

TSD: A Physics-Inspired Trajectory Saliency Detector for Efficient Imitation Learning

Abstract—For imitation learning in robotic manipulation, high data collection costs result in the scarcity of high quality data. In this paper, we leverage the inherent heterogeneity of trajectories to address this challenge. Based on our observations of manipulation tasks, we categorize motions into transitional, precise, and agile types, defining the latter two as trajectory saliency due to their criticality to task success in contrast to the prevalent but less relevant transitional motions. Therefore, we propose the Trajectory Saliency Detector (TSD), a training-free and plug-and-play framework to identify trajectory saliency. TSD employs two physically-grounded metrics: spatial entropy to capture fine-grained manipulation and centripetal acceleration to detect agile maneuvering. We further leverage TSD to develop a dataset compression method that reduces training costs and a dataset expansion strategy that improves data collection efficiency. Extensive experiments in both simulation and real-world settings demonstrate that models trained on TSD-condensed datasets achieve comparable or even superior performance with 25% less data on average. These results validate the effectiveness of our dataset compression and expansion strategies, thereby confirming the utility of TSD. Consequently, TSD offers a scalable and cost-effective pathway to synthesize information-dense datasets for efficient robot learning.

Project page: <https://anonymous.4open.science/w/trajectory-saliency-detector/>

I. INTRODUCTION

The success of Scaling Laws in Natural Language Processing and Computer Vision has catalyzed a shift toward Vision-Language-Action (VLA) models in robotics [1–3]. However, unlike internet-scale textual data, high-quality robotic demonstrations remain a scarce resource, constrained by the costly and labor-intensive nature of teleoperation and kinesthetic teaching. This scarcity creates a significant data bottleneck [4, 5], where enhancing data collection efficiency while simultaneously ensuring data quality represents a critical research issue in robot learning [6].

Empirically, trajectory segments within demonstrations exhibit pronounced heterogeneity [7]. Critical segments capture the essential task logic, whereas transitional motions are often characterized by high levels of redundancy. For example, in a book-retrieval task, the initial reaching phase from the starting position to the shelf is relatively inconsequential. The critical bottleneck lies in the ability to grasp the book accurately and make minor adjustments. However, current data collection paradigms typically leads to excessive human labor due to prioritizing end-to-end trajectories. Furthermore, existing training paradigms in imitation learning typically treat all data points equally, regardless of the inherent redundancy, thereby failing to prioritize learning on saliency

trajectory segments [8–10]. The model exhausts its capacity fitting inconsequential segments, thereby diluting its focus on task-critical phases.

Therefore, leveraging the heterogeneity of data to concurrently enhance acquisition and training efficiency remains a critical challenge. Existing efforts often rely on simulation to generate massive demonstrations for data augmentation [11–14]. While this improves collection throughput, it inevitably inflates training costs. Conversely, current data compression methods typically employ auxiliary models as discriminators to filter large datasets [15–17], which introduces additional training requirements and time consumption. Thus, we propose a plug-and-play detector that leverages data heterogeneity to both guide dataset expansion—thereby reducing collection costs—and achieve informative dataset compression.

To address this, we introduce the Trajectory Saliency Detector (TSD). Our framework formalizes the intuition that fine-grained manipulation and agile maneuvering carry rich informational value. Fine-grained segments involve intricate object interactions where marginal deviations lead to failure [9], while maneuvering segments contain agile obstacle avoidance and dynamic adjustment under environmental constraints.

The core of TSD lies in two physically-grounded metrics: **Spatial Entropy** and **Centripetal Acceleration**. We observe that human demonstrators exhibit high variability in low-precision segments but naturally converge toward consistent paths as precision demands increase. Thus, the spatial distribution density of trajectory points serves as a reliable proxy for precision manipulation [19]. To capture agile maneuvering, we utilize centripetal acceleration to quantitatively distinguish purposeful, high-speed adjustments from ordinary linear motions.

TSD provides two benefits for efficient learning. First, during data collection, it identifies task-critical segments from a minimal initial set, guiding human collectors to prioritize high-value trajectory segments. Second, for large-scale datasets, TSD compress dataset by filtering non-essential fragments, significantly reducing computational overhead and preserving quality.

Through extensive validation in Robosuite [20] and real-world manipulation tasks, we demonstrate that TSD accurately identifies saliency segments. Models trained on TSD-condensed datasets achieve comparable or superior performance—using less data and shorter acquisition times—than those trained on full datasets. Detailed studies further confirm the robustness across varying data scales and the effectiveness of both salient segments type.

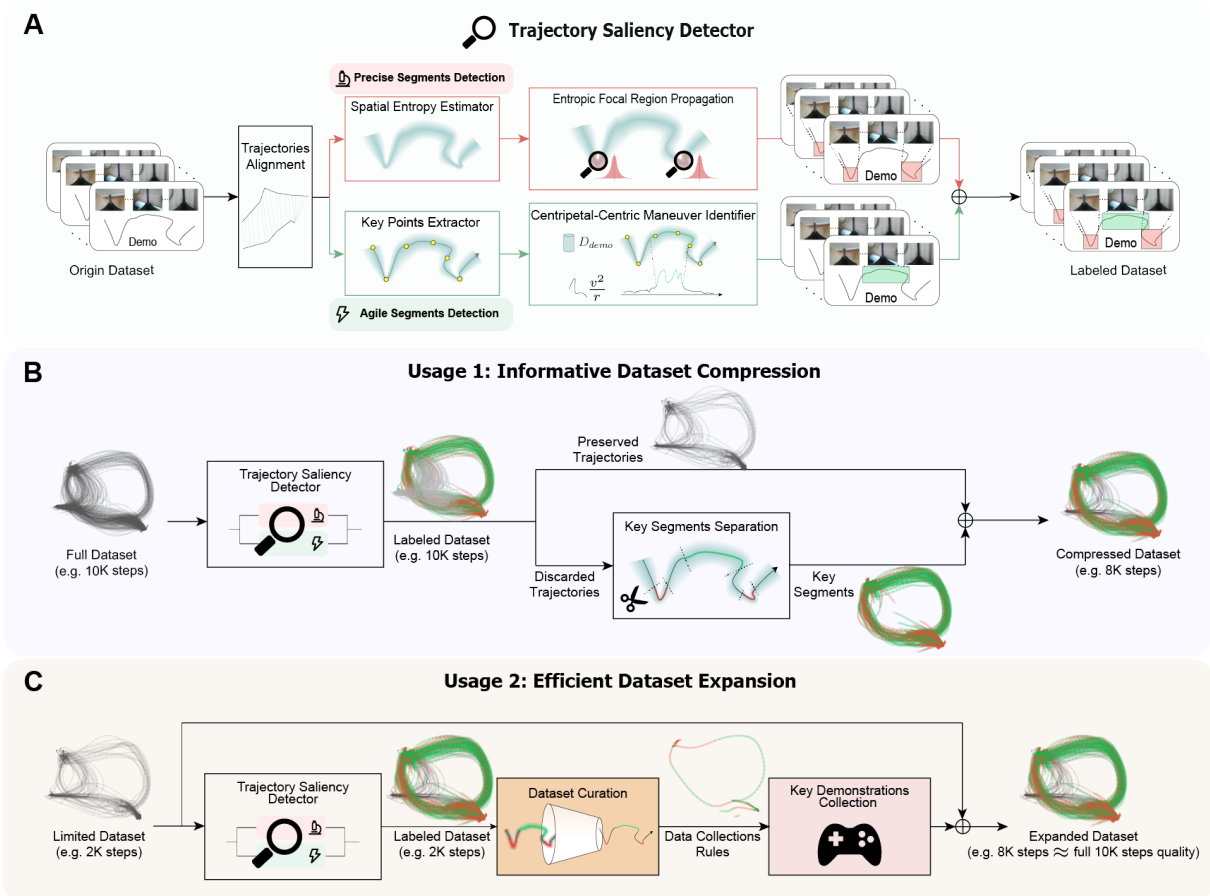


Fig. 1: Framework Overview of TSD. (A) TSD identifies precise and agile segments by using spatial entropy estimation and centripetal-centric maneuver identification. (B) Usage 1 (Informative Dataset Compression): TSD compresses large datasets by discarding redundant trajectories while preserving critical trajectory segments to maintain learning performance. (C) Usage 2 (Efficient Dataset Expansion): Starting from limited data, TSD guides the collection of high-value segments to expand the dataset to a quality level comparable to full-scale training sets.

Our main contributions are summarized as follows:

- We formalize trajectory saliency to characterize data heterogeneity, shifting the focus from uniform treatment to task-critical prioritization.
- We introduce TSD, a training-free mechanism leveraging spatial entropy and centripetal acceleration to automatically identify high-value segments without supervision.
- Simulated and real-world experiments indicate that prioritizing informative segments significantly enhances both collection and training efficiency while maintaining, or even enhancing model performance.

II. RELATED WORKS

A. Dataset Expansion

Current dataset expansion methods primarily aim to maximize data coverage with minimal human labor. Given the low cost and scalability of data acquisition in simulation, existing works heavily rely on synthetic environments, focusing on two main aspects: environmental randomization and spatial randomization.

Regarding environmental randomization, frameworks such as RoboTwin[21] employ domain randomization to generate expert trajectories across diverse objects, backgrounds, and lighting conditions. Other researchers leverage 3D Gaussian Splatting (3DGS) to transform real-world objects into high-fidelity 3D models, enabling precise simulation of object appearance [4], physical parameters [22], and material properties [23].

Parallel progress has been made in spatial randomization. AnchorDream [5] utilizes video diffusion to synthesize new trajectories from sparse initial data. MimicGen [11] decomposes demonstrations into object-centric stages to maintain precise relative poses between the robot and target objects, while DemoGen [12] extracts action-object relationships to autonomously adapt original trajectories for grasping objects at unseen locations. For more complex scenarios, MomaGen [14] extends these capabilities to loco-manipulation tasks by framing data generation as a constrained optimization problem. Similarly, DreamGen [13] integrates fine-tuned video models with pseudo-action generation models for dataset expansion.

Despite these advancements, significant challenges remain.

Efforts to narrow the Sim-to-Real gap are often hindered by the inherent difficulty in modeling complex contact dynamics and non-linear physical interactions. Furthermore, the massive volume of simulation-generated data significantly exceeds that of real-world collections, leading to a substantial decrease in training efficiency [5, 24]. Therefore, by conducting real-world data acquisition based on the critical trajectory segments identified by TSD, we achieve an effective trade-off between collection efficiency and data quality.

B. Dataset Compression

The objective of dataset compression is to identify data that best facilitates model performance, categorized into two granularities: trajectory-level and segment-level pruning.

At the trajectory level, existing methods typically rely on auxiliary discriminators. For instance, Demo-SCORE [15] employs a quality classifier trained on rollouts from multiple policy checkpoints. DemInf [16] learns structured low-dimensional representations of states and actions, using mutual information and k-nearest neighbors to filter high-quality demonstrations. Similarly, DataMIL [17] pretrains on a small-scale dataset to score and select the most contributory trajectories from the larger dataset.

At the segment level, research often focuses on retrieving task-relevant fragments from heterogeneous datasets [25]. Techniques involving optical flow [26] or pre-trained vision models [27, 28] are commonly used for matching. IWR [10] samples data from the prior dataset and applies weighted training to emphasize critical segments. L2R [29] utilizes contrastive and preference learning to derive a quality-scoring network for segments. To mitigate compounding errors, [8, 9] leverage the heterogeneity of robotic segments by downsampling non-essential movement phases.

A common limitation of these methodologies is their dependence on auxiliary models for quality assessment, which invariably incurs significant computational and time overhead. In contrast, our proposed Trajectory Saliency Detector (TSD) provides an intuitive, training-free alternative that offers a distinct advantage in operational efficiency.

III. METHODS

A. Preliminaries

Imitation Learning. Behavioral cloning (BC), the most popular form of imitation learning, is formulated as a supervised learning problem where a policy π_θ is trained to map a temporal context of observations to expert actions. Specifically, to account for partial observability or system dynamics, the input is defined as an observation window $H_t = \{o_{t-k+1}, \dots, o_t\}$ comprising the k most recent frames, and the policy is to minimize a loss function \mathcal{L} over an expert dataset \mathcal{D} :

$$\min_{\theta} \mathbb{E}_{(H,a) \sim \mathcal{D}} [\mathcal{L}(\pi_\theta(H), a)]$$

This mapping $\pi_\theta : H \rightarrow a$ allows the model to capture short-term temporal dependencies.

Markov Process. The core of a Markov Process is the Markov Property, which posits that, given a present state,

the future is independent of the past. Mathematically, for a sequence of states $s_0, s_1, s_2, \dots, s_t$, the transition probability is defined as:

$$P(s_{t+1} | s_t, s_{t-1}, \dots, s_0) = P(s_{t+1} | s_t)$$

In imitation learning, this implies that the optimal action a_t can be determined solely by a short history of recent observations s_t , without needing the entire history. Since each transition (s_t, a_t, s_{t+1}) is locally independent, we can break expert trajectories into individual samples. This allows us to only sample critical moments—such as the agile and precise segments—to increase data density in high-stakes regions of the state space. Thus, the aforementioned Markov Property establishes the mathematical foundation for TSD.

B. Trajectories Alignment

Due to variations in movement and environmental differences during human demonstrations, the time required to complete the same task inevitably varies across trials. Although the progression of task stages remains consistent across demonstrations, the raw recorded data exhibits significant temporal discrepancies. Therefore, aligning different trajectories is a critical preprocessing step. We employ Dynamic Time Warping (DTW) [30] to achieve this alignment. Specifically, DTW finds the optimal matching path between two sequences by non-linearly compressing or expanding the time axis to minimize the cumulative distance. This ensures that corresponding stages of the task are synchronized across the entire dataset, regardless of the execution speed of individual demonstrations.

Given two trajectory sequences $P = (p_1, p_2, \dots, p_n)$ and $Q = (q_1, q_2, \dots, q_m)$, where each point $p_i, q_j \in \mathbb{R}^d$ represents spatial coordinates, we define a hybrid distance metric $d(p_i, q_j)$ using Euclidean distance for translation and quaternion inner product distance for rotation. A cumulative cost matrix $D \in \mathbb{R}^{n \times m}$ is constructed where each element $D(i, j)$ represents the minimum cost to align the sequences up to indices i and j , computed via the recursive relation:

$$D(i, j) = d(p_i, q_j) + \min\{D(i-1, j), D(i, j-1), D(i-1, j-1)\}$$

By performing optimal backtracking from $D(n, m)$, we obtain a warping path $W = \{w_1, \dots, w_k\}$ that minimizes the total alignment cost. Crucially, this cost matrix and the resulting path W provide a deterministic mapping that allows us to precisely localize the segments identified on the reference timeline back to their corresponding positions within the original trajectories. To further refine the alignment process preventing pathological warping and enhance computational efficiency, we incorporate the Sakoe-Chiba window [31] as a global constraint on the warping path W , restricting the allowed matching range between trajectories P and Q within a bandwidth b such that $|i - j| \leq b$

C. Precise Segments Detection

Spatial Entropy Estimator. Based on the spatial distribution patterns inherent in human demonstration data [19], we propose a K-Nearest Neighbor (KNN) based spatial

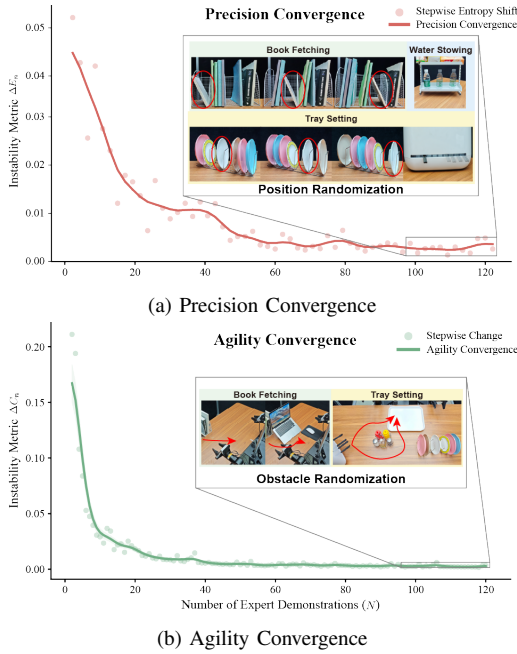


Fig. 2: Trends in TSD detection stability for precise and agile segments relative to demonstration volume. As the dataset expands, TSD’s detection of agile and precise segments stabilizes, demonstrating its consistency despite the randomization present in the demonstrations.

entropy calculation method. To ensure the stability of density estimation while capturing the dynamics of data collection, a sliding time window approach is employed. Following trajectory alignment based on the reference timeline, here we analyze the distribution density of demonstration data around a specific point in the robot’s task space in detail. For this given point at timestamp t , we define the set X consisting of all points within a time window T across all trajectories in the dataset. A distance matrix M (where $|X| = N$) is then computed for all pairs in X . For every point $x \in X$, we identify its k -th nearest neighbor and denote the distance as $R_k(x)$. The corresponding hypersphere volume for $x \in X$ is calculated as

$$V(x) = \frac{\pi^{d/2} R_k(x)^d}{\Gamma(d/2 + 1)}$$

Since density is independent of the end-effector orientation, we set the dimensionality $d = 3$, focusing solely on translational movement.

Then we estimate the Shannon entropy based on the hypersphere volumes of all points in set X to derive the spatial entropy at each point on the reference timeline, which denote as

$$H(t) = \frac{1}{N} \sum_{i=1}^N \ln\left(\frac{N \cdot V(x_i)}{k}\right) + \psi(k) - \psi(1)$$

where ψ represents the Digamma function. In the resulting entropy-over-time curve, local minima (valleys) represent potential precise operations. Under the assumption that expert demonstrations contain no meaningless pauses, we apply a velocity-weighting method to assign higher importance

to low-velocity states, which typically correlate with high-precision tasks. After normalizing the velocity v and its variance σ_v at each time step, a velocity weight

$$\omega = \min(1, \max(\overline{|v_{norm}|}, \lambda \cdot \sigma_{norm})) \cdot \alpha$$

is computed, where $\overline{|v_{norm}|} = \frac{|v|}{\max|v|}$, $\sigma_{norm} = \frac{\sigma_v}{\max(\sigma_v)}$, λ is a variance weight coefficient and α is a scaling factor. The weighted spatial entropy is then defined as:

$$H_{weighted} = H_{min} + \omega \cdot (H_{original} - H_{min})$$

Entropic Focal Region Propagation. Finally, we employ an adaptive low-entropy detection method to identify the temporal locations of fine-grained operations. This method adjusts the sensitivity of valley detection based on the dynamic range of the spatial entropy and the total task duration. To map these locations back to the original trajectories while maintaining stability, we use the median of the corresponding original points within a specific time window in the reference timeline as the mapping result. Recognizing that fine-grained operations usually persist over a duration, we utilize spatiotemporal expansion—extending the identified points within a small spatial and temporal range—to define the operation segments. We avoid applying the valley range from the reference timeline directly to the original trajectories because direct mapping often yields segments that are inconsistently long or short. Instead, by leveraging the characteristic that precise operations involve minimal positional change and require sustained duration, we can accurately isolate the relevant segments.

D. Agile Segments Detection

Key Points Extractor. We employ the Ramer-Douglas-Peucker [32] (RDP) algorithm to downsample the trajectory into a set of key anchor points. Specifically, given a trajectory P containing points $\{p_1, p_2, \dots, p_n\}$, it iteratively simplifies the curve by computing the perpendicular distance d of any point p_i relative to the line segment \overline{AB} formed by the segment’s start and end points:

$$d(p_i, \overline{AB}) = \frac{\|(p_i - A) \times (B - A)\|}{\|B - A\|}$$

If $\max(d) > \epsilon$, the curve is split at the point of maximum distance. Since the RDP algorithm focuses solely on geometric shape without considering system dynamics or task complexity, we further refine the extracted points through a multi-stage filtering process. First, we utilize a velocity threshold and local minima detection to identify and prune pause points and low-velocity drift, which are marked as redundant information. Subsequently, we introduce tortuosity τ as a metric to quantify the complexity of the movement, calculated via a sliding window as the ratio of the actual path length to the chord length:

$$\tau = \frac{\sum_{i=1}^{n-1} \|p_{i+1} - p_i\|}{\|p_n - p_1\|}$$

To enhance the robustness of this selection, a voting mechanism is applied to the endpoints of segments where the

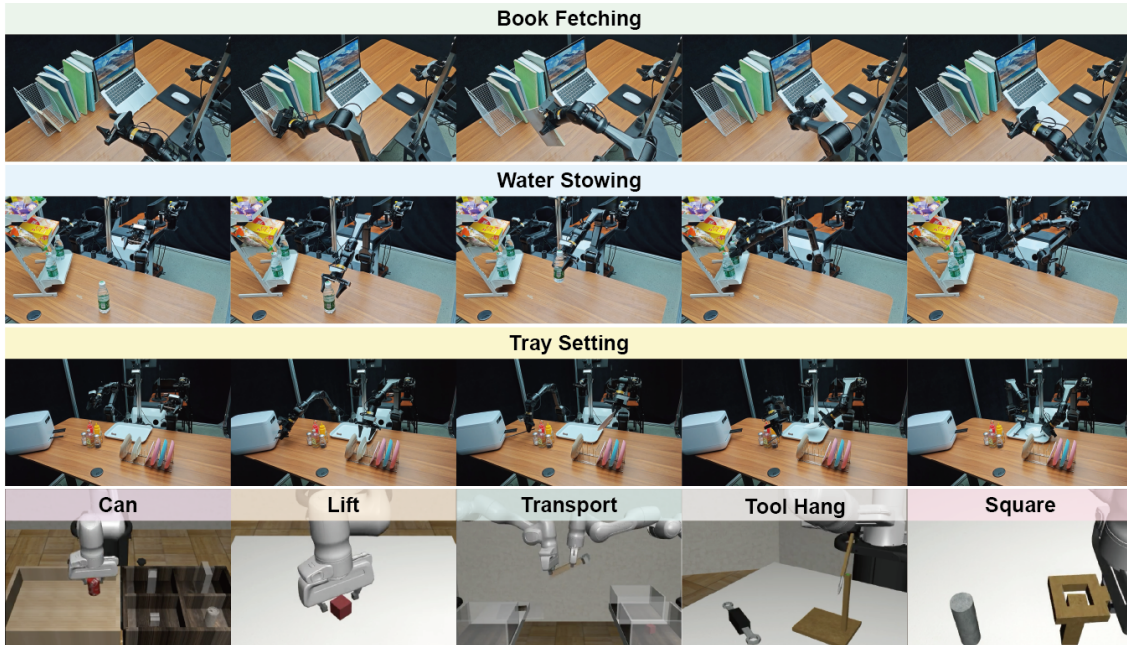


Fig. 3: **Task settings.** We validate the detection performance of TSD and corresponding model training effectiveness across five robomimic simulation tasks, and three real-world manipulation tasks, including single-arm and dual-arm scenarios.

tortuosity ranks in the top tier and exceeds a predefined threshold; only those key points that are consistently identified across multiple complex segments are retained, ensuring the stability and representativeness of the final geometric features.

Centripetal-Centric Maneuver Identifier. Building upon the previously extracted geometric key points, we incorporate the centripetal acceleration associated with each point on the curve to perform a joint analysis. Specifically, we use

$$a_c = v^2 \kappa$$

to serve as a quantitative measure of the dynamic intensity of the robot’s motion, where κ refer to curvature. An adaptive peak detection strategy, similar to what we used in precise segments detection is employed, where prominence thresholds and width constraints are dynamically established based on the overall a_c distribution across the entire trajectory. This allows for the capture of representative peak intervals. Any peak interval that coincides with a previously identified geometric key point is classified as a high-maneuverability operation, thereby achieving unsupervised recognition of critical task segments.

E. Consistency of Saliency Detection

Through experimental analysis showed in IV-D (as illustrated in Fig. 2), we observe that the proposed TSD algorithm exhibits significant consistency in extracting critical task features, characterized by the following two dimensions:

Numerical Consistency. This property signifies the stability of the detection results relative to the dataset size. Once the number of demonstrations reaches a threshold, the quantity and the spatiotemporal localization of identified precise and agile segments converge to a stable state. The

results do not fluctuate significantly with further increases in dataset size, proving that TSD effectively filters out random noise and captures the inherent structural features of the task.

Positional Consistency. This attribute reflects the robustness against randomization. In human demonstrations designed for model generalization, objects are often placed randomly within a workspace. Despite these variations, TSD accurately identifies salient trajectory segments. For precise segments, TSD consistently maps reference timeline to their corresponding locations in the original trajectories; for agile segments, centripetal acceleration features ensures robust detection stability.

IV. EXPERIMENTS

A. Implementation Details

Dataset Setup. Empirically, task scenarios in simulation environments are highly repeatable, enabling researchers to leverage open-source datasets with minimal overhead. In contrast, real-world experiments face significant challenges due to scene heterogeneity, often requiring manual, from-scratch data collection. Given the consistency of TSD—specifically, that its accuracy in detecting salient segments stabilizes rapidly as the data scale increases—we adopt an asymmetric experimental strategy. As showed in Fig. 1, We apply **Usage 1** in the resource-rich simulation to explore the compression performance of TSD on full datasets, while employing **Usage 2** in data-constrained real-world settings to align with practical requirements. Such a bifurcated validation not only proves the effectiveness of TSD but also aligns with the practical logic of robotic deployment.

Specifically, as showed in Fig. 3, for simulation, we utilize the robomimic[33] datasets and validate across five

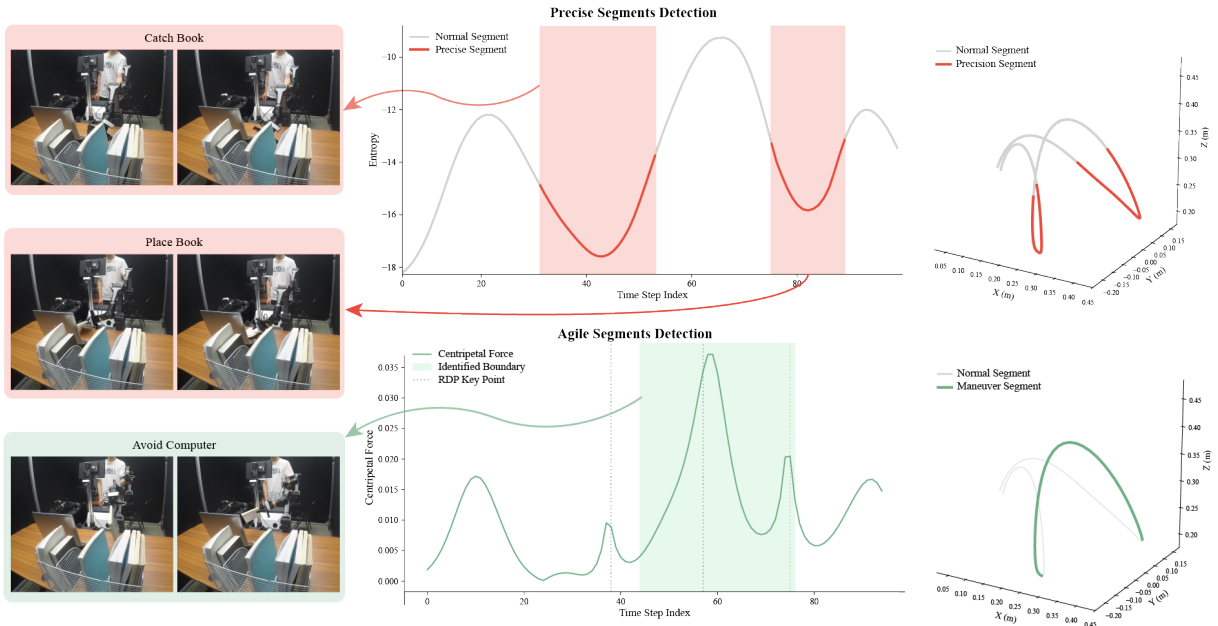


Fig. 4: **Qualitative results** on the book-fetching task. The visualization shows the detected precise and agile segments and the corresponding visual frames, demonstrating TSD’s accuracy in identifying salient segments. Low entropy at the beginning and the end is attributed to the identical starting and ending positions across all trajectories.

tasks. For real-world experiments, we design three daily manipulation tasks: two single-arm tasks—Book Fetching (retrieving a book from a shelf while bypassing a laptop) and Water Stowing (transferring a water bottle to a rack)—and one bimanual task, Tray Setting (simultaneously retrieving chopsticks and a plate while avoiding condiment bottles). All tasks incorporate spatial randomization. Specifically, in real-world tasks, this includes variations for grasping and placing, as well as the stochastic presence of obstacles, as illustrated in Fig. 2. For simulation, we employ the randomization protocols provided by Robosuite[20]. It’s worth noted that `base` serves as the initial point for dataset expansion via TSD. **Bold** values in Table I and II indicate the highest data efficiency (success rate per unit of data) within each group.

Training Setup. To evaluate the practical impact of TSD, we adopt CNN-based Diffusion Policy[34] for all experiments to investigate how TSD-derived dataset compression (Usage 1) and expansion (Usage 2) affect policy performance. We adopt observations from a third-view camera together with wrist-mounted cameras for every task, with recommended resolution provided by Robomimic in simulation and 240×320 for real-world experiments.

B. Simulation Experiments

Across all simulation tasks, success rates scale positively with dataset size, with `base` establishing a performance lower bound. Leveraging this baseline, we validate the efficacy of TSD-derived informative dataset compression through two dimensions. First, under a fixed frame budget, `A2/B2` consistently outperform their randomly sampled counterparts `A1/B1`, proving that TSD-extracted salient segments are superior for learning critical task behaviors.

Second, under a fixed trajectory count, `A2/B2` achieve comparable success rates to `A3/B3` while utilizing only 74.5% and 68.0% of their respective total frames, demonstrating remarkable data efficiency. Notably, in Lift and Transport tasks, `A2` even surpasses `A3`, demonstrating TSD’s ability to filter out redundant transitional motions and mitigate the impact of suboptimal data, leading to more efficient policy learning. The consistent findings across both `A` (50% scale) and `B` (100% scale) further underscore TSD’s robustness and scalability, showing its performance gains are independent of dataset size.

C. Real-World Experiments

Real-world experiments further validate the superiority of TSD in guiding dataset expansion. In Tray Setting and Book Fetching, `A2/B2` consistently outperform their randomly sampled counterparts `A1/B1` under constrained total frame budgets. A significant finding is that `B2` achieves a 3.3% success rate improvement or maintains parity while utilizing only 80.3% to 84.9% of the total frames compared to the full dataset (`B3`). This suggests that TSD-guided dataset expansion serves as an effective behavioral denoising mechanism. By precisely isolating salient segments in `base` as showed in Fig. 4, TSD enables efficient dataset expansion by targeting only key frames, which filters out redundant transitional motions and provides a training distribution with higher information density. In the Water Stowing task, while `B1` exhibits a slightly higher data efficiency, the margin relative to `B2` is negligible. This suggests that the task’s lower behavioral complexity allows random sampling to capture sufficient task-critical information, thereby reaching the performance ceiling of approximately 70% success rate. What’s more, TSD-based dataset expansion across various

TABLE I: Experiment Results

Model		Base	A1	A2(Ours)	A3	B1	B2(Ours)	B3	
Sim	Can	Succ.	53.3%	72.8%	82.0%	83.1%	91.7%	92.0%	93.7%
		Size	4617(19.9%)	7999(34.5%)	7884(33.9%)	11561(49.8%)	13863(59.7%)	13321(57.4%)	23207(100%)
	Lift	Succ.	96.6%	98.8%	99.3%	98.6%	98.6%	99.7%	99.3%
		Size	1992(20.6%)	3465(35.8%)	3399(35.1%)	4886(50.5%)	5919(61.2%)	5839(60.4%)	9666(100%)
	Tool Hang	Succ.	5.1%	35.7%	35.7%	36.4%	46.6%	56.8%	58.6%
		Size	18817(19.6%)	37077(38.6%)	34107(35.5%)	46740(48.7%)	65721(68.5%)	64237(66.9%)	95962(100%)
	Transport	Succ.	24.6%	49.5%	69.5%	73.3%	88.8%	95.3%	89.3%
		Size	18837(20.1%)	38774(41.2%)	40188(42.8%)	46902(49.2%)	78032(83.2%)	79044(84.3%)	93752(100%)
	Square	Succ.	11.1%	66.2%	68.8%	68.6%	73.3%	76.8%	77.5%
		Size	6060(20.1%)	11922(39.5%)	11526(38.2%)	15125(50.5%)	21021(69.7%)	21327(70.7%)	30154(100%)
Real	Tray	Succ.	0.0%	46.7%	60.0%	53.3%	76.7%	86.6%	83.3%
	Setting	Size	5335(21.0%)	11502(45.3%)	11418(45.0%)	12819(50.6%)	21516(84.9%)	21527(84.9%)	25350(100%)
	Water	Succ.	40.0%	53.3%	56.7%	63.3%	70.0%	70.0%	76.6%
	Stowing	Size	2108(20.5%)	4146(40.4%)	4194(40.8%)	5167(50.3%)	7767(75.6%)	7789(75.8%)	10269(100%)
	Book	Succ.	46.7%	53.3%	60.0%	66.7%	76.7%	93.3%	93.3%
	Fetching	Size	3661(19.7%)	7889(42.5%)	7854(42.3%)	9232(49.7%)	14886(80.1%)	14921(80.3%)	18578(100%)

¹ Size(Ratio): Size refer to the total frames in the dataset. Ratio refer to the proportion in the full dataset.

² base: 20% of the full dataset; A1/B1: Randomly sampled trajectories (size-matched to A2/B2); A2/B2: TSD-expanded(real)/compressed(sim) datasets, integrating salient segments from a matching number of trajectories as A3 and B3, respectively; A3/B3: 50% (A) and 100% (B) of the full dataset.

scales enables a reduction in data acquisition costs while simultaneously sustaining or even enhancing the model’s task success rate base on the base dataset.

D. Property Analysis

Numerical Consistency. To quantify the stability of maneuver and precision segments detection, a convergence evaluation method is used. For agility segments, maneuver boundaries of each trajectory are first converted into binary masks M_i (1 for maneuver points, 0 for nominal points). After temporal alignment, the consensus vector of the first n trajectories is calculated as: $C_n(t) = \frac{1}{n} \sum_{i=1}^n M_i(t)$. The Mean Absolute Error (MAE) defined as $\Delta C_n = \frac{1}{T} \sum_{t=1}^T |C_n(t) - C_{n-1}(t)|$, measures the impact of new samples on the identification of trajectory saliency. Similarly, a precision instability metric, $\Delta E_n = \frac{1}{T} \sum_{t=1}^T |E_n(t) - E_{n-1}(t)|$, is used to monitor $E_n(t)$. The convergence toward zero indicates that TSD has identified stable maneuver and precision segments.

As illustrated in Fig. 2, the detection instability for both agile and precise segments exhibits a downward trend and eventually stabilizes as the number of expert demonstrations N increases. Specifically, ΔC_n in Fig. 2b drops sharply and plateaus with only approximately 20 demonstrations, indicating the high sample efficiency in locating agile segments. In contrast, the decline of ΔE_n in Fig. 2a is more gradual, requiring a larger scale of demonstrations to reach a steady state. This difference reflects the inherent complexity of precision distributions compared to maneuver features due to randomization during data collection.

Positional Consistency. First, as shown in Table I, the training performance of models using TSD-augmented datasets consistently outperforms models trained on complete dataset of the same size, nearly matching or even exceeding the performance of the full dataset. This performance gain demonstrates TSD’s ability to accurately locate task-critical states, as any inconsistency in segment detection would have

introduced causal confusion and degraded learning efficiency. Second, as illustrated in Fig. 2, despite the randomization inherent in the collected datasets, TSD’s detection of salient segments eventually maintains remarkable stability. This capability to consistently locate salient segments amidst stochastic variations underscores the positional consistency of TSD.

E. Ablation Study

We conducted ablation studies on the Tool Hang simulation and the Book Fetching real-world task to investigate the relative contributions of precise and agile segments toward constructing datasets. The experimental results in Table II indicate that for manipulation tasks, the simultaneous integration of both segment types yields the most significant enhancement in model performance.

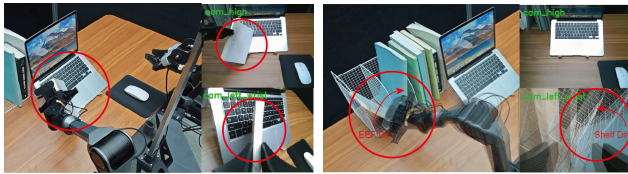
TABLE II: Ablation Experiments Results

Model		B2 (Ours)	C1	C2	D1	D2
Book	Succ.	93.3%	70.0%	63.3%	50.0%	40.0%
	Size	14921	13015	12938	7110	7220
Tool Hang	Succ.	56.8%	46.2%	52.9%	30%	16%
	Size	64237	60930	62283	22870	22179

¹ Size: Size refer to the total frames in the dataset.

² C1: a complete dataset of the same size as C2; C2: B2 without agile segments; D1: a complete dataset of the same size as D2; D2: B2 without precise segments.

Further analysis reveals that when only a single segment type is augmented, precise segments exert a more critical influence on success rates, outperforming agile-only datasets and matching the performance of complete datasets of the same size, underscoring the central role of fine-grained interaction in manipulation tasks. However, the absence of agile segments leads to a degradation in obstacle avoidance. As illustrated in Fig. 5a, the model often exhibits indecision



(a) Dataset w/o. agile segments (b) Dataset w/o. precise segments

Fig. 5: **Typical failures of models trained on datasets lacking specific segments.** The absence of agile segments leads to hesitation and collisions with obstacles, while the lack of precise segments results in imprecise grasping.

during inference, which can lead to collisions with obstacles. Conversely, datasets augmented solely with agile segments fail to complete the operation due to a lack of learned interaction logic, resulting in a sharp decline in success rates. A representative case is shown in Fig. 5b. Although the gripper initially aligns with the target, it drifts at the moment of grasping due to insufficient judgment. In summary, the inclusion of both segment types is indispensable for efficiently constructing an informative dataset.

V. CONCLUSIONS

We present Trajectory Saliency Detector (TSD), a plug-and-play, physics-inspired tool to identify fine-manipulation and agile segments. Leveraging TSD’s ability to isolate salient trajectory segments, we propose two primary usage: informative dataset compression and efficient dataset expansion.

Both simulation and real-robot experiments demonstrate the efficacy of TSD. Specifically, it reduces training costs through informative dataset compression while minimizing human labor in data acquisition via efficient dataset expansion, all without compromising the model’s capabilities.

This approach offers a simple and accessible preprocessing solution for imitation learning datasets. Future work will focus on extending the applicability of this method to systems with higher degrees of freedom, such as dexterous hands or loco-manipulation tasks.

REFERENCES

- [1] F. Lin *et al.*, “Data scaling laws in imitation learning for robotic manipulation,” *arXiv preprint arXiv:2410.18647*, 2024.
- [2] B. Liu *et al.*, “Libero: Benchmarking knowledge transfer for lifelong robot learning,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 44 776–44 791, 2023.
- [3] W. Wu *et al.*, “A pragmatic v1a foundation model,” *arXiv preprint arXiv:2601.18692*, 2026.
- [4] J. Yu *et al.*, “Real2render2real: Scaling robot data without dynamics simulation or robot hardware,” *arXiv preprint arXiv:2505.09601*, 2025.
- [5] J. Ye *et al.*, “Anchordream: Repurposing video diffusion for embodiment-aware robot data synthesis,” *arXiv preprint arXiv:2512.11797*, 2025.
- [6] S. Belkhale, Y. Cui, and D. Sadigh, “Data quality in imitation learning,” *Advances in neural information processing systems*, vol. 36, pp. 80 375–80 395, 2023.
- [7] M. Finzi *et al.*, “From entropy to epiplexity: Rethinking information for computationally bounded intelligence,” *arXiv preprint arXiv:2601.03220*, 2026.
- [8] L. X. Shi *et al.*, “Waypoint-based imitation learning for robotic manipulation,” in *Conference on Robot Learning*, PMLR, 2023, pp. 2195–2209.

- [9] S. Belkhale, Y. Cui, and D. Sadigh, “Hydra: Hybrid robot actions for imitation learning,” in *Conference on Robot Learning*, PMLR, 2023, pp. 2113–2133.
- [10] A. Xie *et al.*, “Data retrieval with importance weights for few-shot imitation learning,” in *Conference on Robot Learning*, PMLR, 2025, pp. 1–16.
- [11] A. Mandlekar *et al.*, “Mimicgen: A data generation system for scalable robot learning using human demonstrations,” in *Conference on Robot Learning*, PMLR, 2023, pp. 1820–1864.
- [12] Z. Xue *et al.*, “Demogen: Synthetic demonstration generation for data-efficient visuomotor policy learning,” *arXiv preprint arXiv:2502.16932*, 2025.
- [13] J. Jang *et al.*, “Dreamgen: Unlocking generalization in robot learning through video world models,” *arXiv preprint arXiv:2505.12705*, 2025.
- [14] C. Li* *et al.*, *Momagen: Generating demonstrations under soft and hard constraints for multi-step bimanual mobile manipulation*, 2025.
- [15] A. S. Chen *et al.*, “Curating demonstrations using online experience,” *arXiv preprint arXiv:2503.03707*, 2025.
- [16] J. Hejna *et al.*, “Robot data curation with mutual information estimators,” *arXiv preprint arXiv:2502.08623*, 2025.
- [17] S. Dass *et al.*, “Datamil: Selecting data for robot imitation learning with datamodels,” *arXiv preprint arXiv:2505.09603*, 2025.
- [18] P. Eckert and A. J. Ijspeert, “Benchmarking agility for multilegged terrestrial robots,” *IEEE Transactions on Robotics*, vol. 35, no. 2, pp. 529–535, 2019.
- [19] L. Guo *et al.*, “Demospeedup: Accelerating visuomotor policies via entropy-guided demonstration acceleration,” *arXiv preprint arXiv:2506.05064*, 2025.
- [20] Y. Zhu *et al.*, “Robosuite: A modular simulation framework and benchmark for robot learning,” *arXiv preprint arXiv:2009.12293*, 2020.
- [21] T. Chen *et al.*, “Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation,” *arXiv preprint arXiv:2506.18088*, 2025.
- [22] H. Fan *et al.*, “Twinaligner: Visual-dynamic alignment empowers physics-aware real2sim2real for robotic manipulation,” *arXiv preprint arXiv:2512.19390*, 2025.
- [23] J. Li *et al.*, “M³A policy: Mutable material manipulation augmentation policy through photometric re-rendering,” *arXiv preprint arXiv:2512.01446*, 2025.
- [24] H. Bi *et al.*, “Motus: A unified latent action world model,” *arXiv preprint arXiv:2512.13030*, 2025.
- [25] M. Du *et al.*, “Behavior retrieval: Few-shot imitation learning by querying unlabeled datasets,” *arXiv preprint arXiv:2304.08742*, 2023.
- [26] L.-H. Lin *et al.*, “Flowretrieval: Flow-guided data retrieval for few-shot imitation learning,” in *Conference on Robot Learning*, PMLR, 2025, pp. 4084–4099.
- [27] S. Nasiriany *et al.*, “Learning and retrieval from prior data for skill-based imitation learning,” in *Conference on Robot Learning*, PMLR, 2023, pp. 2181–2204.
- [28] Y. Kuang *et al.*, “Ram: Retrieval-based affordance transfer for generalizable zero-shot robotic manipulation,” in *Conference on Robot Learning*, PMLR, 2025, pp. 547–565.
- [29] S. Kuhar *et al.*, “Learning to discern: Imitating heterogeneous human demonstrations with preference and representation learning,” in *Conference on Robot Learning*, PMLR, 2023, pp. 1437–1449.
- [30] T. Giorgino, “Computing and visualizing dynamic time warping alignments in r: The dtw package,” *Journal of statistical Software*, vol. 31, pp. 1–24, 2009.
- [31] Z. Geler *et al.*, “Dynamic time warping: Itakura vs sakoe-chiba,” in *2019 IEEE International Symposium on INnovations in Intelligent Systems and Applications (INISTA)*, IEEE, 2019, pp. 1–6.
- [32] D. H. Douglas and T. K. Peucker, “Algorithms for the reduction of the number of points required to represent a digitized line or its caricature,” *Cartographica: the international journal for geographic information and geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.
- [33] A. Mandlekar *et al.*, “What matters in learning from offline human demonstrations for robot manipulation,” in *arXiv preprint arXiv:2108.03298*, 2021.
- [34] C. Chi *et al.*, “Diffusion policy: Visuomotor policy learning via action diffusion,” *The International Journal of Robotics Research*, vol. 44, no. 10-11, pp. 1684–1704, 2025.