure 10: **Objects and tools used in the real-robot evaluation.** The drill is the tool; bars, wheels, cubes, and screws are manipulated parts; baskets are Task III's placement targets.

Between trials the workspace is restored to the layout of the next trial index as described above. The same termination, retry, and reset rules apply to every method and every trial.

E.4 Independent Execution and Scoring

To remove any possibility of bias toward our own method, the rollouts are executed and scored by a dedicated team of trained operators that is independent of the authors. The authors take no part in running the trials, in switching instructions during a trial, or in assigning scores. The same team and the same rubric of Table 6 are applied to SiMDex and the baseline at every budget, so no method receives preferential judgment. The evaluation is moreover blind to method identity: the operators are not informed which method produced the policy under evaluation. This matters in particular for the instruction switching of Sec. E.1, whose timing rests on a human judgment: since the person making that judgment does not know whether SiMDex or the baseline is running, the judgment cannot be systematically biased toward either. Consistent with the ethics statement of the main paper, no person entered the robot workspace during autonomous evaluation; trials were observed and scored from outside it by personnel who could halt the robot at any time.

E.5 Aggregation and Variance Reporting

What the reported numbers are. Each sub-task entry of Table 1 is the mean of that sub-task's score over all 10 trials of the task. The reported $\pm$ is *not* the standard deviation over the ten individual trials: it is the sample standard deviation over the two round-level means ($n = 2$, i.e. $|a - b|/\sqrt{2}$), where each value is the mean of the five trials of one round. It therefore measures the reproducibility of a full evaluation session rather than trial-to-trial spread, and with $n = 2$ it should be read as an indication of session-level stability, not a confidence interval. We report it this way because the two rounds are the unit at which the evaluation was repeated. Because the two rounds repeat the same five layouts (Sec. E.3), this $\pm$ compares two evaluations under identical initial conditions: it isolates the run-to-run repeatability of the policy and hardware, and does not absorb layout variation.

E.6 Objects, Tools, and Workspace

Fig. 10 shows the objects and tools used in the evaluation: a powered drill, bolted double-strip bars, wheels, cubes with screw holes, plastic screws, and the baskets that serve as placement targets. The

parts are industrial-assembly hardware with tight tolerances, which is what makes the tasks contact-rich: the screw-hole cube and the bolted wheel must be driven or turned off their mating parts rather than pulled, and the plastic screw is small enough that a grasp must be precise to succeed at all. The identical object set, in the same physical condition, is used for every method and every trial; objects are not substituted or re-selected between methods. Details of the robot platform and teleoperation hardware are omitted here for anonymity and are not required to interpret the protocol: the evaluation is defined entirely by the tasks, instructions, rubric, and layouts specified above.

F Additional Results

This section reports the full numbers behind Figure 4, which the main text shows only as bar charts. We first define the four robot-data budgets and give per-sub-task scores for every task, budget, and method, then add conditional pass rates, qualitative rollouts, and the multi-view recording setup.

F.1 Complete Data-Scaling Results

Table 7 defines the four robot-data budgets. The mined human data is identical across all budgets, as are the model and every hyperparameter, so the only variable is the robot-data budget (App. D.5). The one exception is that the smallest budget ($0.25\times$) is trained for 20K steps rather than 40K, to avoid overfitting on a very small robot-data budget; this applies equally to both methods.

Table 7: **Robot-data budgets used in the scaling study.** Mined human data is fixed at $\sim$1.49M samples throughout. Samples are frame-level, scaled from the $1\times$ setting of Sec. 4.2.

Robot-data budget	Demonstrations	Frame-level samples	Training steps
$0.25\times$	$\sim$3.1 h	$\approx$0.34M	20K
$0.5\times$	$\sim$6.2 h	$\approx$0.68M	40K
$1\times$	$\sim$12.4 h	$\approx$1.35M	40K
$2\times$	$\sim$24.9 h	$\approx$2.71M	40K

Table 8 gives the complete per-sub-task results. Unlike Table 1 in the main text, all scaling results come from a single rollout round of five trials per task, whereas Table 1 averages two rounds (ten trials). The $1\times$ column of Table 8 therefore differs numerically from Table 1, as already noted in the caption of Figure 4. Per-task success rate is the task's composite score divided by its maximum, and the overall success rate is the arithmetic mean of the three per-task success rates.

What the full table adds. The table makes three statements of the main text checkable cell by cell. First, SiMDex's overall success rate exceeds the baseline at all four budgets, while the baseline's falls monotonically from 50.9% at $2\times$ to 29.4% at $0.25\times$. Second, Flick Wheel benefits at all four budgets and Pick & Place at all but the smallest. Third, the Drill gap varies in sign; with one round of five trials per configuration, single-budget per-task differences are only indicative.

F.2 Sub-task Conditional Pass Rates

In Drill and Flick Wheel the sub-tasks form a chain (App. E.2), so a zero at one stage forces every later stage to zero as well. Raw sub-task scores therefore conflate two different things: how often the policy reaches a stage at all, and how well it does once it is there. To separate the two, we report the conditional pass rate, defined as a stage's score divided by the score of the stage immediately before it. Table 9 gives these rates for Drill and Flick Wheel. Pick & Place is excluded because its four objects are scored separately rather than forming a chain. All values are computed directly from the unrounded sub-task scores behind Table 1 and introduce no new data.

The advantage compounds along the chain on Flick Wheel. On Flick Wheel, SiMDex's advantage is not confined to the initial grasp. The conditional pass rate rises from 0.22 to 0.55 at Twist and from 0.00 to 0.21 at Flick. In other words, the relative gap between the two methods widens the further the chain moves toward fine-grained finger behavior. This is consistent with our claim that mined human data supplies the fine-grained hand skills that randomly sampled data does not.

Table 8: **Complete results behind Figure 4.** Sub-task scores are means over the five trials of a single rollout round (for $1\times$, the first round of Table 1). Total is the composite score; Success Rate $=$ Total / max $\times 100\%$. Overall SR is the mean of the three per-task success rates. $\Delta =$ SiMDex $-$ GR-Dexter, computed from unrounded scores, so it may differ by one unit in the last digit from the difference of the printed entries; the success-rate deltas are the values printed in Figure 4.

Task I: Drill (max 3)

Budget	Method	Grasp	Assem.	Drill	Total	SR(%)↑
$0.25\times$	GR-Dexter	0.47	0.27	0.07	0.80	26.6
	SiMDex (Ours)	**0.60**	**0.33**	**0.13**	**1.07**	**35.5**
	Δ	$+0.13$	$+0.07$	$+0.07$	$+0.27$	$+8.9$
$0.5\times$	GR-Dexter	0.47	0.40	**0.20**	1.07	35.5
	SiMDex (Ours)	**0.67**	**0.67**	0.13	**1.47**	**48.9**
	Δ	$+0.20$	$+0.27$	-0.07	$+0.40$	$+13.4$
$1\times$	GR-Dexter	**0.80**	**0.80**	**0.33**	**1.93**	**64.5**
	SiMDex (Ours)	0.67	0.40	0.07	1.13	37.7
	Δ	-0.13	-0.40	-0.27	-0.80	-26.7
$2\times$	GR-Dexter	**1.00**	0.87	0.20	**2.06**	**68.8**
	SiMDex (Ours)	0.74	0.87	**0.26**	1.87	62.2
	Δ	-0.26	0.00	$+0.07$	-0.20	-6.6

Task II: Flick Wheel (max 3)

Budget	Method	Grasp	Twist	Flick	Total	SR(%)↑
$0.25\times$	GR-Dexter	0.50	0.00	0.00	0.50	16.7
	SiMDex (Ours)	**0.60**	**0.20**	**0.10**	**0.90**	**30.0**
	Δ	$+0.10$	$+0.20$	$+0.10$	$+0.40$	$+13.3$
$0.5\times$	GR-Dexter	0.70	0.27	0.10	1.07	35.5
	SiMDex (Ours)	**0.80**	**0.47**	**0.20**	**1.47**	**48.9**
	Δ	$+0.10$	$+0.20$	$+0.10$	$+0.40$	$+13.3$
$1\times$	GR-Dexter	0.50	0.00	0.00	0.50	16.7
	SiMDex (Ours)	**0.90**	**0.53**	**0.10**	**1.53**	**51.1**
	Δ	$+0.40$	$+0.53$	$+0.10$	$+1.03$	$+34.5$
$2\times$	GR-Dexter	0.60	0.07	0.00	0.67	22.2
	SiMDex (Ours)	**0.70**	**0.27**	0.00	**0.97**	**32.2**
	Δ	$+0.10$	$+0.20$	0.00	$+0.30$	$+10.0$

Task III: Pick & Place (max 4)

Budget	Method	Obj.1	Obj.2	Obj.3	Obj.4	Total	SR(%)↑
$0.25\times$	GR-Dexter	0.20	**0.87**	**0.40**	**0.33**	**1.80**	**45.0**
	SiMDex (Ours)	0.20	0.80	0.40	0.33	1.73	43.2
	Δ	0.00	-0.07	0.00	0.00	-0.07	-1.8
$0.5\times$	GR-Dexter	0.33	0.73	0.40	0.60	2.06	51.5
	SiMDex (Ours)	**0.47**	**1.00**	**0.93**	**0.67**	**3.07**	**76.7**
	Δ	$+0.13$	$+0.27$	$+0.54$	$+0.07$	$+1.01$	$+25.2$
$1\times$	GR-Dexter	0.26	0.60	0.60	0.73	2.19	54.8
	SiMDex (Ours)	**0.80**	**0.87**	**1.00**	**0.80**	**3.47**	**86.7**
	Δ	$+0.54$	$+0.27$	$+0.40$	$+0.07$	$+1.28$	$+31.9$
$2\times$	GR-Dexter	0.26	0.87	0.53	0.80	2.46	61.6
	SiMDex (Ours)	**0.53**	**0.93**	**0.80**	0.80	**3.07**	**76.7**
	Δ	$+0.27$	$+0.07$	$+0.27$	0.00	$+0.60$	$+15.1$

Overall Success Rate (mean of the three per-task rates)

Budget	GR-Dexter	SiMDex (Ours)	Δ
$0.25\times$	29.4	**36.2**	$+6.8$
$0.5\times$	40.9	**58.2**	$+17.3$
$1\times$	45.3	**58.5**	$+13.2$
$2\times$	50.9	**57.0**	$+6.2$

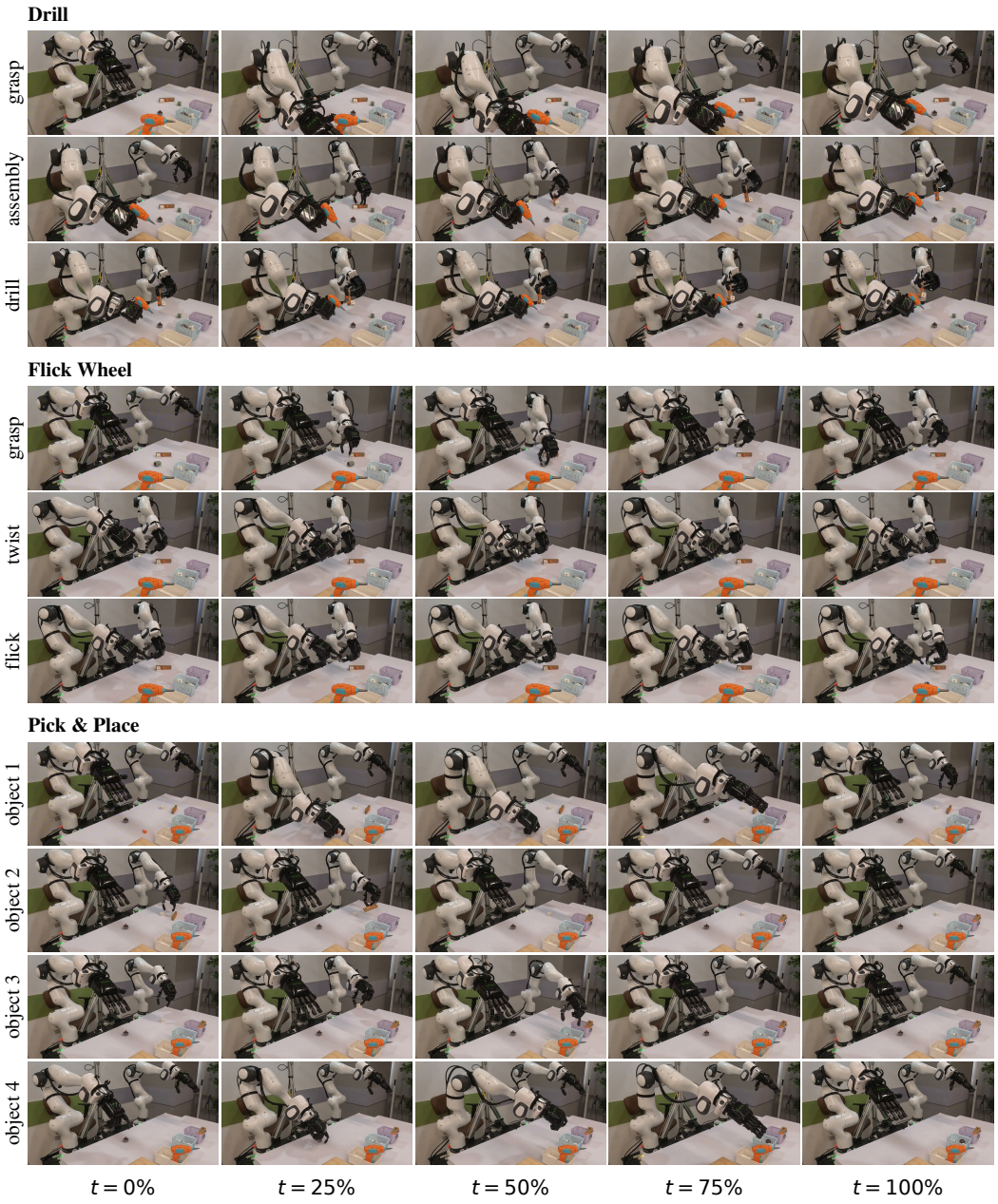

Figure 11: **SiMDex rollouts on all ten evaluation sub-tasks.** One row per sub-task, with columns sampling progress within that sub-task ($t = 0\%$ to $t = 100\%$) rather than wall-clock time. Each row is a single rollout and not a restatement of the ten-trial means of Table 1; see Sec. F.3.

On Drill, the drilling stage itself is unchanged. On Drill, the conditional pass rate of the drilling stage itself is unchanged between the two methods, at 0.42 for both. The overall gap on Drill under the standard setting comes from the two earlier and more generic stages, not from the drilling action itself. This matches App. C.3, where no clip retrieved for the drilling stage shows drilling.

How to read these numbers. With two rounds per method, these are trends, not exact values. This is especially true of Drill, whose Assem. stage carries a standard deviation of ± 0.33 in Table 1. They should therefore be read as trends, not as precise point estimates.

Table 9: **Conditional pass rates on the two strictly sequential tasks.** Stage 1 is the raw score; Stages 2 and 3 are conditioned on the preceding stage, using the unrounded scores behind Table 1.

Task	Method	Stage 1	Stage 2 | 1	Stage 3 | 2
Drill	GR-Dexter	**0.80**	**1.00**	0.42
	SiMDex (Ours)	0.73	0.86	0.42
Flick Wheel	GR-Dexter	0.60	0.22	0.00
	SiMDex (Ours)	**0.85**	**0.55**	**0.21**

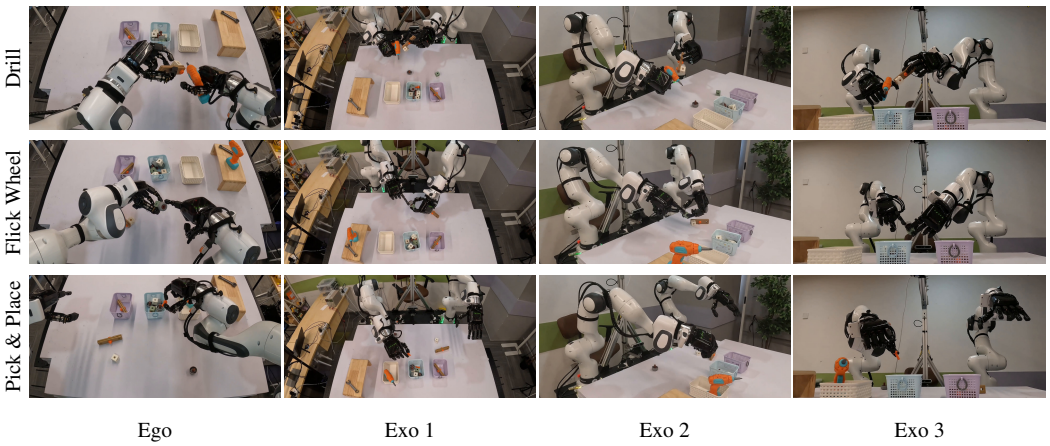

Figure 12: **The four synchronized camera views.** One row per task: the same instant of one rollout from all four cameras. The exocentric cameras are used for recording only, never in training.

F.3 Qualitative Rollouts

Figure 11 shows SiMDex rollouts on all ten sub-tasks, one row per sub-task. The five columns of each row sample progress *within* that sub-task, taken at even intervals from the start of the sub-task to its end, and labeled $t = 0\%$ through $t = 100\%$. We sample by progress rather than by wall-clock time because the three tasks were recorded at different playback speeds, so sampling by time would make the rows incomparable across tasks. All frames come from the Exo 2 side-on camera of Figure 12, the one view that shows both the state of the bench and the contact inside the hand. Every frame in this subsection and the next is cropped only to remove the top and bottom bands carrying the burnt-in text overlay, with no zoom and no reframing of any kind.

These are single rollouts that illustrate what the sub-task columns of Table 1 mean, not the ten-trial means: the Drill rollout shown does not complete the drilling sub-task. The Flick Wheel row comes from an additional rollout recorded outside the scored trials, shown to illustrate the behavior rather than typical performance; in the scored trials of Table 1, that sub-task averages 0.10.

F.4 Multi-View Recording Setup

Every rollout is recorded by four synchronized cameras: one egocentric view and three exocentric views. Figure 12 shows the same instant of the same rollout from all four cameras, one row per task. The three exocentric cameras are used for recording only and were never used in training. Scoring is done live by the evaluation team of App. E.4, not from any recording, so the multi-view setup does not affect any number in Table 1. We keep this setup for post-hoc failure analysis, because different cameras expose different failure modes. For Pick & Place, Exo 3 sees the screw clear the hand's silhouette against a plain wall, while the top-down cameras see it only against cluttered trays.