

ESPADA: Execution Speedup via Semantics Aware Demonstration Data Downsampling for Imitation Learning

anonymous authors

Abstract—Behavior-cloning based visuomotor policies enable precise manipulation but often inherit the slow, cautious tempo of human demonstrations, limiting practical deployment. However, prior studies on acceleration methods mainly rely on statistical or heuristic cues that ignore task semantics and can fail across diverse manipulation settings. We present ESPADA, a semantic and spatially aware framework that segments demonstrations using a VLM–LLM pipeline with 3D gripper–object relations, enabling aggressive downsampling only in non-critical segments while preserving precision-critical phases, without requiring extra data or architectural modifications, or any form of retraining. To scale from a single annotated episode to the full dataset, ESPADA propagates segment labels via Dynamic Time Warping (DTW) on dynamics-only features. Across both simulation and real-world experiments with ACT and DP baselines, ESPADA achieves approximately a 2x speedup while maintaining success rates, narrowing the gap between human demonstrations and efficient robot control.

I. INTRODUCTION

Imitation learning (IL) has emerged as a central paradigm in robot learning [1]–[7], offering a practical alternative to reinforcement learning by bypassing explicit reward design and costly online exploration. By leveraging expert demonstrations, IL enables robots to acquire manipulation skills in a data-efficient manner. While early applications were limited to simple pick-and-place tasks, recent advances have extended IL to long-horizon [3], contact-rich [8], and visually complex manipulations [1], [9]. Widely used policies such as Action Chunking Transformer (ACT) [1] and Diffusion Policy (DP) [2] illustrate this practicality and serve as strong baselines for imitation-based manipulation.

Despite these successes, deployments in IL often suffer from insufficient execution speed. Human demonstrators tend to act slowly and cautiously to ensure safety and maximize task success. Moreover, prior studies have intentionally adopted slow demonstrations due to three main factors: (i) camera frame-rate constraints, (ii) research that slower motions can improve training stability, and (iii) the anthropomorphism gap between human kinematics and robotic morphology [10]. In short, these factors collectively bias human operators toward conservative motions, producing trajectories that are far more temporally saturated than necessary, thereby causing learned policies to inherit this slow tempo at execution time [11].

Simply replaying demonstrations faster or uniformly subsampling observations can push trajectories out of distribution, inducing compounding error and degraded performance. In response, several methods have sought to improve execution efficiency; SAIL [12] leverages AWE [13] features with DBSCAN [14] clustering to identify coarse phases, and

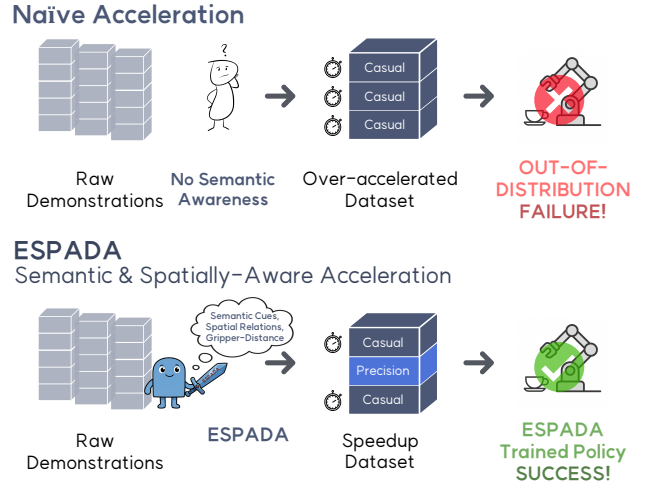


Fig. 1: **Naïve and heuristic-based acceleration breaks precision behavior in manipulation tasks.** Our model, ESPADA uses semantics and 3D spatial cues to preserve contact-critical phases while accelerating transit motions.

DemoSpeedup [15] identifies casual segments by estimating action-distribution entropy with a pre-trained proxy policy, treating high-entropy regions as safe to accelerate. However, these scenario-assumption based approaches rely on hand-crafted heuristics, and the narrow scenario space makes them fragile to even mild deviations.

SAIL [12], for instance, implicitly assumes that precision-critical behavior manifests as densely sampled regions in trajectory space and implements this assumption via clustering, but a density-based view of precision is intuitively valid only in highly restricted scenarios. On the other hand, DemoSpeedup [15] assumes that high action entropy signals accelerable segments, but entropy is not a reliable indicator of precision: (1) multimodal strategies arising from scenario variability (e.g., random object initialization) can yield high entropy despite strict precision demands, (2) repetitive path-fixed motions may have low entropy without actually requiring precision, causing accelerable segments to be sacrificed.

Fundamentally, both approaches rely on scenario-dependent assumptions and attempt to infer precision from motion statistics rather than task semantics, limiting their ability to distinguish accelerable from precision-critical phases and preventing them from scaling robustly in a task- and scenario-agnostic manner.

Accordingly, we introduce **ESPADA**, a semantic-driven trajectory segmentation framework that selectively acceler-

ates demonstrations without extra hardware, additional data, or additional policy trainings. Prior methods rely on heuristic motion statistics, such as density clusters or action entropy, to implicitly approximate precision. In contrast, ESPADA replaces these assumptions with explicit scene semantics and gripper-object 3D relations. These cues reveal the task intent (e.g., approach, align, adjust) and the actual interaction state between the gripper and the target object, enabling the system to determine exactly where acceleration is safe while preserving genuine precision-critical phases.

Concretely, we extract per-frame 3D coordinates of grippers and key objects using open-vocabulary segmentation [16], [17] and video-based depth estimation [18], [19], which we adopt instead of single-image estimators because depth can be computed offline and video models exploit temporal context across the entire sequence, yielding more stable and semantically coherent geometry. In addition, to remain compatible with the standard visuomotor imitation-learning formulation—where demonstrations rely solely on monocular onboard observations without auxiliary sensing—we derive all 3D cues from monocular video rather than requiring extra depth sensors.

These geometric cues, together with image observations, are summarized by a vision-language model [20] into semantic scene descriptions. We further convert all spatial and semantic observations into a compact language representation so that a large language model—currently the strongest general-purpose reasoning module—can perform segment classification in a token-efficient and structurally interpretable form. Next, the LLM reasons over these descriptions and trends in the gripper-object distance over time to classify segments into *casual* (aggressively accelerable) and *precision*. Finally, we accelerate the casual segments via *replicate-before-downsample* with *geometric consistency* [15], reducing temporal density while preserving task success.

Our contributions are three-fold: (i) The first semantic and 3D-relation-aware policy acceleration framework via demonstration downsampling without any additional sensor data or retraining. (ii) A scalable label transfer scheme that propagates segment labels from a single annotated episode to the rest of the dataset using banded DTW. (iii) Experimental validation in simulation and real-world settings, with up to a 3.6 \times execution speedup while maintaining or improving success rates.

II. RELATED WORK

Speeding up imitation learning execution: While modern visuomotor policies such as ACT [1] and DP [2] provide strong manipulation performance, they typically inherit the slow, human-paced timing of demonstrations. Uniform downsampling or increasing the control rate can speed up execution, but both risk pushing observations Out of Distribution (OOD) and amplifying errors. Recent work proposes learning when it is safe to compress time. DemoSpeedup estimates action-distribution entropy from a proxy policy and applies replicate-before-downsample with

geometric constraints, reporting up to $\sim 3\times$ acceleration without success loss [15]. In parallel, SAIL estimates “motion complexity” from waypoints and uses DBSCAN to segment trajectories; segments with lower complexity are accelerated while higher-complexity segments are preserved [12]. While effective, these prior methods rely on narrow, scenario-specific assumptions — often tied to clustering hyperparameters or entropy as a coarse proxy — which limits their robustness. Our approach replaces entropy/feature clustering with semantic reasoning, yielding segments that better align with manipulation intent.

Temporal segmentation and phase discovery: Classical phase-discovery pipelines often rely on fixed features and clustering (e.g., DBSCAN [14]) to recover phases from kinematics or vision [12], [13], sometimes assisted by motion primitives. These pipelines can be brittle across tasks and cameras because phase boundaries depend on feature scaling and neighborhood thresholds. Other approaches use latent structure learning for phase discovery [6], [7], but they still struggle to distinguish precision-critical contact phases from benign transits. ESPADA instead uses 3D gripper-object distance trends as grounded signals, deferring semantic interpretation to an large language model (LLM), which produces coherent manipulation chunks.

Vision-language for robotics: VLMs and LLM provide complementary capabilities: grounded perception from images and structured reasoning over text. This modularity has been explored for generalist robots [9], [21]–[23]. Instead of training bespoke video-understanding models, we adopt a VLM \rightarrow LLM pipeline: category-free segmentation (Grounded DINO + SAM [16], [24]), depth [18], [19], and semantic summaries (InternVL [20]), followed by LLM-based segmentation. The motivation for this modular design is that spatial relations—such as gripper-object geometry—must be explicitly surfaced as linguistic cues for downstream reasoning, which is difficult to guarantee with monolithic video-understanding models. By converting spatial structure into interpretable language tokens, the LLM can perform fine-grained temporal and semantic reasoning with more reliability and controllability. This design is auditable, improves as foundation models improve, and transfers to an online variant for fast/precise mode switching.

Positioning: ESPADA addresses several specific limitations of DemoSpeedup and SAIL. While those methods assume that high entropy or low complexity reliably indicate “casual” motion, ESPADA detects when that assumption breaks by consulting explicit relational and semantic cues, specifically gripper-object distance trends and scene semantics. Unlike entropy alone, our segmentation tends to produce more stable, coherent boundaries and avoids misclassifying fine, contact-critical motions as safe to downsample. Empirically, ESPADA produces fewer fragmented boundaries and more coherent motion chunks, simplifying per-segment compression factor selection and reducing reliance on delicate clustering hyperparameters.

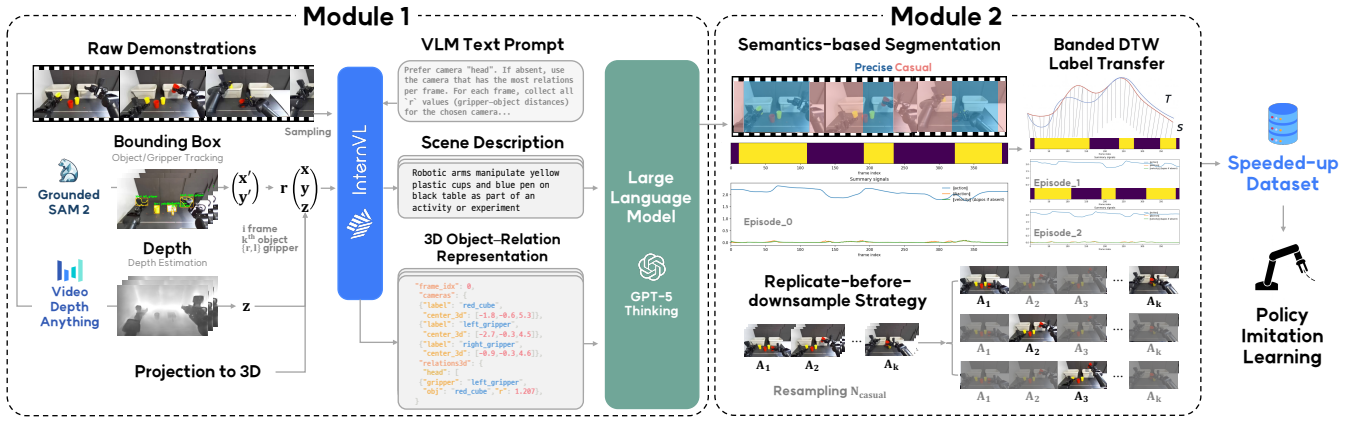


Fig. 2: **Overview of ESPADA.** We use Grounded-SAM2 and Video Depth Anything (VDA) to extract 3D object-gripper relations, summarize the episode with a VLM, and segment trajectories with an LLM into *precision* and *casual* spans. Segment-wise downsampling is then applied with replicate-before-downsample and geometric consistency, producing faster yet safe demonstrations for imitation learning. To reduce annotation cost, we annotate only episode 0 via the VLM→LLM pipeline, and propagate its labels to other episodes with *banded DTW label transfer*, which aligns action sequences under temporal variation while refining boundaries.

III. PROBLEM SETUP

We consider a dataset of robot manipulation demonstrations $\mathcal{D} = \{(o_t, a_t)\}_{t=1}^T$, where o_t are observations (RGB images, proprioception) and a_t are low-level actions (joint position commands). Demonstrations are collected at control frequencies $f_{\text{ctrl}} \in [30, 50]$ Hz, producing temporally dense trajectories. Policies such as ACT [1] and DP [2] predict fixed-horizon action chunks $A_t = \{a_t, \dots, a_{t+K-1}\}$ from recent observations.

A core issue is that human demonstrations are performed slowly and cautiously, yielding oversampled sequences. Uniformly downsampling often pushes trajectories out-of-distribution, because aggressive temporal thinning alters the local action–state transitions seen during training, introduces temporal aliasing in contact-rich or high-curvature segments, and disrupts the smoothness assumptions under which behavior-cloned policies generalize. Our goal is to accelerate demonstrations offline by selectively reducing temporal density in *casual phases* while applying only mild reduction in *precision-critical phases*, without modifying the runtime control loop or the policy architecture.

Formally, we segment each trajectory as $\mathcal{S} = \{(s_i, e_i, y_i)\}_{i=1}^M$, with $y_i \in \{\text{casual}, \text{precision}\}$. We then transform \mathcal{D} by

$$\mathcal{T}(\mathcal{D}, \mathcal{S}, N) = \bigcup_{i=1}^M \left\{ \text{RBD}(\mathcal{D}[s_i : e_i], N_{y_i}) \right\}$$

where $y_i \in \{\text{precision}, \text{casual}\}$, and RBD denotes downsampling with *replicate-before-downsample* [15], ensuring that all original frames are preserved across replicas. Here, *casual* indicates segments that can be safely downsampled without compromising task fidelity, while *precision* denotes precision-critical spans that are retained at near full resolution, with only minimal acceleration applied when safe. For stability, we enforce *geometric*

consistency [15] by adjusting accelerated chunk horizons K' so that the spatial displacement $\sum_{k=0}^{K'-1} \|\Delta \mathbf{x}_{t+k}\|$ matches that of the original horizon K . N_{y_i} denotes the number of replicas in RBD, determined by the maximum acceleration ratio.

IV. METHOD

Our pipeline converts raw demonstrations into *semantically and spatially informed* segments that can be selectively accelerated, then constructs an acceleration-aware training set via replicate-before-downsample (RBD) with geometric consistency. Figure 2 provides an overview.

A. Context- and Spatial-Aware Segmentation via VLM → LLM

a) Object tracking with interactive keyframe seeding.:

First, we obtain open-vocabulary tracks from demonstration videos using Grounded-SAM2 [16], [17]. In addition to text prompts, users can provide sparse keyframe annotations (boxes or point-groups) via a lightweight UI. We maintain a label↔id mapping across keyframes and perform IoU-based association to propagate user labels to SAM2 track IDs. During propagation, we use a keep-alive strategy (bbox carry-over for short outages) and periodic re-detection with Grounding DINO, reconnecting lost tracks via a score that mixes IoU and color-histogram similarity. This reduces fragmentations and preserves object identity across occlusions.

To bootstrap object grounding, we first sample ~ 10 representative frames from episode 0 and feed them into a InternVL 3.5 [20] to obtain a compact natural-language description of the overall task. For the same frames, we apply Grounding DINO v2 to detect and segment task-relevant entities such as `left_gripper`, `right_gripper`, and target objects (e.g., `yellow_cup`). If bounding box predictions fail for some frames, we allow lightweight manual correction (bounding box only) through the UI. The corrected boxes

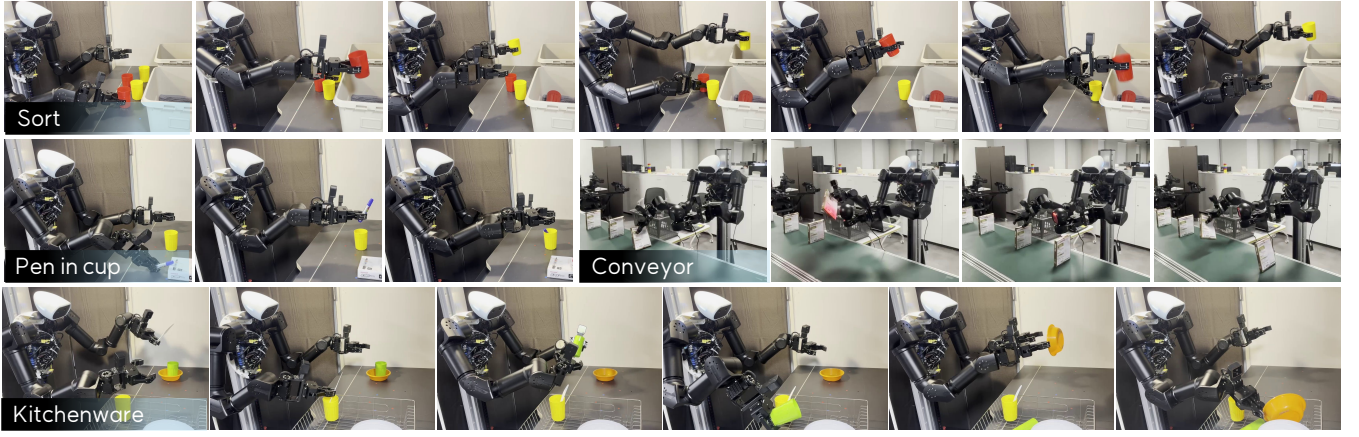


Fig. 3: **Real-world evaluation of ESPADA on the AI Worker robot across four representative manipulation tasks.** (i) Sort – classifying colored objects into bins, (ii) Pen in cup – placing a pen into a cup, (iii) Conveyor – transferring curry into a basket along a moving belt, and (iv) Kitchenware – handling bowls and cups.

serve as anchors for SAM2, which then propagates object masks and bounding boxes consistently across the entire episode. This hybrid strategy (automatic detection + sparse manual fallback + SAM2 propagation) ensures that every frame obtains reliable per-object segmentation, even under occlusion or detector failure.

b) Depth estimation and 3D back-projection.: We estimate per-frame depths with VDA/DA2 [18], [19] (metric or relative; optionally scaled by a factor z_{scale}). As we obtained the pixel coordinates (u, v) of each object of interest in the previous step, given the corresponding depth Z , we can recover its 3D position in the camera coordinate frame via standard back-projection:

$$\mathbf{p} = ZK^{-1}[u, v, 1]^T, \quad (1)$$

This yields a `center_3d` for each tracked mask. We then compute frame-wise gripper-object distances,

$$r_t(g, o) = \|\mathbf{p}_t^{(g)} - \mathbf{p}_t^{(o)}\|_2, \quad (2)$$

for $g \in \{\text{gripper_left}, \text{gripper_right}\}$ and task-relevant objects o . For multi-view sequences, we build per-camera `relations_3d` from the set of $r_t(g, o)$ values, and prefer the head camera if present; otherwise we select the camera with the most valid relations at a frame. We rely on *temporal trends* in r_t rather than absolute scale, avoiding the need for extrinsics.

c) LLM-Based Segmentation Conditioned on VLM Summaries.: From the sampled frames (typically 4–8) and their structured 3D cues, we query a VLM (InternVL [20]) for a strict-JSON, chronologically ordered episode summary. We then attach this VLM-produced summary as a task descriptor to the LLM prompt. Using this VLM-produced descriptor, we perform LLM-based segmentation as follows. The LLM receives: (i) a JSONL stream with frame-wise `center_3d` and `relations_3d` for the *full* episode, and (ii) the *VLM summary descriptor*. It outputs non-overlapping, inclusive index ranges labeled `precision` or `casual`. We encode *policy hints* to favor robust, human-like chunks:

- **Intent criteria.** Sustained near-contact plateaus and low-variance micro-adjustments \Rightarrow precision; long approach/retreat or persistent far separation \Rightarrow casual.
- **Stability.** Minimum segment length $L_{\min}=8$; merge same-label segments across gaps shorter than $G_{\min}=5$; require ≥ 3 consecutive frames to switch labels (hysteresis); ignore micro-oscillations shorter than $L_{\text{micro}}=6$.
- **Parsimony.** Prefer 3–4 segments unless strong evidence suggests otherwise.

Because the model may leave small gaps when confidence is low, we run a deterministic *coverage completion* pass: fill gaps by extending the nearest high-confidence neighbor that best matches the local r_t trend, then re-apply the stability rules. The final set $\mathcal{S} = \{(s_i, e_i, y_i)\}_{i=1}^M$ provides *full frame coverage* with $y_i \in \{\text{precision}, \text{casual}\}$ and per-segment confidence.

Finally, to respect LLM context limits for long demonstrations, we apply token-budgeted sampling and JSON slimming. Demonstrations often have thousands of frames, easily exceeding LLM context limits. We therefore compute the maximum feasible sample count K by binary search over the measured per-frame JSON length and select K *evenly spaced* indices, ensuring trajectory-wide coverage under a fixed character budget. We further compact prompts by float rounding and whitespace-free JSON serialization, reducing token overhead by ~ 30 –40% without changing semantics.

B. Banded DTW Label Transfer from Episode-0

For datasets where only episode 0 is labeled, we propagate its segment labels (`precision` / `casual`) to the remaining episodes via *banded* Dynamic Time Warping (DTW).

a) Proprioceptive DTW Alignment.: From each episode we build a per-frame feature vector using only proprioception and actions. Concretely, we concatenate z-scored features to

form $\phi_t \in R^D$:

$$\phi_t = [a_t, \Delta a_t, v_t, \Delta v_t, \|a_t\|, \|v_t\|, \|\Delta a_t\|, \|\Delta v_t\|, \angle(a_t, a_t + \Delta a_t), \angle(v_t, v_t + \Delta v_t)]. \quad (3)$$

where a_t are actions, q_t are joint positions, v_t are joint velocities if available (otherwise we use Δq_t as a proxy), and $\angle(\cdot, \cdot)$ is the angle between successive vectors. Given episode 0 features $X_0 \in R^{T_0 \times D}$ and target features $X_k \in R^{T_k \times D}$ for episode k , we run DTW with a Sakoe–Chiba band of half-width $b = \lfloor \rho \cdot \max(T_0, T_k) \rfloor$ with $\rho \in [0.05, 0.10]$ (default $\rho=0.08$). This yields an alignment path $\mathcal{P} \subset [1, T_0] \times [1, T_k]$. We convert it into a monotone index map $m : \{1, \dots, T_0\} \rightarrow \{1, \dots, T_k\}$ by averaging all matched target indices per source frame and enforcing non-decreasingness.

b) Segment-wise Label Transfer and Refinement. For each episode-0 labeled segment $\mathcal{S}_0 = \{(s_i, e_i, y_i)\}_{i=1}^M$ with label $y_i \in \{\text{precision}, \text{casual}\}$, we obtain the target span $\mathcal{S}_k = \{(m(s_i), m(e_i), y_i)\}_{i=1}^M$ and snap both ends within a local window of $\pm W$ frames (default $W=12$) by minimizing the ℓ_2 distance between short mean-pooled feature summaries. Mapped segments are sorted and trimmed to remove overlaps while preserving order. If a path break occurs, we drop only the affected segment. Any uncovered frames default to precise when expanded to per-frame labels. The banded DTW runtime is $\mathcal{O}(\max(T_0, T_k) \cdot b)$, i.e., near-linear in sequence length. With 50 Hz episodes (~ 500 – $2k$ frames), transfers run quickly on CPU and require no proxy models.

C. Segment-wise Downsampling and Dataset Compilation

Given the final segmentation \mathcal{S} , we construct an acceleration-aware dataset by applying replicate-before-downsample with a larger downsampling factor for casual spans and a smaller downsampling factor for precision spans.

a) Replicate-before-downsample. To maintain full state coverage under temporal compression, we adopt a replicate-before-downsample strategy [15]. For a segment $[s, e]$ and downsampling factor N , we create N replicas with offsets $m \in \{0, \dots, N-1\}$ and retain frames $\{t \in [s, e] \mid (t-s) \bmod N = m\}$. Taking the union across m recovers the original support, thereby preserving full state diversity in the downsampled dataset and preventing loss of observation coverage during model training.

b) Geometric Consistency for Chunked Policies.

Temporal acceleration alters the per-chunk spatial displacement, undermining the horizon K that the policy has been optimized to perform best at. To maintain geometric fidelity under accelerated demonstrations, we adopt the geometry-consistent downsampling scheme [15] and rescale the effective chunk horizon K' so that its spatial displacement remains consistent with the original:

$$\sum_{k=0}^{K'-1} \|\Delta \mathbf{x}_{t+k}\| \approx \sum_{k=0}^{K-1} \|\Delta \mathbf{x}_{t+k}\|, \quad (4)$$

where \mathbf{x}_t denotes the end-effector pose. In practice, $K' \approx \frac{1}{2}K$ performs well and approximately satisfies Eq. (4) across tasks.

c) Gripper Event Precision Forcing. We apply gripper event precision forcing method to safeguard contact-rich phases from being over-accelerated. For each trajectory, we detect gripper movements by checking the change in the normalized gripper command g_t and mark a frame as a candidate event if $|g_{t+4} - g_t| \geq 0.03$. All marked frames are then clustered along the temporal axis using DBSCAN [14]. For each cluster, we take the minimum and maximum frame indices, pad them by two frames on both sides, and override the corresponding window to be precision on top of the base LLM segmentation results.

V. EXPERIMENTS

Our experimental evaluation is guided by the following research questions:

- **RQ1.** Does ESPADA achieve a higher success rate across diverse manipulation tasks, even under more aggressive acceleration settings, compared to baselines?
- **RQ2.** How accurately does ESPADA distinguish precision-critical from casual segments compared to entropy-based segmentation methods?
- **RQ3.** What are the respective roles of the 3D gripper–object distance r_t and VLM-generated scene descriptions in improving segmentation quality?

A. Setup

We evaluate our approach in both simulation and real-world settings using ACT and DP [1], [2] as the baseline policy architectures, and compare our accelerated model against policies trained on the original dataset and those using the entropy-based acceleration method DemoSpeedup [15] under each architecture.

Simulation. In Aloha [4], we evaluate two representative manipulation tasks—Transfer Cube and Insertion—each provided with 50 expert demonstrations at 50 Hz. Policies are trained from single head-camera observations. Experiments were conducted with precision/casual acceleration factors of (2x, 4x). In BiGym [25], we evaluate 7 long-horizon manipulation tasks that involve target reaching and articulated object interaction in home-like environments. Policies are trained with different numbers of demonstrations per task, while failed episodes are filtered out.

Real-world. Experiments are conducted on the ROBOTIS AI-Worker, a dual-gripper humanoid robot equipped with two wrist-mounted cameras and a head-mounted camera. We evaluate four representative tasks—*Sort* (bin sorting), *Pen in Cup* (insertion), *Kitchenware* (bowl and cup handling), and *Conveyor* (dynamic transfer)—as shown in Fig. 3, measuring both throughput and episode length across models. All policies follows the baseline-matched hyperparameters [15] for both training and inference, and the accelerated segments use a chunk horizon of roughly half the original. DP exhibited limited robustness to large out-of-distribution (OOD) deviations during preliminary experiments. To avoid conflating this effect with the impact of temporal acceleration, we reset the initial robot pose to lie within the training-time distribution for all tasks except *Conveyor*.

Metrics. We evaluated whether time efficiency could be improved without compromising task success. We report the *task completion success rate* and the *average episode execution length*, where task failure is defined as the inability to proceed within 10 seconds in real-world experiments.

B. Simulation Results

In Aloha, As shown in Table II, While naïve 2× acceleration lowers success rates, our method even improves them while achieving up to 2.64× speedup over the original. Relative to DemoSpeedup, it matches performance on all *Insertion* while demonstrating a similar level of acceleration, and achieves the highest success on *Transfer Cube*(ACT) while being slightly less aggressive in shortening episodes.

High Segmentation Quality Under Random Scenario. Random initialization of object position in the Aloha environment increases entropy during the approach–grasp phase, leading the entropy-based baseline to mislabel this interaction-critical region as a casual segment. To evaluate segmentation quality, we compare segmentation outputs against ground-truth manually annotated by human evaluators using explicit physical-interaction criteria. Against this reference, our method achieves higher IoUs—0.1989 vs. 0.1745 for insertion and 0.2649 vs. 0.2013 for transfer cube—demonstrating robustness to initialization-induced variability. ALOHA Sim also reports subtask-level success metrics, and in the initial interaction-detection subtask, which is particularly sensitive to randomness in object placement, our method attains 91% success compared to 87% for the entropy-based baseline, further indicating the stability of semantic grounding in early-phase boundary identification.

Long-Horizon Speedup and Sensitivity to Unstable Visual Scenes. In BiGym, our method achieves significantly higher success rates than the simple 2× baseline (ACT: 66%→73%, DP: 47%→60%) while maintaining performance comparable to the original 1× policy and providing up to 2.3× acceleration. Interestingly, acceleration and task success were not inversely related; faster execution often improved success by reducing compounding errors and preventing drift into OOD states. While our approach performs on par with DemoSpeedup in most tasks, we still observe failures in some cases, likely due to unstable visual observations—often outside the object scene as the robot moves—which undermine gripper–object recognition and VLM semantic grounding. We leave it to future work to improve semantic grounding through more stable viewpoints and richer multimodal signals such as joint states and haptics.

C. Real-world Results

As shown in Table III, ESPADA achieves the highest overall success rates while providing strong acceleration across all tasks. Under the 2×/4× setting (precision range 2×, casual range 4× acceleration), ACT+Ours achieves 90.0% success at 2.21× speedup, whereas DemoSpeedup drops to 45.0% despite achieving a marginally higher speedup by aggressively classifying many spans as casual. This over-acceleration is most evident in the *Conveyor* task, where

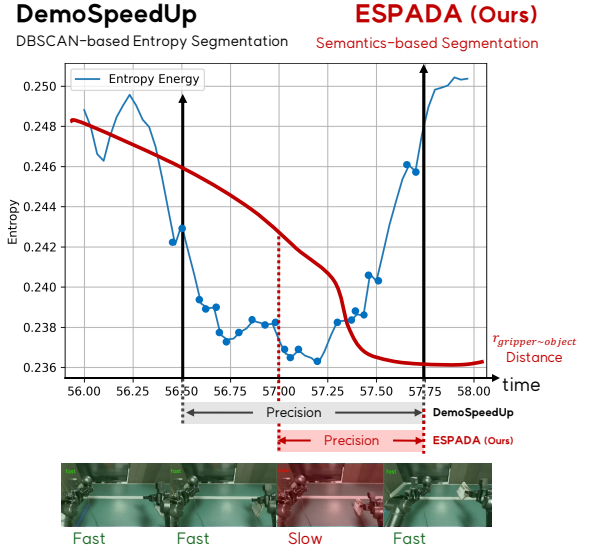


Fig. 4: **Precision-phase estimation in the conveyor scenario based on low entropy (DemoSpeedup, black regions) versus semantics (Ours, red regions).** In repetitive and relatively simple segments such as grasping curry on the conveyor, DemoSpeedup misclassifies them as precision-critical due to low action entropy. In contrast, our semantic analysis correctly identifies these spans as accelerable.

DemoSpeedup collapses to 1/20 success while ESPADA maintains 18–19/20. A similar trend holds for DP: ESPADA attains both the best success rate (85.4%) and the largest speedup (2.41×), outperforming DP+DemoSpeedup (79.6%).

Casual-exploiting Segmentation. By combining temporal trends in the gripper–object distance with VLM-generated scene descriptions—and leveraging the reasoning capability of an LLM to interpret these cues—ESPADA reliably identifies genuine precision phases, such as near-contact adjustments, while aggressively compressing spans that are truly casual. In contrast, entropy-based segmentation implicitly treats low action entropy as a proxy for precision. This assumption fails in repetitive motions: entropy often remains low even when no fine control is required, causing DemoSpeedup to systematically overestimate precision-critical spans. As shown in Fig. 4 for the *Conveyor* task, this misclassification marks large portions of repetitive scooping as non-accelerable, restricting potential speed gains and obscuring accelerable casual segments that ESPADA correctly recovers. Low entropy, in other words, does not necessarily imply high precision demands.

Quantitatively, the same pattern appears in the 1/3-acceleration setting, where the number of trials yields stable statistics. ESPADA achieves slightly shorter episode lengths not by compressing true precision phases, but by avoiding DemoSpeedup’s overextension of low-entropy repetitive segments. Across tasks, ESPADA consistently preserves success-critical precision phases without compromising success rate, while more accurately identifying accelerable ca-

TABLE I: BiGym Simulation Results

Method	Sandwich Remove		More Plate		Load Cups		Put Cups	
	success rate(\uparrow)	episode len(\downarrow)	success rate(\uparrow)	episode len(\downarrow)	success rate(\uparrow)	episode len(\downarrow)	success rate(\uparrow)	episode len(\downarrow)
ACT	53%	368	54%	157	61%	319	61%	288
ACT-2x	46%	193	46%	119	50%	195	54%	141
ACT+DemoSpeedup	77%	156	53%	91	59%	176	62%	132
ACT+Ours	80%	176	24%	91	54%	173	60%	149
DP	52%	352	52%	170	15%	419	12%	386
DP-2x	51%	247	41%	125	11%	177	7%	243
DP+DemoSpeedup	54%	217	49%	113	38%	171	21%	205
DP+Ours	46%	200	40%	79	34%	162	38%	218

Method	Saucepan to Hob		Drawers Close		Cupboard Open		Averaged	
	success rate(\uparrow)	episode len(\downarrow)	success rate(\uparrow)	episode len(\downarrow)	success rate(\uparrow)	episode len(\downarrow)	success rate(\uparrow)	speed-up(\uparrow)
ACT	86%	383	100%	119	100%	146	74%	1.0 \times
ACT-2x	81%	224	87%	84	96%	103	66%	1.7 \times
ACT+DemoSpeedup	92%	163	100%	63	100%	81	78%	2.1 \times
ACT+Ours	94%	148	100%	56	100%	81	73%	2.3 \times
DP	79%	324	96%	114	100%	181	58%	1.0 \times
DP-2x	41%	242	81%	65	94%	161	47%	1.5 \times
DP+DemoSpeedup	79%	169	89%	59	100%	103	61%	1.9 \times
DP+Ours	76%	148	88%	56	100%	116	60%	2.0\times

TABLE II: Aloha-Sim Simulation Results

Method	Insertion		Transfer Cube	
	success rate(\uparrow)	episode len(\downarrow)	success rate(\uparrow)	episode len(\downarrow)
ACT	21%	452	72%	291
ACT-2x	13%	238	70%	162
ACT+DemoSpeedup _(repro)	28%	166	66%	127
ACT+Ours	28%	171	72%	141
DP	16%	431	66%	281
DP-2x	12%	245	61%	146
DP+DemoSpeedup _(repro)	26%	173	64%	137
DP+Ours	26%	193	58%	121

sual spans.

Precision-preserving Segmentation. In *Kitchenware* (ACT, 2 \times /4 \times), ESPADA achieves **16/20** successes versus **1/20** for DemoSpeedup, which shows ESPADA reliably maintains precision-critical phases even under high acceleration. In contact-rich manipulation, near-contact spans must not be down-sampled; ESPADA preserves this precision-critical interaction, whereas DemoSpeedup down-samples the delicate cup-grasp phase too aggressively, leading to only 1/20 success. The gripper-object-distance trend feeded into LLM allow it to infer phase intent (approach \rightarrow align \rightarrow close), while conservatively gating gripper events as well—thereby retaining precision spans and avoiding to compress precise motion.

Robustness in Dynamic Scenario. *Conveyor* task shows that manipulation task in dynamic scene exposes fundamental limitations of entropy-based acceleration. When the target object(curry) first enters the camera view, the system must hold the arm still and wait for the correct picking configuration. However, action entropy is naturally high during this early transient, causing DemoSpeedup to repeatedly misclassify this span as casual and trigger arm descent earlier than intended. Consequently, the joint state collapses into an unrecoverable configuration under ACT’s strong joint-state-conditioned action chunking tendency, leading to failures.

In contrast, ESPADA explicitly identifies this waiting phase as precision by leveraging semantic cues from VLM descriptions together with the temporal trend of the gripper-object distance. The model correctly holds the arm

still until the object reaches the appropriate pickup zone, preventing early descent and ensuring stable execution even under irregular conveyor timing.

D. Ablations

We ablate the effect of gripper-object distance r and the VLM scene description using four variants: w/o r , w/o description, w/o both, and our full model (Table IV). We report IoU and the predicted number of segments with respect to the ground-truth segmentation.

For **Insertion**, removing r collapses IoU (0.5166 \rightarrow 0.0224), indicating that r is essential for alignment-sensitive interactions. For **Transfer Cube**, dropping the description sharply reduces IoU (0.3064 \rightarrow 0.0584), suggesting that textual cues help disambiguate phases with similar geometry. All variants recover the correct number of segments, but only our full model achieves tight temporal alignment.

Overall, this ablation confirms that the r value encodes precise temporal segmentation cues, while scene description provides semantic grounding, and both are necessary for high-fidelity alignment.

VI. CONCLUSION

We presented ESPADA (Execution Speedup via Spatially Aware Demonstration Data Downsampling), a semantic segmentation framework that accelerates demonstrations without requiring additional data, hardware, or policy retraining. By exploiting scene semantics and gripper-object spatial relations, ESPADA distinguishes acceleration from precision-critical segments, producing motion-aligned chunks and reducing temporal redundancy via replicate-before-downsample [15] with geometric consistency. Integrated with ACT and DP, ESPADA achieves natural motion chunking, preserves task success, and generalizes across both simulation and real hardware.

Limitations. ESPADA still faces challenges: inaccurate masks or object tracking may distort spatial relations, monocular depth estimation introduces noise, and further validation is needed for large-scale deployment. Addressing these issues

TABLE III: Real-world Results.

Method	Pen in Cup		Sort		Kitchenware		Conveyor		Averaged	
	success rate(↑)	episode len(↓)	success rate(↑)	episode len(↓)	success rate(↑)	episode len(↓)	success rate(↑)	episode len(↓)	success rate(↑)	speed-up(↑)
ACT	29/30	18.67	27/30	37.52	8/20	38.68	13/20	9.89	72.9%	1.0x
ACT+DemoSpeedup 1/3	29/30	15.52	29/30	29.29	10/20	25.62	4/20	7.55	65.8%	1.34x
ACT+DemoSpeedup 2/4	27/30	5.36	24/30	13.23	1/20	17.39	1/20	4.49	45.0%	2.59x
ACT+Ours 1/3	29/30	15.32	29/30	29.32	10/20	22.76	19/20	7.36	84.6%	1.40x
ACT+Ours 2/4	29/30	6.57	28/30	15.56	16/20	20.72	18/20	4.51	90.0%	2.21x
DP	11/15	21.55	10/15	48.29	0/15	x	0/20	x	35.0%	1.0x
DP+DemoSpeedup 2/4	15/15	8.66	13/15	23.58	10/15	27.50	13/20	6.15	79.6%	2.12x
DP+Ours 2/4	15/15	5.83	15/15	21.54	10/15	23.15	15/20	7.41	85.4%	2.41x

TABLE IV: Ablation on IoU and number of segments.

Method	Insertion		Transfer Cube	
	IoU	#Seg.	IoU	#Seg.
w/o r	0.0224	3/3	0.2791	1/2
w/o Desc.	0.1024	3/3	0.0584	3/2
w/o r , Desc.	0.1111	3/3	0.0693	3/2
Ours	0.5166	3/3	0.3064	2/2

will be crucial for advancing ESPADA as a reliable and general framework for safe and efficient policy acceleration.

REFERENCES

- [1] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, “Learning fine-grained bimanual manipulation with low-cost hardware,” *arXiv preprint arXiv:2304.13705*, 2023.
- [2] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, “Diffusion policy: Visuomotor policy learning via action diffusion,” *The International Journal of Robotics Research*, p. 02783649241273668, 2023.
- [3] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter *et al.*, “ π 0: A vision-language-action flow model for general robot control. corr. abs/2410.24164, 2024. doi: 10.48550/arXiv:2410.24164,” *arXiv preprint arXiv:2410.24164*.
- [4] T. Z. Zhao, J. Tompson, D. Driess, P. Florence, K. Ghasemipour, C. Finn, and A. Wahid, “Aloha unleashed: A simple recipe for robot dexterity,” *arXiv preprint arXiv:2410.13126*, 2024.
- [5] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, “3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations,” *arXiv preprint arXiv:2403.03954*, 2024.
- [6] A. Mandlekar *et al.*, “robomimic: A framework for robot learning from demonstration,” *Conference on Robot Learning (CoRL)*, 2021.
- [7] F. Ebert *et al.*, “Bridge data: A large-scale dataset for robotic imitation learning,” *Conference on Robot Learning (CoRL)*, 2022.
- [8] J. J. Liu, Y. Li, K. Shaw, T. Tao, R. Salakhutdinov, and D. Pathak, “Factr: Force-attending curriculum training for contact-rich policy learning,” *arXiv preprint arXiv:2502.17432v1*, 2025.
- [9] A. Brohan *et al.*, “Open-x embodiment: Extending rt-x to diverse robots,” *arXiv preprint arXiv:2306.08592*, 2023.
- [10] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song, “Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots,” in *Robotics: Science and Systems*, 2024.
- [11] J. Xie, Z. Wang, J. Tan, H. Lin, and X. Ma, “Subconscious robotic imitation learning,” *arXiv preprint arXiv:2412.20368*, 2024.
- [12] N. Ranawaka Arachchige, Z. Chen, W. Jung, W. C. Shin, R. Bansal, P. Barroso, Y. H. He, Y. C. Lin, B. Joffe, S. Kousik *et al.*, “Sail: Faster-than-demonstration execution of imitation learning policies,” *arXiv e-prints*, pp. arXiv–2506, 2025.
- [13] L. X. Shi, A. Sharma, T. Z. Zhao, and C. Finn, “Waypoint-based imitation learning for robotic manipulation,” *arXiv preprint arXiv:2307.14326*, 2023.
- [14] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *kdd*, vol. 96, no. 34, 1996, pp. 226–231.
- [15] L. Guo, Z. Xue, Z. Xu, and H. Xu, “Demospeedup: Accelerating visuomotor policies via entropy-guided demonstration acceleration,” *arXiv preprint arXiv:2506.05064*, 2025.
- [16] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu *et al.*, “Grounding dino: Marrying dino with grounded pre-training for open-set object detection,” *arXiv preprint arXiv:2303.05499*, 2023.
- [17] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson *et al.*, “Sam 2: Segment anything in images and videos,” *arXiv preprint arXiv:2408.00714*, 2024.
- [18] S. Chen, H. Guo, S. Zhu, F. Zhang, Z. Huang, J. Feng, and B. Kang, “Video depth anything: Consistent depth estimation for super-long videos,” in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 22 831–22 840.
- [19] Z. Yang *et al.*, “Depth anything v2,” *arXiv preprint arXiv:2406.09414*, 2024.
- [20] W. Wang, Z. Gao, L. Gu, H. Pu, L. Cui, X. Wei, Z. Liu, L. Jing, S. Ye, J. Shao *et al.*, “Internvl3. 5: Advancing open-source multi-modal models in versatility, reasoning, and efficiency,” *arXiv preprint arXiv:2508.18265*, 2025.
- [21] A. Brohan *et al.*, “Rt-1: Robotics transformer for real-world control at scale,” *Robotics: Science and Systems (RSS)*, 2023.
- [22] M. Ahn *et al.*, “Do as i can, not as i say: Grounding language in robotic affordances,” *Robotics: Science and Systems (RSS)*, 2022.
- [23] D. Driess *et al.*, “Palm-e: An embodied multimodal language model,” *International Conference on Learning Representations (ICLR)*, 2023.
- [24] A. Kirillov *et al.*, “Segment anything,” *International Conference on Computer Vision (ICCV)*, 2023.
- [25] N. Chernyadev, N. Backshall, X. Ma, Y. Lu, Y. Seo, and S. James, “Bigym: A demo-driven mobile bi-manual manipulation benchmark,” *arXiv preprint arXiv:2407.07788*, 2024.