ROSER: FEW-SHOT ROBOTIC SEQUENCE RETRIEVAL FOR SCALABLE ROBOT LEARNING

Zillur Rahman *
University of Nevada, Las Vegas

Eddison Pham
University of Toronto

Alejandro Daniel Noel
LatentWorlds AI, NL

Cristian Meo †
LatentWorlds AI, NL

ABSTRACT

A critical bottleneck in robot learning is the scarcity of task-labeled, segmented training data, despite the abundance of large-scale robotic datasets recorded as long, continuous interaction logs. Existing datasets contain vast amounts of diverse behaviors, yet remain structurally incompatible with modern learning frameworks that require cleanly segmented, task-specific trajectories. We address this data utilization crisis by formalizing robotic sequence retrieval: the task of extracting reusable, task-centric segments from unlabeled logs using only a few reference examples. We introduce ROSER, a lightweight few-shot retrieval framework that learns task-agnostic metric spaces over temporal windows, enabling accurate retrieval with as few as 3-5 demonstrations, without any task-specific training required. To validate our approach, we establish comprehensive evaluation protocols and benchmark ROSER against classical alignment methods, learned embeddings, and language model baselines across three large-scale datasets (e.g., LIBERO, DROID, and nuScenes). Our experiments demonstrate that ROSER consistently outperforms all prior methods in both accuracy and efficiency, achieving sub-millisecond per-match inference while maintaining superior distributional alignment. By reframing data curation as few-shot retrieval, ROSER provides a practical pathway to unlock underutilized robotic datasets, fundamentally improving data availability for robot learning.

1 INTRODUCTION

The promise of generalist robot learning depends critically on access to massive, diverse datasets spanning tasks, embodiments, and environments (Brohan et al., 2023). Yet despite significant progress in large-scale data collection (Caesar et al., 2020; Liu et al., 2023; Khazatsky et al., 2024), a fundamental bottleneck persists: the vast majority of existing robotic data remains structurally incompatible with modern learning frameworks (e.g., Vision-Language-Action models (Shao et al., 2025), World Models (Meo et al., 2024)). While state-of-the-art approaches demand cleanly segmented, task-labeled trajectories (Mandlekar et al., 2019; Ebert et al., 2022), real-world robot logs are recorded as long, continuous streams lacking task boundaries, semantic labels, or hierarchical annotations (Mandlekar et al., 2019; Ebert et al., 2022).

This structural mismatch creates a data utilization crisis. Datasets like nuScenes (Caesar et al., 2020), DROID (Khazatsky et al., 2024), and LIBERO (Liu et al., 2023) contain thousands of hours of diverse behaviors, yet extracting reusable task segments requires prohibitively expensive human annotation or domain-specific heuristics that do not generalize (Memmel et al., 2025; Kumar et al., 2025).

We propose re-framing data curation as a few-shot retrieval problem. Rather than requiring dense supervision, a small number of reference demonstrations should suffice to identify all semantically similar segments within large, unlabeled datasets. This paradigm shift eliminates exhaustive anno-

*Correspondance to: zillur.mle@gmail.com
†Advisor: cristianmeo@latentworlds.ai, Website: www.latentworlds.ai

tation, enables rapid adaptation to new tasks, and naturally handles execution variability through learned similarity rather than brittle template matching.

However, robotic sequence representation poses fundamental challenges. Unlike images or text, robot trajectories are variable-length, high-dimensional time series with complex temporal dependencies, control noise, and execution stochasticity (Argall et al., 2009; Braglia et al., 2024). Existing approaches fall short: classical trajectory alignment methods like dynamic time warping (DTW) lack semantic understanding (Sakoe & Chiba, 1978), embedding-based approaches remain sensitive to execution variability (Rhodes & Lee, 2024; Memmel et al., 2025; Zhu, 2024), and language or vision-guided methods (Zeng et al., 2023; Shao et al., 2025) struggle with fine-grained kinematic structure. Recent work on trajectory representations emphasizes learning compact embeddings that capture spatio-temporal and task-level semantics (Zhu, 2024; Memmel et al., 2025), yet these methods typically require task-specific fine-tuning or dense supervision.

Drawing on metric-based few-shot learning (Snell et al., 2017; Koch et al., 2015; Sung et al., 2018), we introduce Robotic Sequence Retrieval (ROSER), a lightweight framework that learns task-agnostic metric spaces directly from raw proprioceptive time series. ROSER constructs prototype representations from a handful of reference examples and retrieves similar segments via learned distances over short temporal windows, requiring no task-specific training at deployment. This approach is based on Prototypical Networks (Snell et al., 2017), which represent each task by means of the embedding of its support examples, enabling closest-prototype comparisons that naturally support the quick retrieval of new tasks. To the best of our knowledge, this is among the first works to study few-shot robotic sequence retrieval using only proprioceptive time-series signals.

Our main contributions are: (C1) Formalizing robotic sequence retrieval and proposing ROSER, a few-shot framework that enables accurate retrieval with 3-5 examples.(C2) Establish evaluation protocols and benchmarking across three large-scale datasets against classical, learned, and foundation model baselines. (C3) Demonstrating that ROSER outperforms all baselines in accuracy and efficiency, providing a practical solution to unlock underutilized robotic data at scale.

2 Related Works

Data Retrieval in Robotics and Autonomous Driving. Large-scale robotic and driving datasets enable advances in manipulation, navigation, and scene understanding, yet retrieving task-relevant segments remains challenging. Methods such as STRAP (Memmel et al., 2025) and COLLAGE (Kumar et al., 2025) rely on heuristics, which are often dataset-specific and not scalable. Visual place recognition (VPR) surveys (Lowry et al., 2015; Masone & Caputo, 2021) and temporal contrastive learning (Dave et al., 2021) show that learned embeddings, static or temporal, can improve the retrieval of semantically meaningful segments. Multi-modal embeddings like CLIP (Radford et al., 2021a) further enable few-shot, cross-domain generalization. In autonomous driving, BEV-based retrieval (Tang et al., 2024) requires multi-view images and large labeled datasets, while front-facing models for action or scene classification (Noguchi & Tanizawa, 2023; Vellenga et al., 2025; 2024; Wang, 2025) are data- and compute-intensive. In contrast, we aim to retrieve scenes using only a few labeled samples with a lightweight, generalizable model suitable for resource-constrained environments.

Few-Shot Learning. Few-Shot Learning (FSL) addresses the gap between human-like learning, which requires few examples, and traditional deep learning, which relies on massive annotated datasets. Meta-learning is a prominent paradigm in FSL. Koch et al. (2015) introduced Siamese Networks to learn similarity functions between image pairs. Snell et al. (2017) proposed Prototypical Networks, representing each class by the mean of its examples in embedding space, enabling nearest-prototype classification. Finn et al. (2017) developed Model-Agnostic Meta-Learning (MAML), which finds parameter initializations optimized for fast adaptation to new tasks. More recently, CLIP Radford et al. (2021b) demonstrated that large-scale image-text pretraining enables strong zero- and few-shot transfer by aligning visual and linguistic embeddings. These approaches provide the foundation for few-shot retrieval of driving scenes from minimal labeled data.

Time-Series Motifs. Time-series motifs are recurring, characteristic subsequences within sequential data that can capture meaningful patterns without requiring full supervision. Early approaches such as shapelets identify discriminative subsequences for classification by searching for patterns that best

separate classes in labeled datasets (Grabocka et al., 2014). More recent work leverages scalable motif discovery frameworks, such as STUMPY (Law, 2019), which uses efficient matrix (Yeh et al., 2016) computations to detect repeated patterns across long sequences, enabling fast retrieval of relevant subsequences in large datasets. Van Wesenbeeck et al. (2023) extend this line of work to local contrastive motif learning, using self-supervised embeddings to identify semantically meaningful motifs across variable-length sequences, even in low-data settings. Despite these advances, existing motif-based methods have not been widely applied to robotics or autonomous driving data, where task-relevant segments are rare, temporally extended, and high-dimensional. By leveraging motif discovery for few-shot retrieval, our approach can efficiently identify meaningful behavioral patterns from minimal demonstrations, providing a scalable and generalizable solution for real-world robotic learning.

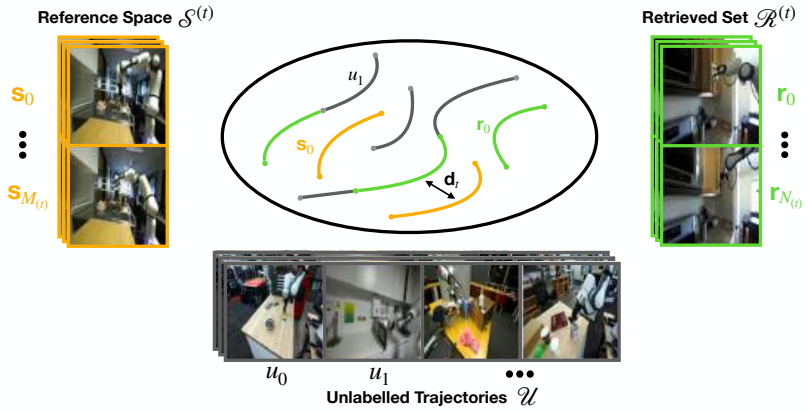

Figure 1: Retrieval Framework. The time-series encoder f_θ uses labeled dataset $\mathcal{S}$ and create a metric space where similar tasks are grouped together and create prototypes while different tasks are separated from each others. The embeddings of unlabeled trajectories $\mathcal{U}$ are compared with prototypes using learned metric $Dist(.)$ to find the closest match and create retrieved set $\mathcal{R}$.

3 ROSER

In this work, we propose a few-shot metric learning framework for retrieving semantically similar robotic and driving maneuvers from long, unsegmented time-series demonstrations. Our approach learns a task-agnostic embedding space in which short temporal sequences corresponding to the same maneuver are clustered together, while dissimilar behaviors are separated. The learned embedding enables scalable retrieval using only a few reference examples per task.

Preliminaries $\mathcal{S}$ denotes the labeled meta-training set used to learn the encoder f_θ. For a target task t, $\mathcal{S}^{(t)}$ denotes its support set of size K_{shot}, and $\mathbf{c}^{(t)}$ is the corresponding prototype (Eq. 1). $\mathcal{U}$ denotes the unlabeled corpus of long unsegmented trajectories from which we retrieve matching subsequences using sliding windows of size W and stride s (see Table 6 and Appendix A for a full notation overview).

3.1 Problem Formulation

Let $\mathcal{S} = \{(\mathbf{X}_i, y_i)\}$ be a dataset of labeled robotic tasks, where $\mathbf{X}_i \in \mathbb{R}^{L_i \times F}$ represents a sequence of robot states (e.g., joint angles, end-effector pose, gripper state, velocity, yaw rate) of length L_i, and $y_i \in \{1, \ldots, C\}$ is the task label.

Our goal is to learn an embedding function,

$$f_\theta : \mathbb{R}^{L \times F} \to \mathbb{R}^{d_{emb}}$$

that maps raw robotics time-series segments into a shared metric space. At test time, given a target task t specified by a small support set $\mathcal{S}^{(t)}$ (with K_{shot} examples), our goal is to retrieve all matching

subsequences from a large, unlabeled offline dataset $\mathcal{U} = \{\mathbf{X}_j\}$ where $\mathbf{X}_j \in \mathbb{R}^{L_j \times F}$ and typically $L_j \gg L_i$. Our retrieval framework is illustrated in Figure 1.

3.2 Time-Series Encoder

We adopt a Prototypical Network architecture (Snell et al., 2017) to learn a metric space for task retrieval. Unlike standard image-based implementations, our encoder f_θ must handle variable-length sequences inherent to robotic tasks.

We use a 1D convolutional network as the sequence encoder to extract time-series data features. Our choice of a streamlined 1D CNN backbone over higher-capacity architectures, such as Transformers (Vaswani et al., 2023) or Large Language Model (LLM) adaptations (Chang et al., 2023), is motivated by the extreme data scarcity inherent in few-shot robot imitation. In our setting, f_θ must learn a discriminative metric space using a minimal support set of reference samples. While Transformers have shown success in large-scale sequence modeling, they possess a "weak" inductive bias, requiring massive datasets to learn temporal dependencies from scratch (Dosovitskiy et al., 2021). In contrast, 1D CNNs provide two critical inductive biases for robotic trajectories, locality and shift-equivariance.

Locality: Robotic control signals are governed by physical laws where the state at time t is most strongly correlated with its immediate neighbors. Convolutional kernels explicitly exploit this local structure.

Temporal Shift-Equivariance: A specific maneuver (e.g., a "grasp") remains semantically identical regardless of when it occurs within the sliding window. CNNs are mathematically equivariant to such shifts, whereas Attention mechanisms must rely on complex positional encodings to learn this invariance (Ismail Fawaz et al., 2019).

Furthermore, high-parameter models are prone to manifold collapse and overfitting when the support set is limited to only a few labeled examples (Snell et al., 2017). By utilizing a constrained convolutional architecture, we enforce a smoother latent manifold that generalizes more robustly to the novel task variations encountered in our few-shot experiments. The architecture details of the sequence encoder is reported in Appendix A.3.

3.3 Prototype Construction

Prototypical Networks operate by computing a "prototype" $\mathbf{c}^t$ for each task t, defined as the mean embedding of its support examples (Snell et al., 2017). For a task t with support set $\mathcal{S}^{(t)}$:

$$\mathbf{c}^{(t)} = \frac{1}{K_{\text{shot}}} \sum_{\mathbf{x} \in \mathcal{S}^{(t)}} f_\theta(\mathbf{x}) \tag{1}$$

where $\mathcal{S}^{(t)}$ is the reference support set sampled within the current training episode. The prototype $\mathbf{c}^{(t)}$ thus represents the task's centroid in the embedding space, acting as the representative template for episodic optimization.

3.4 Episodic Training Paradigm

To enable few-shot generalization, we train the network using an episodic paradigm. In each training iteration, we sample an "episode" rather than a standard mini-batch. An episode is constructed by randomly selecting N_{way} tasks from the task set $\mathcal{T}$. For each selected task t, we sample K_{shot} support examples to form $\mathcal{S}^{(t)}$ and N_q query examples to form $\mathcal{Q}^{(t)}$.

The training objective is to minimize the negative log-probability of the true task t for each query example $\mathbf{x} \in \mathcal{Q}^{(t)}$ via a softmax over distances in the latent manifold:

$$p(y = t|\mathbf{x}) = \frac{\exp\left(-\text{Dist}(f_\theta(\mathbf{x}), \mathbf{c}^{(t)})\right)}{\sum_{t'} \exp\left(-\text{Dist}(f_\theta(\mathbf{x}), \mathbf{c}^{(t')})\right)} \tag{2}$$

where $\text{Dist}(\mathbf{z}, \mathbf{c}^{(t)}) = ||\mathbf{z} - \mathbf{c}^{(t)}||^2$ is the squared Euclidean distance. The loss function $\mathcal{L}$ for an episode is defined as the average negative log-likelihood over all query samples across all sampled

Algorithm 1 Robotic Sequence Retrieval Task (General)

Input: Reference support set $\mathcal{S}$, Unlabeled dataset $\mathcal{U}$
Components: Encoder (or feature map) f_θ, distance $\text{Dist}(\cdot,\cdot)$, template builder $\text{BuildTemplates}(\cdot)$
Hyperparameters: W (window size), s (stride), τ (NMS threshold)

Template Construction:
Compute templates $\mathcal{C} = \text{BuildTemplates}(\mathcal{S}, f_\theta)$
Initialize retrieval set $\mathcal{R} = \emptyset$
for each template $\mathbf{c} \in \mathcal{C}$ **do**
  for each trajectory $u \in \mathcal{U}$ **do**
    for each window $\mathbf{w}_t \in u$ with stride s **do**
      Compute distance $d_t = \text{Dist}(f_\theta(\mathbf{w}_t), \mathbf{c})$
      $\mathcal{R} = \mathcal{R} \cup \{(d_t, \text{start}_t, \text{end}_t)\}$
    end for
  end for
end for
Sort $\mathcal{R}$ by distance d_t in ascending order
Apply Non-Maximum Suppression (NMS) on $\mathcal{R}$ using threshold τ
Output: Ranked list of retrieved task candidates

Algorithm 2 ROSER Episodic Training

Input: Labeled training data $\mathcal{S}$
Hyperparameters: $N_{\text{way}}, K_{\text{shot}}, N_q$

repeat
  Sample N_{way} task classes from $\mathcal{S}$
  for each task $t \in \{1, \ldots, N_{\text{way}}\}$ **do**
    Sample support set $\mathcal{S}^{(t)}$ (K_{shot} samples) and query set $\mathcal{Q}^{(t)}$ (N_q samples) from $\mathcal{S}$
    Compute prototype $\mathbf{c}^{(t)} = \text{Aggregate}(\{f_\theta(\mathbf{x}) : \mathbf{x} \in \mathcal{S}^{(t)}\})$
  end for
  $\mathcal{L} = \sum_t \sum_{\mathbf{x} \in \mathcal{Q}^{(t)}} -\log \frac{\exp(-\text{Dist}(f_\theta(\mathbf{x}), \mathbf{c}^{(t)}))}{\sum_{t'} \exp(-\text{Dist}(f_\theta(\mathbf{x}), \mathbf{c}^{(t')}))}$
  Update θ via backpropagation on $\mathcal{L}$
until convergence
Output: Trained encoder f_θ

tasks:

$$\mathcal{L} = -\frac{1}{N_{\text{way}} N_q} \sum_{t=1}^{N_{\text{way}}} \sum_{\mathbf{x} \in \mathcal{Q}^{(t)}} \log p(y = t|\mathbf{x}) \tag{3}$$

This optimization objective forces the encoder f_θ to cluster embeddings of the same task type tightly around their respective prototypes while maximizing the inter-class distance between different task types.

3.5 Retrieval

Following training, the encoder f_θ is deployed for retrieval on unsegmented offline sequences. Task-specific prototypes $\mathbf{c}^{(t)}$ are computed from the K_{shot}-shot support examples for each task following equation 1. Once the prototype is established, a sliding window search is executed across the unlabeled sequences from the dataset $\mathcal{U}$. We extract temporal windows $\mathbf{w}_t$ where the window size W and stride s depends on the sequence length and each window is mapped into the metric space via the trained encoder to compute the squared Euclidean distance $d_t = ||f_\theta(\mathbf{w}_t) - \mathbf{c}^{(t)}||^2$.

To handle the high density of overlapping candidates generated by the sliding window, we incorporate a post-processing stage using Non-Maximum Suppression (NMS) (Hosang et al., 2017). By ranking detected windows according to their alignment distance and filtering those that exceed an overlap threshold τ, we ensure that the system retrieves distinct physical maneuvers rather than redundant

Table 1: Performance of baselines and ROSER on LIBERO dataset. Dark color represent best result and light color represent second best result. All models have same top-k results. Time-series models like Stumpy, Dtaidistance, and ROSER perform better compared to Large Models. ROSER performs best or second best in all of the metrics.

MODEL	WD (↓)	DTW NN (↓)	SPECTRAL WD (↓)	TEMP CORR. (↑)	DENSITY (↑)	DIVERSITY (↓)
GEMMA	0.17	12.98	0.0028	0.51	0.32	16.29
LLAMA	0.16	12.08	0.0024	0.57	0.37	14.61
MOMENTFM	0.14	11.04	0.0023	0.62	0.41	15.10
QWEN	0.17	13.04	0.0029	0.51	0.29	15.07
SHAPELET	0.19	11.81	0.0034	0.56	0.40	11.74
DTAIDISTANCE	0.092	**6.42**	0.0022	0.70	**0.75**	**7.27**
STUMPY	0.11	7.93	**0.0018**	**0.72**	0.64	10.92
ROSER	**0.086**	6.57	0.0019	**0.72**	0.69	8.8

temporal fragments of the same event. This combined approach allows for precise and scalable identification of target tasks within continuous, high-dimensional robotic data. Finally, the remaining candidates are ranked by their distance d_t in ascending order, and the top-k non-overlapping windows are retained as the definitive retrieved task candidates.

To evaluate the efficacy of our proposed encoder and compare with baselines, we formulate a general task retrieval framework (Algorithm 1) that accommodates both learned representations and classical distance-based measures. We represent all models through a unified retrieval interface defined by an encoder f_θ and a distance metric $\text{Dist}(\cdot)$. For ROSER, episodic training (Algorithm 2) optimizes θ to minimize a discriminative loss $\mathcal{L}_{\text{task}}$. For high-capacity foundation models such as LLMs or time-series foundation models, we bypass training and utilize frozen pre-trained weights to extract features. Finally, for non-parametric baselines like DTW (Meert et al., 2020) and STUMPY (Law, 2019), f_θ is an identity mapping, and retrieval is performed via direct sequence alignment.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Baselines and Ablations. We compare our approach with representative sequence and motif-based retrieval methods such as STUMPY (Law, 2019; Yeh et al., 2016), Dtaidistance (Meert et al., 2020), and Shapelets (Grabocka et al., 2014). We also evaluate three LLM-based embedding models (Qwen3 (Yang et al., 2025), Gemma3 (Team & et al., 2025), and LLama-Nemotron (Babakhin et al., 2025)) as well as one time-series foundation model (MOMENT (Goswami et al., 2024)).

Datasets We evaluate our approach on three large-scale datasets LIBERO (Liu et al., 2023), DROID (Khazatsky et al., 2024), and nuScenes (Caesar et al., 2020), covering robot manipulation and autonomous driving, focusing on fundamental manipulation tasks and driving maneuvers. Each dataset provides rich multimodal data and supports reproducible evaluation of learning, transfer, and robustness. The details of the datasets are reported in Appendix A.4.

Sliding-window retrieval and post-processing. Retrieval is performed by extracting fixed-length windows of size W with stride s from each sequence in $\mathcal{U}$ and ranking windows by $\text{Dist}(f_\theta(\mathbf{w}), \mathbf{c}^{(t)})$ (Algorithm 1). To avoid redundant overlaps, we apply Non-Maximum Suppression (NMS) with threshold τ. The window size W for retrieval is set to the mean length of the support examples, and we use stride $s = W/4$ to ensure dense coverage of potential start times.

Implementation Details. We implement a K-shot robotics data retrieval model to retrieve data from large unlabeled datasets. Each sample is represented as a multivariate time series. We train for 200 episodes using Adam optimizer (Kingma & Ba, 2017) with learning rate $1\text{e}{-3}$ on an NVIDIA A100 GPU.

Evaluation Metrics Primary objective. Our primary goal is to retrieve task-consistent segments (i.e. semantically correct maneuvers) from long unsegmented logs. Because exhaustive segment-level labels are not always available at scale, we evaluate retrieval quality using complementary

Table 2: Performance of baselines and ROSER on nuScenes dataset. Time-series models like Stumpy, Dtaidistance, and ROSER perform better compared to Large Models. ROSER performs best or second best in most of the metrics.

MODEL	WD (↓)	DTW NN (↓)	SPECTRAL WD (↓)	TEMP CORR. (↑)	DENSITY (↑)	DIVERSITY (↓)
GEMMA	1.02	36.56	0.00071	0.35	0.59	51.69
LLAMA	0.73	30.74	0.00073	0.37	0.79	50.05
MOMENTFM	0.67	24.09	0.00086	0.42	0.79	48.86
QWEN	0.96	43.95	0.00083	0.28	0.49	74.38
SHAPELET	0.93	32.25	0.0015	0.14	0.46	48.82
DTAIDISTANCE	0.54	13.19	0.00069	0.48	1.07	**19.98**
STUMPY	0.51	21.38	0.00071	0.53	0.85	52.70
ROSER	**0.27**	**12.84**	**0.00056**	**0.53**	**1.30**	26.69

Table 3: Performance of baselines and ROSER on DROID dataset. Time-series models like Stumpy, Dtaidistance, and ROSER perform better compared to Large Models. ROSER performs best or second best in most of the metrics.

MODEL	WD (↓)	DTW NN (↓)	SPECTRAL WD (↓)	TEMP CORR. (↑)	DENSITY (↑)	DIVERSITY (↓)
GEMMA	0.23	30.33	0.0021	0.35	0.69	49.54
LLAMA	0.23	29.56	0.0018	0.37	0.72	48.66
MOMENTFM	0.23	29.88	0.0019	0.36	0.64	48.87
QWEN	0.24	31.03	0.0023	0.33	0.65	50.05
SHAPELET	0.26	29.64	0.0028	0.37	0.72	46.50
DTAIDISTANCE	0.21	25.23	**0.0017**	0.37	0.80	**34.97**
STUMPY	0.20	28.41	0.0019	**0.44**	0.82	48.17
ROSER	**0.16**	**22.98**	0.0021	0.39	**0.96**	37.00

unsupervised proxies: *distributional similarity* measures global statistical alignment between retrieved and reference data; *temporal dynamics* measures preservation of realistic temporal structure and trajectory evolution; and *density and diversity* measure the compactness of retrieved segments and their spread in feature space. To quantify these properties, we employ a set of low-level alignment metrics such us, Wasserstein Distance (WD), DTW Nearest Neighbor (DTW NN), Spectral Wasserstein Distance (Spectral WD), and high-level behavioral consistency metrics like, Temporal Correlation, Density, and Diversity, capturing both precise trajectory alignment and task-relevant semantic structure. A detailed mathematical formulation for each metric is provided in Appendix A

4.2 RESULTS

Which representation families best support robotic sequence retrieval across domains? To understand which representation paradigms are most suitable for robotic sequence retrieval, we compare ROSER against (i) LLM-based embeddings (Llama, Gemma, Qwen), (ii) a time-series foundation model (MomentFM), and (iii) classical time-series matching methods (STUMPY, Dtaidistance, Shapelets) across LIBERO, nuScenes, and DROID. Tables 1–3 reveal a consistent pattern: methods that operate directly on time-series structure dominate large language embeddings on both distributional and temporal criteria. In particular, ROSER achieves the most favorable overall profile (best or second-best across nearly all metrics), indicating that a task-adaptive metric space learned from a small support set better captures the task-relevant kinematics than frozen text embeddings. Classical matching baselines remain competitive on specific axes (e.g., strong temporal correlation or spectral alignment), but their performance is less consistent across tasks and datasets, suggesting limited robustness to execution variability and feature relevance. MomentFM generally improves over LLM embeddings, yet still trails the top time-series baselines, supporting the hypothesis that retrieval benefits from representations that are tuned (via episodic training) to the task manifold rather than only pretrained generically.

When do classical matching methods fail, and what does ROSER retrieve instead? To probe retrieval failure modes, we compare qualitative and quantitative behavior across tasks with varying execution variability. We find that classical matching approaches (e.g., motif-based STUMPY and

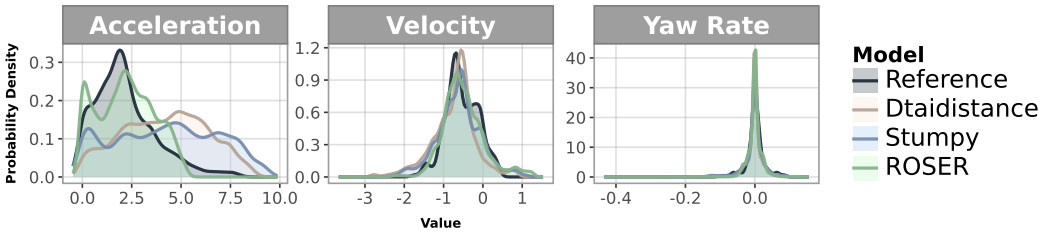

Figure 2: Feature-level distribution visualization for task "regular stop" in the nuScenes benchmark.

DTW-based Dtaidistance) are reliable for highly stereotyped segments, but degrade on tasks where the same semantic maneuver admits multiple valid kinematic realizations (speed profiles, timing, or joint-space detours). In these cases, ROSER is more likely to retrieve behaviorally consistent segments that match the task's defining dynamics rather than superficial pointwise similarity. For example, LIBERO "bottom drawer open" requires collision-avoiding arm motion; ROSER retrieves trajectories that preserve the longer reach-and-avoid structure, while strong baselines often drift to visually similar but semantically different pick-and-place behaviors (Figure 17). We report qualitative results in detail for all tasks in Appendix C.

This effect is also visible in nuScenes. Table 2 shows that time-series baselines remain competitive on simple maneuvers, but ROSER is substantially more robust for stops and turns where velocity/acceleration/yaw-rate magnitudes and timing vary across scenes. Because ROSER's metric is shaped by episodic training, it can implicitly reweight features and temporal patterns that are most diagnostic for the queried maneuver. Figure 2 illustrates this: ROSER retrieves "regular stop" segments whose acceleration distribution aligns closely with the reference, while STUMPY and Dtaidistance tend to mix in non-stopping segments that share only partial low-level similarity.

How many reference examples are effective for Few-Shot learning? To evaluate the sample efficiency of our few-shot learning approach, we performed an ablation study on the number of reference examples (K-shot) used to train the encoder and compute the task prototypes. As detailed in Table 4, our model demonstrates a high degree of data efficiency, maintaining competitive retrieval performance even when the target set is reduced to 3 or 5 examples. While the best distributional similarity, marked by a Wasserstein Distance of 0.086 on LIBERO and 0.27 on nuScenes, is achieved using 10 reference samples, the performance degradation observed at 7 and 5 samples is minimal. Specifically, the Temporal Correlation and DTW NN metrics remain remarkably stable across all datasets, indicating that the Prototypical Network can reliably define the task manifold even with a sparse support set. However, a slight decline in performance at $K = 3$ (e.g., WD increasing to 0.22 in DROID) suggests that a minimum of 5 to 7 samples is the "sweet spot" for balancing minimal human labeling effort with high retrieval accuracy. These findings confirm that our approach effectively bypasses the need for large-scale expert demonstrations, enabling the definition of complex robotic behaviors from a handful of random observations. While more data marginally improves the prototype's stability, the marginal utility of adding samples beyond $K = 10$ is expected to diminish, reinforcing the effectiveness of our few-shot formulation for rapid task adaptation.

What is the quality-latency trade-off across methods? Table 5 reports average per-match retrieval time across datasets. LLM-based embeddings are orders of magnitude slower per comparison, making exhaustive sliding-window retrieval impractical at scale. Classical time-series matching (Dtaidistance, STUMPY) is substantially faster, but still incurs non-trivial per-match cost when the candidate set is large. In contrast, ROSER achieves consistently low latency (sub-millisecond per-match on LIBERO and similarly low overhead on nuScenes and DROID) while also delivering the strongest retrieval quality in Tables 1–3. This combination suggests that a lightweight encoder with a learned metric is a practical design point for large-scale log mining: it enables high-throughput search without relying on heavyweight foundation models.

What is the trade-off between Diversity and Distributional Similarity? Figure 3 illustrates a consistent positive correlation between intra-class distance (diversity) and Wasserstein distance across datasets, indicating an inherent trade-off between distributional similarity and behavioral diversity.

Table 4: ROSER performance when number of target samples is varied. We train ROSER in K-3 to K-10 shot manner. Performance degradation is insignificant when we lower K, indicating the robustness of the encoder's ability to learn discriminative features from a few samples.

Dataset	K-Shot	WD (↓)	DTW NN (↓)	Spectral WD (↓)	Temp Corr. (↑)	Density (↑)	Diversity (↓)
Libero	10	**0.086**	**6.57**	**0.0019**	**0.72**	**0.69**	8.8
	7	0.10	6.84	0.0022	0.71	0.59	**8.38**
	5	0.12	8.41	0.0026	0.68	0.49	9.61
	3	0.11	7.06	0.0025	0.69	0.55	8.72
nuScenes	10	**0.27**	12.84	0.00056	0.53	1.30	26.69
	7	0.35	**12.62**	**0.00050**	**0.54**	1.24	23.65
	5	0.34	12.73	0.00070	**0.54**	1.27	24.24
	3	0.36	12.92	0.00070	0.49	1.27	**22.68**
DROID	10	**0.16**	**22.98**	0.0021	0.39	**0.96**	37.00
	7	**0.16**	23.96	**0.0019**	0.37	0.89	**36.99**
	5	0.17	23.20	0.0020	**0.40**	0.88	37.50
	3	0.22	25.18	0.0022	0.38	0.72	37.75

Table 5: Average retrieval time per match (seconds) across datasets. Each entry reports the mean wall-clock time required to perform a single query–candidate matching operation, measured per offline demonstration and averaged over all retrievals.

Model	LIBERO (↓)	nuScenes (↓)	DROID (↓)
Gemma	0.1558	0.1348	0.1347
LLaMA	0.8566	0.7929	0.7919
MomentFM	0.1088	0.2092	0.1040
Qwen	0.4449	0.4225	0.4176
Shapelet	0.002526	**0.001717**	**0.001617**
DTAIDistance	0.09030	0.1805	0.1770
Stumpy	0.009833	0.01999	0.01818
ROSER	**0.0005163**	0.003873	0.004549

On DROID, the relationship is moderate (Spearman $\rho = 0.51$, Pearson $r = 0.63$, Kendall $\tau = 0.42$), while LIBERO shows the strongest coupling ($\rho = 0.65$, $r = 0.76$, $\tau = 0.47$), suggesting that in manipulation tasks, closer alignment to the target data distribution is often accompanied by reduced variability among retrieved trajectories. NuScenes exhibits a weaker but still reliable association ($\rho = 0.48$, $r = 0.69$, $\tau = 0.36$), consistent with the higher intrinsic multimodality of navigation behaviors where multiple diverse trajectories may remain distributionally valid. This relationship between diversity and distributional similarity is consistent with previous work in generative model evaluation that sees both as related but separate goals, which often exhibits a trade-off (Naeem et al., 2020; Alaa et al., 2022; Khayatkhoei & AbdAlmageed, 2023). We report the results of our other experiments in Appendix B.

5 Conclusion

In this work, we addressed a critical bottleneck in robot learning: the structural incompatibility between how robotic data is collected and how modern learning algorithms consume it. By re-framing data curation as a few-shot retrieval problem, we demonstrated that vast repositories of underutilized robotic logs can be unlocked with minimal supervisio, requiring only a handful of reference examples rather than exhaustive human annotation. Our contributions are both methodological and empirical. We formalized robotic sequence retrieval as a principled task, introduced ROSER as a lightweight few-shot solution operating in learned metric spaces, and established comprehensive evaluation protocols spanning distributional similarity, temporal dynamics, and diversity-precision trade-offs. Through extensive benchmarking on LIBERO, DROID, and nuScenes, covering manipulation and autonomous driving at scale, we showed that ROSER consistently outperforms classical time-series

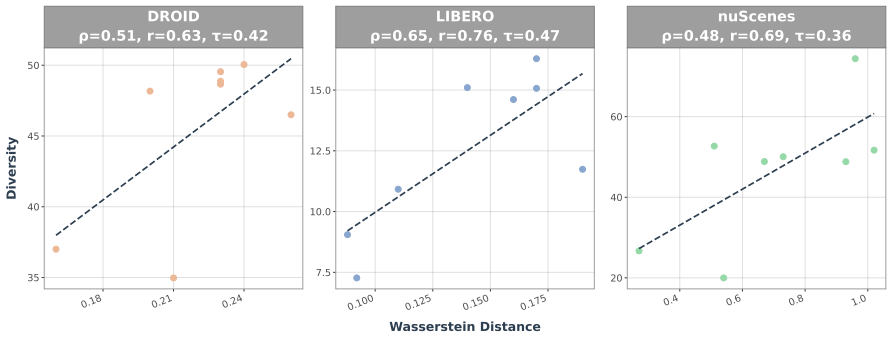

Figure 3: Relationship between Wasserstein Distance and Intra-class distance (diversity). Each point corresponds to a single retrieval model evaluated on the dataset. For each dataset, we report Spearman's rank correlation (ρ), Pearson correlation (r), and Kendall Tau (τ).

methods, learned embedding approaches, and large language model baselines in both accuracy and efficiency. These results have immediate practical implications. ROSER enables researchers to rapidly curate training data for novel tasks by providing few demonstrations, facilitates cross-dataset transfer by identifying analogous behaviors across different embodiments and environments, and supports continual learning by efficiently indexing new data as it becomes available. With sub-millisecond per-match inference and no task-specific training required, ROSER provides a scalable pathway to convert raw robot logs into structured, reusable datasets suitable for imitation learning, policy evaluation, and foundation model pretraining. Beyond its immediate applications, this work establishes robotic sequence retrieval as a fundamental research problem deserving of sustained attention. As the community continues to scale data collection, through initiatives like the Open X-Embodiment (Team & et al., 2023), the ability to efficiently organize, retrieve, and repurpose this data will become increasingly critical. Our benchmark, evaluation protocols, and baseline comparisons provide a foundation for future work in this direction.

5.1 Limitations and Future Work

Ambiguous transitions and multi-modal behaviors. Retrieval is most challenging when a target maneuver occurs as a brief transition inside a longer trajectory or when multiple behaviors share similar low-level dynamics (e.g., short decelerations that are not true stops). In these cases, classical matching methods may over-match based on local similarity, while ROSER can still produce false positives when the support set under-specifies the task.

Feature dropouts and domain shift. ROSER is sensitive to missing or corrupted proprioceptive channels and to shifts in sensor scaling across datasets. Our feature ablations (Appendix B) partially quantify this effect; mitigating strategies include per-feature normalization, hard-negative mining, and incorporating additional modalities (e.g., vision) when available.

Future Work. While ROSER demonstrates strong performance on proprioceptive time series, several promising directions remain unexplored. For instance, incorporating visual observations would enable retrieval based on semantic scene content, complementing the kinematic matching our current approach provides. We are actively developing multimodal extensions that fuse trajectory embeddings with vision-language models (Radford et al., 2021b) to support this richer query space. Finally, by continuously identifying underrepresented behaviors in existing datasets, such systems could prioritize data collection for the most informative regions of the task space, accelerating the path toward generalist robot learning.

REFERENCES

Ahmed M. Alaa, Boris van Breugel, Evgeny Saveliev, and Mihaela van der Schaar. How faithful is your synthetic data? sample-level metrics for evaluating and auditing generative models, 2022. URL https://arxiv.org/abs/2102.08921.

Brenna D. Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, 2009.

Yauhen Babakhin, Radek Osmulski, Ronay Ak, Gabriel Moreira, Mengyao Xu, Benedikt Schifferer, Bo Liu, and Even Oldridge. Llama-embed-nemotron-8b: A universal text embedding model for multilingual and cross-lingual tasks, 2025. URL https://arxiv.org/abs/2511.07025.

Giovanni Braglia, Davide Tebaldi, André Eugenio Lazzaretti, and Luigi Biagiotti. Arc-length-based warping for robot skill synthesis from multiple demonstrations. arXiv preprint, 2024. Temporal alignment for demonstration variability.

Anthony Brohan et al. Rt-x: Generalizable robotic skill learning from heterogeneous datasets. *arXiv preprint arXiv:2303.17580*, 2023.

Holger Caesar, Varun Bankiti, Alex Lang, et al. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Ching Chang, Wen-Chih Peng, and Tien-Fu Chen. Llm4ts: Two-stage fine-tuning for time-series forecasting with pre-trained llms. *ArXiv*, abs/2308.08469, 2023. URL https://api.semanticscholar.org/CorpusID:272980553.

Ishan Dave, Rohit Gupta, Mamshad Nayeem Rizve, and Mubarak Shah. Tclr: Temporal contrastive learning for video representation. In *arXiv preprint arXiv:2101.07974*, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Frederik Ebert, Yanlai Yang, Karl Schmeckpeper, Bernadette Bucher, Georgios Georgakis, Kostas Daniilidis, Chelsea Finn, and Sergey Levine. Bridge Data: Boosting Generalization of Robotic Skills with Cross-Domain Datasets. In *Proceedings of Robotics: Science and Systems (RSS) XVIII*, New York City, NY, USA, 2022. doi: 10.15607/RSS.2022.XVIII.063.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 1126–1135. JMLR.org, 2017.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: A family of open time-series foundation models, 2024. URL https://arxiv.org/abs/2402.03885.

Josif Grabocka, Nicolas Schilling, Martin Wistuba, and Lars Schmidt-Thieme. Learning time-series shapelets. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pp. 392–401, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450329569. doi: 10.1145/2623330.2623613. URL https://doi.org/10.1145/2623330.2623613.

Jan Hosang, Rodrigo Benenson, and Bernt Schiele. Learning non-maximum suppression, 2017. URL https://arxiv.org/abs/1705.02950.

Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4):917–963, March 2019. ISSN 1573-756X. doi: 10.1007/s10618-019-00619-1. URL http://dx.doi.org/10.1007/s10618-019-00619-1.

Mahyar Khayatkhoei and Wael AbdAlmageed. Emergent asymmetry of precision and recall for measuring fidelity and diversity of generative models in high dimensions, 2023. URL https://arxiv.org/abs/2306.09618.

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

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, 2015.

Sateesh Kumar, Shivin Dass, Georgios Pavlakos, and Roberto Martín-Martín. Collage: Adaptive fusion-based retrieval for augmented policy learning, 2025. URL https://arxiv.org/abs/2508.01131.

Sean M. Law. STUMPY: A Powerful and Scalable Python Library for Time Series Data Mining. *The Journal of Open Source Software*, 4(39):1504, 2019.

Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Bhaskar Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD)*, pp. 2–11. ACM, 2003. URL https://www.cs.ucr.edu/~eamonn/SAX.pdf.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *arXiv preprint arXiv:2306.03310*, 2023.

Stephanie Lowry, Niko Sünderhauf, Paul Newman, John Leonard, and David Cox. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 2015.

Ajay Mandlekar, Jonathan Booher, Max Spero, Albert Tung, Anchit Gupta, Yuke Zhu, Animesh Garg, Silvio Savarese, and Li Fei-Fei. Scaling robot supervision to hundreds of hours with roboturk: Robotic manipulation dataset through human reasoning and dexterity. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1048–1055. IEEE, 2019.

Carlo Masone and Barbara Caputo. A survey on deep visual place recognition. *IEEE Access*, 2021.

Wannes Meert, Kilian Hendrickx, Toon Van Craenendonck, Pieter Robberechts, Hendrik Blockeel, and Jesse Davis. DTAIDistance, August 2020. URL https://github.com/wannesm/dtaidistance.

Marius Memmel, Jacob Berg, Bingqing Chen, Abhishek Gupta, and Jonathan Francis. Strap: Robot sub-trajectory retrieval for augmented policy learning. In *Proceedings of ICLR*, 2025.

Cristian Meo, Mircea Lica, Zarif Ikram, Akihiro Nakano, Vedant Shah, Aniket Rajiv Didolkar, Dianbo Liu, Anirudh Goyal, and Justin Dauwels. Masked generative priors improve world models sequence modelling capabilities. *arXiv preprint arXiv:2410.07836*, 2024.

Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models, 2020. URL https://arxiv.org/abs/2002.09797.

Chihiro Noguchi and Toshihiro Tanizawa. Ego-vehicle action recognition based on semi-supervised contrastive learning. *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 5977–5987, 2023. URL https://api.semanticscholar.org/CorpusID:263795932.

Alec Radford, Jong Wook Kim, Chris Hallacy, and et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021a.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021b. URL https://arxiv.org/abs/2103.00020.

Travers Rhodes and Daniel D. Lee. Timewarpvae: Simultaneous time-warping and representation learning of trajectories, 2024. URL https://arxiv.org/abs/2310.16027.

Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, 1978.

Rui Shao, Wei Li, Lingsen Zhang, et al. Large vlm-based vision-language-action models for robotic manipulation: A survey. *arXiv preprint*, 2025.

Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, 2017.

Flood Sung et al. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Tao Tang, Dafeng Wei, Zhengyu Jia, Tian Gao, Changwei Cai, Chengkai Hou, Peng Jia, Kun Zhan, Haiyang Sun, Jingchen Fan, Yixing Zhao, Xiaodan Liang, Xianpeng Lang, and Yang Wang. Bev-tsr: Text-scene retrieval in bev space for autonomous driving. In *AAAI Conference on Artificial Intelligence*, 2024. URL https://api.semanticscholar.org/CorpusId:266725357.

Gemma Team and et al. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Open X-Embodiment Collaboration Team and et al. Open X-Embodiment: Robotic learning datasets and RT-X models. https://arxiv.org/abs/2310.08864, 2023.

Daan Van Wesenbeeck, Aras Yurtman, Wannes Meert, and Hendrik Blockeel. Locomotif: Discovering time-warped motifs in time series, 2023. URL https://arxiv.org/abs/2311.17582.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

Koen Vellenga, H. Joe Steinhauer, Göran Falkman, and Tomas Björklund. Evaluation of video masked autoencoders' performance and uncertainty estimations for driver action and intention recognition. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 7414–7422, 2024. doi: 10.1109/WACV57701.2024.00726.

Koen Vellenga, H. Joe Steinhauer, Jonas Andersson, and Anders Sjögren. Latent uncertainty representations for video-based driver action and intention recognition, 2025. URL https://arxiv.org/abs/2510.05006.

Yilin Wang. Driving behavior recognition via self-discovery learning. *ArXiv*, abs/2503.14194, 2025. URL https://api.semanticscholar.org/CorpusID:277103531.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Lexiang Ye and Eamonn Keogh. Time series shapelets: A new primitive for data mining. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009.

Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. Matrix profile i: All pairs similarity joins for time series: A unifying view that includes motifs, discords and shapelets. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 1317–1322, 2016. doi: 10.1109/ICDM.2016.0179.

Fanlong Zeng, Wensheng Gan, et al. Large language models for robotics: A survey. *arXiv preprint*, 2023.

Jiahui Zhu. Sttraj2vec: A spatio-temporal trajectory representation learning approach. *Knowledge-Based Systems*, 2024.

A APPENDIX

A.1 EVALUATION METRICS

Notation	Description
$\mathcal{R}, \mathcal{S}$	Offline, Retrieved and reference trajectory datasets (collections of tasks)
$\mathcal{R}^{(t)}, \mathcal{S}^{(t)}$	Trajectories for task t
$\mathbf{r}_r^{(t)}, \mathbf{s}_i^{(t)}$	Individual trajectories from task t of $\mathcal{R}^{(t)}$ or $\mathcal{S}^{(t)}$
$L_r^{(t)}, L_i{}^{(t)}$	Lengths of the r-th retrieved and i-th reference trajectories in task t
F	Number of features per time step
$r_{r,l,f}^{(t)}, s_{i,l,f}^{(t)}$	Feature f at time l of trajectory $\mathbf{r}_r^{(t)}$ or $\mathbf{s}_i^{(t)}$
$\mathbf{r}_{r,f}^{(t)}, \mathbf{s}_{i,f}^{(t)}$	Feature-wise temporal signals (vectors of length $L_r^{(t)}$ or $L_i^{(t)}$)
$\hat{P}_f^{\mathcal{X}}$	Mean normalized power spectral density for feature f over set $\mathcal{X}$
k	Number of nearest neighbors for local support radii
ρ_i	Distance to the k-th nearest neighbor in the reference set
W	Window length used to compare time series of unequal length
n	Number of sliding windows used in window-based metric computation
$\mathcal{W}(\cdot,\cdot)$	1-Wasserstein (Earth Mover's) distance
$\star$	Discrete cross-correlation operator
$\mathrm{DTW}(\cdot,\cdot)$	Multivariate Dynamic Time Warping distance
d	Arbitrary distance metric between vectors $\in \mathbb{R}^d$

Table 6: Notation used throughout the evaluation metrics

Note: In this appendix, we use $\mathcal{S}$ to denote the *reference* trajectory set for metric computation; this is distinct from the labeled meta-training set $\mathcal{S}$ used in the main method section.

The retrieved and reference datasets may contain trajectories of varying lengths. To enable a consistent and fair evaluation of retrieval quality across these time series, we employ the following window-based metric computation method.

Let the retrieved and reference datasets be collections of tasks:

$$\mathcal{R} = \{\mathcal{R}^{(t)}\}_{t\in\mathcal{T}}, \quad \mathcal{R}^{(t)} = \{\mathbf{r}_r^{(t)} \in \mathbb{R}^{L_r^{(t)}\times F}\}_{r=1}^{N_t},$$

$$\mathcal{S} = \{\mathcal{S}^{(t)}\}_{t\in\mathcal{T}}, \quad \mathcal{S}^{(t)} = \{\mathbf{s}_i^{(t)} \in \mathbb{R}^{L_i^{(t)}\times F}\}_{i=1}^{M_t}.$$

where $\mathcal{T}$ indexes tasks, $L_r^{(t)}, L_i^{(t)}$ are trajectory lengths, and F is the feature dimension.

Define the sliding-window size for each retrieved-reference pair:

$$W_{ri}^{(t)} = \min(L_r^{(t)}, L_i^{(t)}).$$

Extract overlapping windows of length $W_{ri}^{(t)}$ from the longer trajectory. Let $\mathbf{x}^{(t,k)}$ denote the k-th window of trajectory $\mathbf{x}$, and let the number of windows be

$$n = \max(L_r^{(t)}, L_i^{(t)}) - W_{ri}^{(t)} + 1.$$

Then, the metric between $\mathbf{r}_r^{(t)}$ and $\mathbf{s}_i^{(t)}$ is

$$\mathcal{M}^{(t)}(\mathbf{r}_r^{(t)}, \mathbf{s}_i^{(t)}) = \frac{1}{n}\sum_{k=1}^{n} \mathcal{M}\big(\mathbf{r}_r^{(t,k)}, \mathbf{s}_i^{(t,k)}\big),$$

where if the trajectories are equal length ($L_r^{(t)} = L_i^{(t)}$), then $n = 1$ and $\mathbf{r}_r^{(t,1)} = \mathbf{r}_r^{(t)}$, $\mathbf{s}_i^{(t,1)} = \mathbf{s}_i^{(t)}$.

A.1.1 Distributional Similarity

Distributional metrics assess whether the retrieved trajectories match the marginal feature distributions of the reference set, regardless of temporal ordering.

Wasserstein Distance. For task t, collect feature values across all trajectories and timesteps:

$$R_f^{(t)} = \{r_{r,l,f}^{(t)} \mid \mathbf{r}_r^{(t)} \in \mathcal{R}^{(t)},\, l = 1 \dots L_r^{(t)}\}$$

$$S_f^{(t)} = \{s_{i,l,f}^{(t)} \mid \mathbf{s}_i^{(t)} \in \mathcal{S}^{(t)},\, l = 1 \dots L_i^{(t)}\}$$

The 1D Wasserstein distance for feature f is computed as

$$\mathcal{W}(R_f^{(t)}, S_f^{(t)}) = \int_{-\infty}^{\infty} \left| F_{R_f^{(t)}}(x) - F_{S_f^{(t)}}(x) \right| dx \approx \sum_{i=1}^{n} \left| \mathrm{CDF}_{R_f^{(t)}}(x_i) - \mathrm{CDF}_{S_f^{(t)}}(x_i) \right| \Delta x_i$$

and the per-task feature-wise distance is

$$d(\mathcal{R}^{(t)}, \mathcal{S}^{(t)}) = \frac{1}{F} \sum_{f=1}^{F} \mathcal{W}(R_f^{(t)}, S_f^{(t)}).$$

A.1.2 Temporal Dynamics

Temporal metrics measure whether retrieved trajectories exhibit realistic evolution over time, capturing alignment, phase shifts, and sequential correlations.

Nearest-Neighbor Dynamic Time Warping (DTW). For task t, each retrieved trajectory is aligned to its most similar reference trajectory:

$$\mathrm{DTW}_{\mathrm{NN}}(\mathcal{R}^{(t)}, \mathcal{S}^{(t)}) = \frac{1}{|\mathcal{R}^{(t)}|} \sum_{\mathbf{r}_r^{(t)} \in \mathcal{R}^{(t)}} \min_{\mathbf{s}_i^{(t)} \in \mathcal{S}^{(t)}} \mathrm{DTW}(\mathbf{r}_r^{(t)}, \mathbf{s}_i^{(t)}).$$

Sliding windows are applied when $L_r^{(t)} \neq L_i^{(t)}$, averaging DTW over all windows.

Temporal Cross-Correlation (TCC). For task t, feature-wise signals are cross-correlated:

$$\mathrm{TCC}^{(t)} = \frac{1}{F\,|\mathcal{R}^{(t)}|\,|\mathcal{S}^{(t)}|} \sum_{f=1}^{F} \sum_{\mathbf{r}_r^{(t)} \in \mathcal{R}^{(t)}} \sum_{\mathbf{s}_i^{(t)} \in \mathcal{S}^{(t)}} \frac{\max_\tau (\mathbf{r}_{r,f}^{(t)} \star \mathbf{s}_{i,f}^{(t)})(\tau)}{\|\mathbf{r}_{r,f}^{(t)}\|_2\, \|\mathbf{s}_{i,f}^{(t)}\|_2}.$$

Cross-correlation is computed over windows if lengths differ.

Power Spectral Density (Wasserstein-based) Distance. To compare the frequency content of two datasets for task t, we define:

$$\mathrm{PSD}(\mathcal{R}^{(t)}, \mathcal{S}^{(t)}) = \frac{1}{F} \sum_{f=1}^{F} \mathcal{W}\big(\hat{P}_f^{\mathcal{R}^{(t)}}, \hat{P}_f^{\mathcal{S}^{(t)}}\big),$$

where the per-feature PSDs are:

$$\hat{P}_f^{\mathcal{R}^{(t)}} = \frac{1}{|\mathcal{R}^{(t)}|} \sum_{\mathbf{r} \in \mathcal{R}^{(t)}} \frac{\left|\mathrm{FFT}_W(\mathbf{r}_f)\right|^2}{\sum_{k=1}^{W} \left|\mathrm{FFT}_W(\mathbf{r}_f)[k]\right|^2}$$

$$\hat{P}_f^{\mathcal{S}^{(t)}} = \frac{1}{|\mathcal{S}^{(t)}|} \sum_{\mathbf{s} \in \mathcal{S}^{(t)}} \frac{\left|\mathrm{FFT}_W(\mathbf{s}_f)\right|^2}{\sum_{k=1}^{W} \left|\mathrm{FFT}_W(\mathbf{s}_f)[k]\right|^2}$$

FFT resolution determined by W.

A.1.3 Density and Diversity

Density measures how many retrieved trajectories lie near the reference set, while diversity captures how different retrieved trajectories are from one another.

k-NN-Based Density (Naeem et al., 2020). For task t, flatten trajectories or windowed subsequences:

$$\tilde{\mathbf{r}}_r^{(t)}, \tilde{\mathbf{s}}_i^{(t)} \in \mathbb{R}^{W \cdot F}, \quad W = \min(L_r^{(t)}, L_i^{(t)}).$$

Let $d(\cdot, \cdot)$ be a distance metric. Define the k-NN radius for each reference trajectory:

$$\rho_i^{(t)} := \text{distance to its } k\text{-th nearest neighbor in } \mathcal{S}^{(t)} \setminus \{\mathbf{s}_i^{(t)}\}.$$

Density is then defined as

$$\text{Density}^{(t)} = \frac{1}{|\mathcal{R}^{(t)}|} \sum_{i=1}^{|\mathcal{R}^{(t)}|} \frac{1}{k} \sum_{j=1}^{|\mathcal{S}^{(t)}|} \mathbf{1}\{d(\tilde{\mathbf{r}}_r^{(t)}, \tilde{\mathbf{s}}_i^{(t)}) \leq \rho_i^{(t)}\}.$$

Neighborhood size k can be used as a hyperparameter.

Intra-class Distance (ICD). Internal diversity for task t is measured via

$$\text{ICD}^{(t)} = \frac{2}{|\mathcal{R}^{(t)}|(|\mathcal{R}^{(t)}| - 1)} \sum_{i<j} \text{DTW}(\mathbf{r}_r^{(t)}, \mathbf{r}_i^{(t)}).$$

Sliding windows are applied if sequences differ in length.

A.2 Baseline Implementations

This appendix rigorously details the baseline retrieval methods evaluated in our experiments. For each benchmark, we fix a retrieval budget k, corresponding to the number of subsequences retrieved per task: k=150 for LIBERO, k=50 for nuScenes, and k=80 for DROID. Given a reference trajectory segment, each baseline retrieves the top-k most similar subsequences from an offline dataset. We group baselines into (i) LLM-based embedding methods, (ii) foundation time-series models, and (ii) classical matching-based time-series methods.

For a given task $t \in \mathcal{T}$, let's assume one reference sample from $\mathcal{S}$ is:

$$\mathbf{s}_i^{(t)} \in \mathbb{R}^{L_i^{(t)} \times F}$$

where L_i denotes the sequence length and F the number of features. Now, the offline dataset is

$$\mathcal{U} = \{\mathbf{u}_j \in \mathbb{R}^{L_j \times F}\}_{j=1}^{N}, \qquad L_j >> L_i^{(t)} \; \forall i,$$

from which fixed-length candidate subsequences are created using sliding window technique.

A.2.1 LLM-based Embedding Baselines

These baselines convert multivariate time-series subsequences into textual representations and apply pretrained language embedding models for similarity-based retrieval. The following embedding models were used as baselines for this paper:

- **Qwen3 Embedding (0.6B)** (Yang et al., 2025)
- **LLaMA Nemotron Embed (1B)** (Babakhin et al., 2025)
- **Gemma3 Embedding (0.3B)** (Team & et al., 2025)

Symbolic Encoding via SAX. Consider a candidate subsequence of length L, $u \in \mathcal{U}$.

Each feature dimension $f \in \{1, \ldots, F\}$ is independently normalized over time:

$$\tilde{u}_{l,f} = \frac{u_{l,f} - \mu_f}{\sigma_f}, \qquad l = 1, \ldots, L,$$

where

$$\mu_f = \frac{1}{L}\sum_{l=1}^{L} u_{l,f}, \quad \sigma_f^2 = \frac{1}{L}\sum_{l=1}^{L}(u_{l,f} - \mu_f)^2.$$

If $\sigma_f = 0$, we set $\tilde{x}_{l,f} = 0$ for all l.

The normalized sequence is discretized using a SAX-inspired symbolic encoding (Lin et al., 2003). Let B denote the number of bins (we use $B = 26$). Unlike classical SAX, which applies Piecewise Aggregate Approximation and fixed Gaussian breakpoints, we discretize each subsequence using per-subsequence empirical quantiles:

$$\mathcal{E}_f = \left\{ q_f\left(\frac{k}{B}\right) \right\}_{k=0}^{B},$$

where $q_f(\cdot)$ is the quantile function of $\{\tilde{u}_{l,f}\}_{l=1}^{L}$.

Each value $\tilde{u}_{l,f}$ is mapped to a symbol

$$s_{l,f} \in \mathcal{A} = \{\mathtt{A}, \mathtt{B}, \ldots, \mathtt{Z}\}, \quad s_{l,f} = \mathcal{A}_k \;\text{ if }\; \tilde{u}_{l,f} \in [\mathcal{E}_{f,k}, \mathcal{E}_{f,k+1}), \quad k \in \{0, \ldots, 25\}.$$

This yields, for each feature f, an ordered symbolic string $\{s_{l,f}\}_{l=1}^{L}$. The final textual representation concatenates the F feature-wise strings while preserving temporal order.

Prompt Template for LLM-based Embedding. Each multivariate time series is converted into a structured textual prompt after SAX discretization. For each feature dimension $f \in \{1, \ldots, F\}$, an ordered symbolic string

$$\mathbf{s}_f = \{s_{l,f}\}_{l=1}^{L}$$

is produced and formatted as a single line in the prompt.

The resulting prompt template was used:

<feature 1> [SAX Representation] : $\mathbf{s}_1$
<feature 2> [SAX Representation] : $\mathbf{s}_2$
$\vdots$
<feature F> [SAX Representation] : $\mathbf{s}_F$

where "<feature f>" is the f-th feature of the multivariate time series.

LLM Inference. If we use a sliding window of size W, one candidate window from an unlabeled sample u is $u_{l:l+W}$ where l is the initial index and $l < L_j$.

For simplicity, from now on, let's assume, each reference trajectory $x = \mathbf{s}_i^{(t)}$. We define the reference embedding representation as

$$\mathbf{z_x} = f_{\text{LLM}}(\text{Prompt}(x)) \in \mathbb{R}^E,$$

which encodes the aggregated textual representation of the reference.

Similarly, for simplicity, let's assume each candidate window is $y = u_{l:l+W}$. It's embedding is:

$$\mathbf{z}_y = f_{\text{LLM}}(\text{Prompt}(y)) \in \mathbb{R}^E.$$

Similarity is measured using cosine distance:

$$d_l = 1 - \frac{\mathbf{z_x}^\top \mathbf{z}_y}{\|\mathbf{z_x}\|_2 \, \|\mathbf{z}_y\|_2}.$$

We select the top-k candidates with lowest d_l.

A.2.2 MOMENT-FM.

MOMENT (Goswami et al., 2024) is a pretrained foundation model for multivariate time-series representation learning. The reference x is embedded once:

$$\mathbf{z}_x = f_{\text{MOMENT}}(x),$$

and each candidate window y is embedded independently.

$$\mathbf{z}_y = f_{\text{MOMENT}}(y).$$

Retrieval is performed using cosine similarity,

$$d_l = 1 - \frac{\mathbf{z_x}^\top \mathbf{z}_y}{\|\mathbf{z_x}\|_2 \, \|\mathbf{z}_y\|_2},$$

We select the top-k candidates with lowest d_l.

A.2.3 MATCHING-BASED TIME-SERIES BASELINES.

STUMPY (Matrix Profile Matching). STUMPY computes the matrix profile, which stores the minimum z-normalized Euclidean distance between a query and all subsequences of equal length (Yeh et al., 2016; Law, 2019). For multivariate inputs, we compute the z-normalized Euclidean distance jointly across all feature dimensions. We use the implementation [1] from (Law, 2019).

For a reference trajectory x and a candidate full trajectory y, the distance is

$$d_i = \left\| \frac{x - \boldsymbol{\mu}_x}{\boldsymbol{\sigma}_x} - \frac{y - \boldsymbol{\mu}_y}{\boldsymbol{\sigma}_y} \right\|_2,$$

where $\boldsymbol{\mu}_x$ and $\boldsymbol{\sigma}_x$ are the mean and standard deviation of x along the feature dimension, and $\boldsymbol{\mu}_y$ and $\boldsymbol{\sigma}_y$ are the mean and standard deviation of y along the feature dimension.

We select the top-k candidates with lowest d_l.

Dynamic Time Warping (DTW). DTW aligns sequences under non-linear temporal distortions (Sakoe & Chiba, 1978). We use the implementation [2] from (Meert et al., 2020). For a reference trajectory segment x and a candidate window y, DTW computes

$$d_l = \text{DTW}(x, y) = \min_{\pi} \sum_{(n,m)\in\pi} \|x_n - y_m\|_2,$$

where $n = 1 \dots L_i^{(t)}$, $m = 1 \dots L_j$, and π is a valid warping path.

We select the top-k candidates with lowest d_l.

Shapelet-based Matching. Shapelets are discriminative subsequences originally optimized for time-series classification (Ye & Keogh, 2009). We use a pretrained shapelet-based classifier to score fixed-length candidate windows. Each window y is evaluated using the classifier's predicted class probabilities

$$p_l = \max_c P(c \mid y),$$

which serves as a confidence-based retrieval score. We define the retrieval cost as

$$d_l = -p_l,$$

We select the top-k candidates with lowest d_l.

[1] https://stumpy.readthedocs.io/en/latest/Tutorial_Pattern_Matching.html
[2] https://dtaidistance.readthedocs.io/en/latest/usage/subsequence.html

A.3 SEQUENCE ENCODER

The embedding function f_θ is implemented as a 1D Convolutional Neural Network (CNN) with L sequential convolutional blocks. Each block consists of a 1D convolution, optional batch normalization, and a non-linear activation (ReLU). For an input sequence $\mathbf{X} \in \mathbb{R}^{L \times F}$, the network computes a sequence of hidden representations through N_l blocks:

$$\mathbf{h}_1 = \text{ReLU}(\text{BN}(\text{Conv}_{1\text{D}}(\mathbf{X}))) \tag{4}$$

$$\mathbf{h}_l = \text{ReLU}(\text{BN}(\text{Conv}_{1\text{D}}(\mathbf{h}_{l-1}))) \quad \text{for } l = 2, \ldots, N_l \tag{5}$$

To aggregate these local temporal features into a fixed-size vector representation, we apply Global Average Pooling (GAP) over the temporal dimension. To handle sequences of variable length, a binary mask is applied so that padded timesteps do not affect the global-pooled embedding.

$$\mathbf{z} = f_\theta(\mathbf{X}) = \frac{1}{L'} \sum_{t=1}^{L'} \mathbf{h}_{N_l} \in \mathbb{R}^{d_{emb}} \tag{6}$$

where L' is the temporal resolution of the final feature map and average is computed only over valid (non-padded) timesteps. This ensures invariance to sequence length while preserving salient temporal patterns.

A.4 DATASETS

We explain all 3 datasets we used in this study in details here.

LIBERO (Liu et al., 2023): LIBERO (LIfelong learning BEchmark on RObot manipulation) is a simulation-based benchmark for studying lifelong learning, knowledge transfer, and multi-task policy generalization in robot manipulation. We focus on a subset of fundamental tasks, grouped as: *Bottom Drawer Open/Close, Top Drawer Open/Close, Microwave Open/Close, Stove On/Off, and Pick-and-Place*. The benchmark contains 130 language-conditioned tasks organized into suites, including LIBERO-90 for pretraining and LIBERO-10 for downstream evaluation. High-quality human teleoperation demonstrations are provided for all tasks, enabling sample-efficient policy training and rigorous comparison of continual learning approaches. Dataset contains end effector (EE) pose, gripper states, robot joint states and environment videos. We use EE pose, gripper states, and joint states in this study.

DROID (Khazatsky et al., 2024): DROID (Distributed Robot Interaction Dataset) is a large-scale real-world robot manipulation dataset collected across hundreds of diverse indoor environments. We evaluate fundamental manipulation tasks, including: *Close/Open Cabinet, Close Drawer, Pick-and-Place, and Turn Faucet/Knobs*. The dataset contains tens of thousands of human teleoperation demonstration trajectories with synchronized multi-view RGB video streams, depth, robot proprioceptive measurements (joint positions and velocities), end-effector poses, and optional natural language annotations. DROID provides extensive real-world variation and metadata, supporting training and evaluation of policies that generalize across environments and tasks, and includes released hardware details and code for reproducible data collection and policy development. In this study, we use EE pose, gripper states and joint states similar as LIBERO. We randomly sample around 500 relevant demonstrations using the provided language instructions for this study.

nuScenes (Caesar et al., 2020): nuScenes is a large-scale multimodal dataset for autonomous driving research, collected using a full sensor suite on autonomous vehicles. We focus on fundamental driving maneuvers, including: *Left Turn, Right Turn, Straight Driving, and Regular Stop*. The dataset provides synchronized data from six cameras, five radars, and one LiDAR sensor, offering 360° coverage of each scene. Each scene spans 20 seconds and is densely annotated with 3D bounding boxes for 23 object classes and attributes, along with object trajectories, motion information, and contextual metadata. In this study, we only use dynamics data like forward velocity, forward acceleration, and yaw rate.

B ABLATION STUDIES

In this section, we explain in details the rest of our experiments to validate ROSER. We ablated the number of retrieved top-k and the proprioceptive features.

Table 7: ROSER Metrics for Varying Top-k. We retrieve a range of samples for each task. Performance is better when the retrieval number is small. Our assumption is tasks are imbalanced and retrieving more samples may include false positives.

DATASET	TOP-k	WD (↓)	DTW NN (↓)	SPECTRAL WD (↓)	TEMP CORR. (↑)	DENSITY (↑)	DIVERSITY (↓)
LIBERO	50	**0.080**	**4.96**	**0.0017**	**0.78**	**0.82**	**6.15**
	70	0.081	5.27	0.0018	0.77	0.77	6.75
	90	0.083	5.70	0.0019	0.75	0.71	7.46
	110	0.085	6.09	0.0019	0.74	0.67	8.11
	130	0.086	6.36	0.0020	0.73	0.64	8.62
	150	0.086	6.57	0.0019	0.72	0.69	8.8
NUSCENES	20	**0.25**	**11.29**	0.00052	**0.57**	**1.44**	**22.22**
	30	**0.25**	11.54	**0.00050**	0.56	1.40	23.68
	40	0.27	12.40	0.00054	0.54	1.32	25.60
	50	0.27	12.84	0.00056	0.53	1.30	26.69
DROID	40	**0.16**	22.81	**0.0020**	0.39	**0.98**	36.85
	50	**0.16**	**22.74**	**0.0020**	**0.40**	0.97	**36.64**
	60	**0.16**	22.87	0.0021	**0.40**	0.97	36.74
	70	**0.16**	23.11	0.0021	0.39	0.95	37.03
	80	**0.16**	22.98	0.0021	0.39	0.96	37.00

What is the impact of the retrieval threshold top-(k) on the quality of the retrieved data distribution? We study the effect of the retrieval threshold k by sweeping the number of retrieved trajectories on LIBERO, nuScenes, and DROID datasets. Trajectories are ranked by Euclidean distance in the learned embedding space to task prototypes, and the top-k samples are selected. Table 7 reveals a consistent trade-off between distributional similarity, temporal alignment, and set-level properties. Across all datasets, smaller k yields stronger *distributional similarity*, as indicated by lower Wasserstein Distance (WD). At the same time, *temporal alignment* is highest at low k, reflected by lower DTW Nearest Neighbor distance, lower Spectral WD, and higher Temporal Correlation. In contrast, increasing k improves the *set-level properties* of the retrieved collection: Density increases and Diversity improves (lower ICD), indicating broader coverage of task variations. Concretely, the best distributional similarity and temporal alignment are achieved at k=50 for Libero, k=20 for nuScenes, and k=40 for DROID. Beyond these points, WD, DTW NN, Spectral WD, and Temporal Correlation degrade monotonically, while Density and Diversity improve. These results show that Euclidean distance to the task prototype provides a reliable ranking of trajectory relevance, and that low-k retrieval is preferable when preserving distributional similarity and temporal structure is critical, whereas larger k favors coverage and diversity.

Which sensory features are critical for cross-dataset task retrieval, and how do feature dependencies vary between manipulation and navigation domains? To identify the most informative features for task-relevant retrieval, we conducted a feature ablation study across all datasets, removing one category of sensory input at a time. As shown in Table 8, the manipulation-heavy datasets, LIBERO and DROID, exhibit remarkably similar behavior, where joint states and end-effector (EE) position serve as the primary pillars of the task manifold. For both datasets, the exclusion of Joint States led to the most substantial performance drops in DTW NN and distributional Density, confirming that the high-dimensional articulation of the arm is essential for disambiguating complex manipulation tasks. Besides, the exclusion of soint states and end-effector position led to the substantial drop in Wasserstein Distance. Interestingly, the removal of Joint States also led to an artificial inflation of the Diversity metric (e.g., from 8.8 to 12.13 in LIBERO), suggesting that without joint information, the retrieval results become more chaotic and less consistent with the prototype. In contrast, for the nuScenes navigation dataset, the model is most sensitive to Velocity, reflecting the fundamental importance of speed profiles in characterizing driving behaviors like turns or braking. Across all domains, we find that while the model is robust to the removal of secondary features like gripper states or yaw rate, the "kinematic backbone", joint states for robots and velocity for vehicles—is indispensable. This cross-dataset consistency between LIBERO and DROID validates that ROSER successfully identifies a universal manipulation embedding that prioritizes arm configuration over binary gripper states.

Table 8: ROSER performance when one of the features is removed. For manipulation tasks, end-effector pose and joint states are the most important. For navigation, velocity and acceleration play major roles.

Dataset	Removed Feature	WD (↓)	DTW NN (↓)	Spectral WD (↓)	Temp Corr. (↑)	Density (↑)	Diversity (↓)
Libero	N/A	**0.086**	**6.57**	**0.0019**	**0.72**	**0.69**	**8.8**
	EE Pos	0.12	7.9	0.0023	0.67	0.51	9.2
	Gripper States	0.10	7.52	0.0023	0.67	0.56	9.12
	Joint States	0.12	9.28	0.0022	0.69	0.53	12.13
nuScenes	N/A	**0.27**	**12.84**	0.00056	0.53	1.30	26.69
	Velocity	0.41	17.22	**0.0005**	**0.54**	0.95	40.7
	Acceleration	0.35	17.43	0.0006	0.48	1.14	32.72
	Yaw Rate	0.28	13.76	0.0007	0.43	**1.34**	**26.5**
DROID	N/A	**0.16**	22.98	0.0021	0.39	**0.96**	**37.00**
	EE Pos	0.21	24.49	0.0021	0.39	0.74	42.15
	Gripper States	**0.16**	**22.93**	**0.0019**	**0.40**	0.86	38.28
	Joint States	0.22	24.89	0.0025	0.39	0.76	38.42

C Distribution and Qualitative Results

This appendix reports qualitative results for all tasks evaluated across the benchmarks. For each task or maneuver, we present representative demonstrations together with feature-level distribution plots comparing the top-performing methods: Dtaidistance, STUMP, and ROSER (ours).

C.1 LIBERO Benchmark

C.1.1 Microwave Open

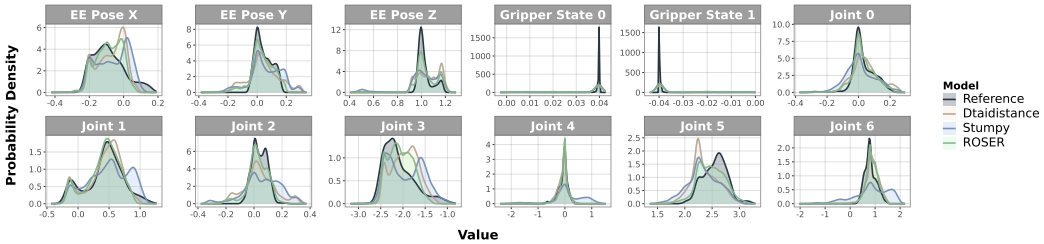

Figure 4: Feature-level distribution visualization for microwave open task in the LIBERO benchmark.

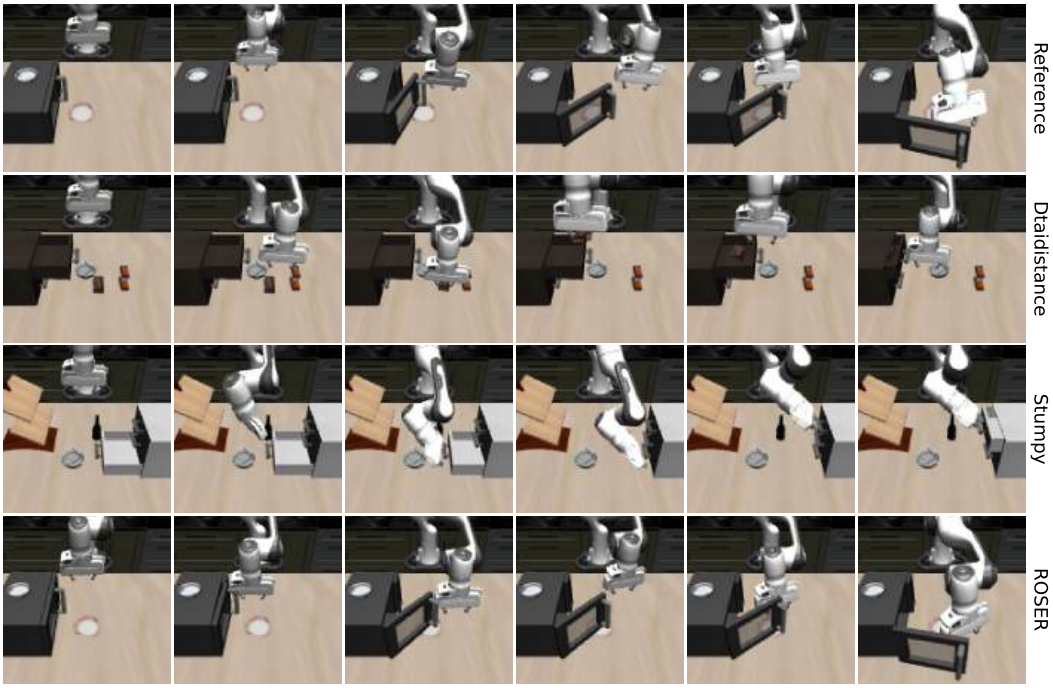

Figure 5: LIBERO qualitative results for microwave open task. ROSER retrieves correct sample while Dtaidistance and Stumpy retrieves drawer close task

C.1.2 MICROWAVE CLOSE

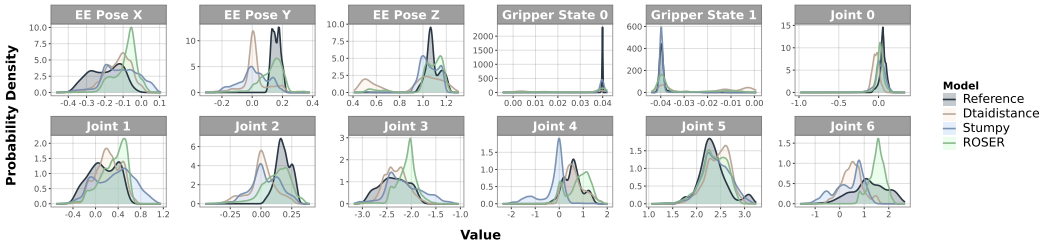

Figure 6: Feature-level distribution visualization for microwave close task in the LIBERO benchmark.

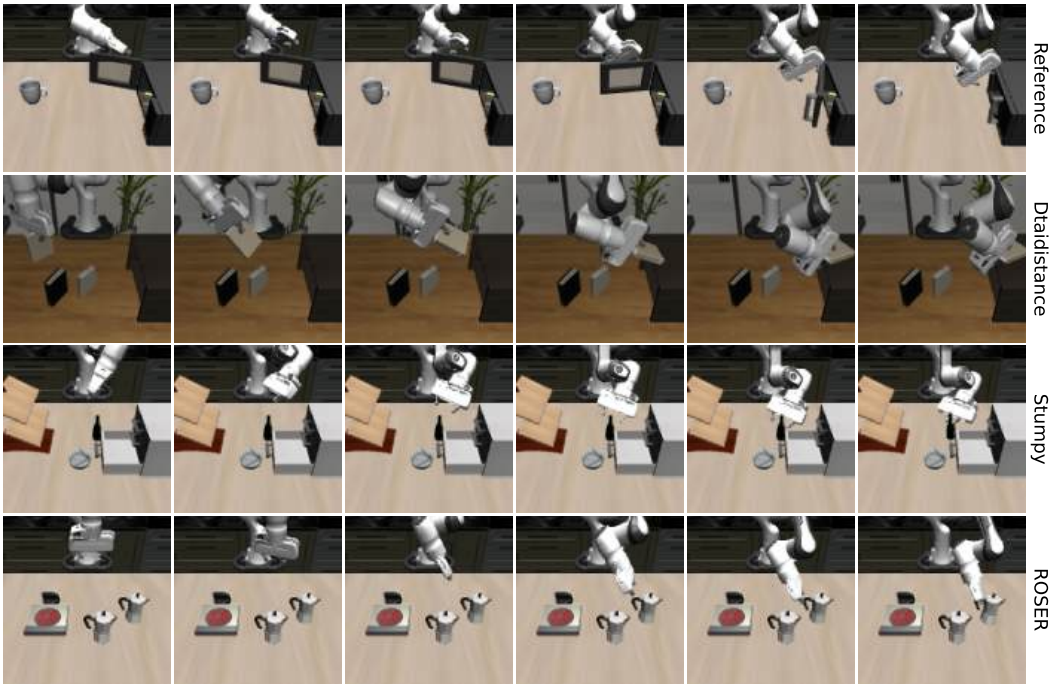

Figure 7: LIBERO qualitative results for microwave close task. All the models fail to retrieve correct tasks on this.

C.1.3 STOVE ON

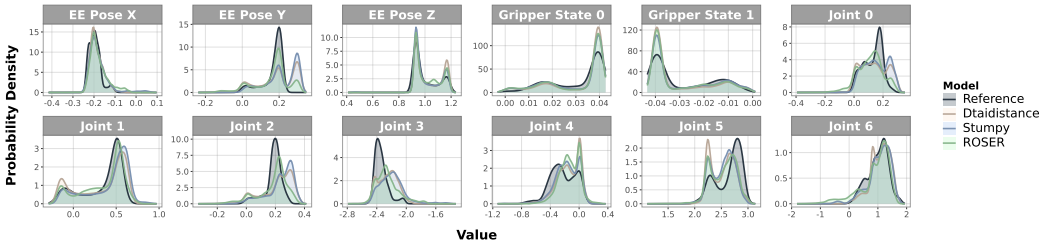

Figure 8: Feature-level distribution visualization for stove on task in the LIBERO benchmark.

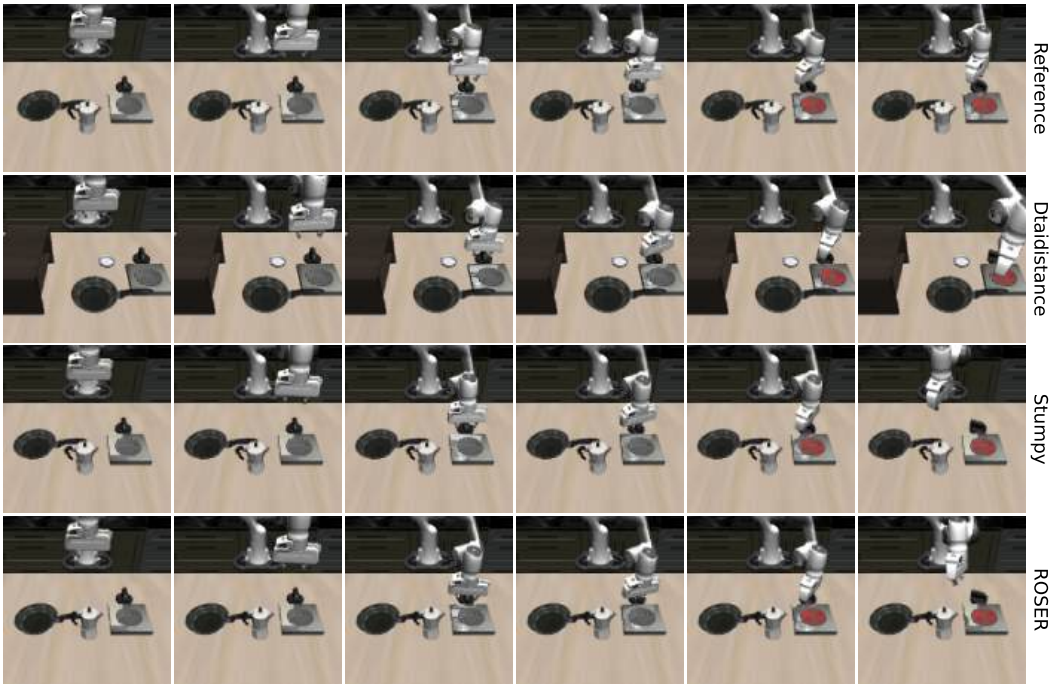

Figure 9: LIBERO qualitative results for stove on task. All models successfully retrieve samples for this task.

C.1.4 STOVE OFF

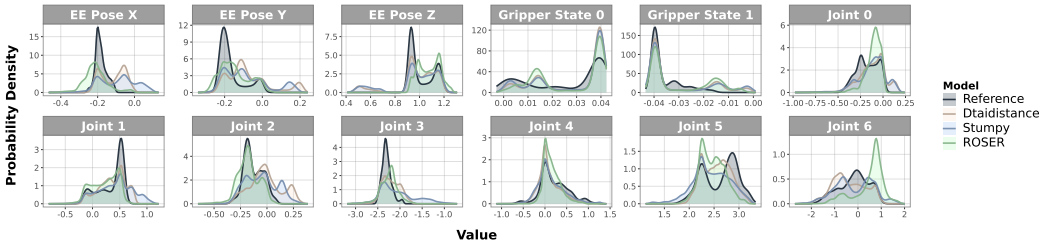

Figure 10: Feature-level distribution visualization for stove off task in the LIBERO benchmark.

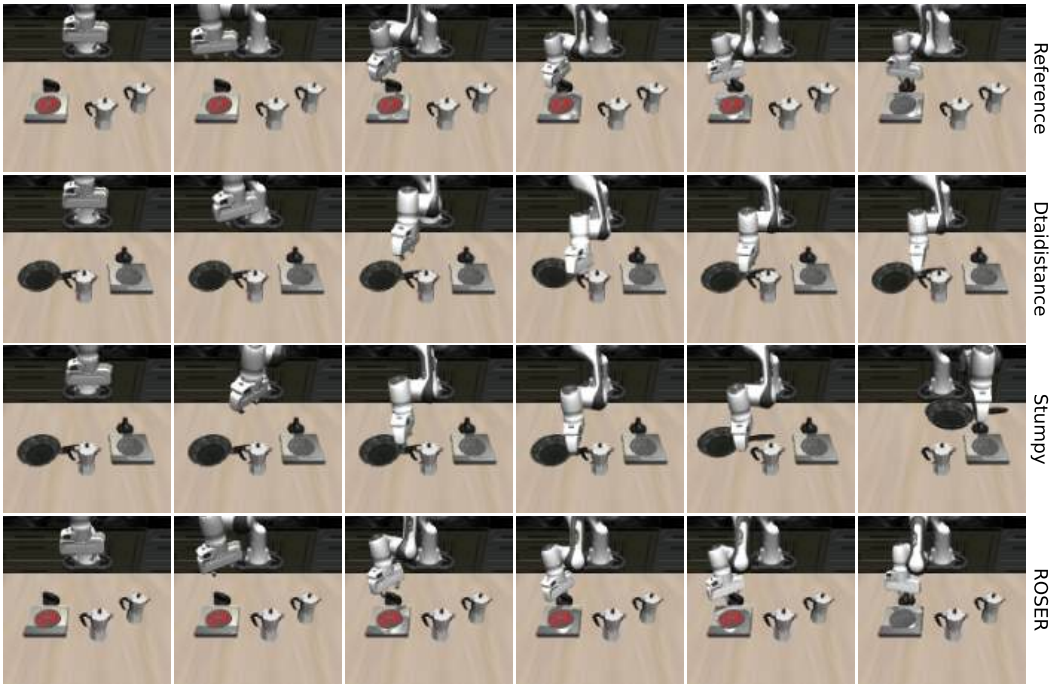

Figure 11: LIBERO qualitative results for stove off. ROSER successfully retrieves stove off tasks while Dtaidistance and Stumpy retrieves pick and place tasks

C.1.5 TOP DRAWER OPEN

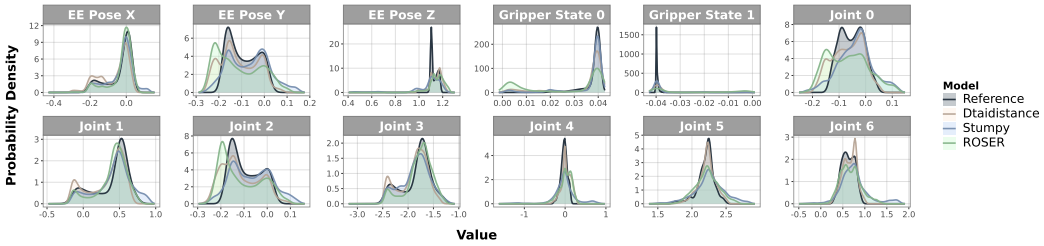

Figure 12: Feature-level distribution visualization for top drawer open task in the LIBERO benchmark.

Figure 13: LIBERO qualitative results for top drawer open. ROSER and Stumpy successfully retrieve the task while Dtaidistance retrieves pick and place task.

C.1.6 TOP DRAWER CLOSE

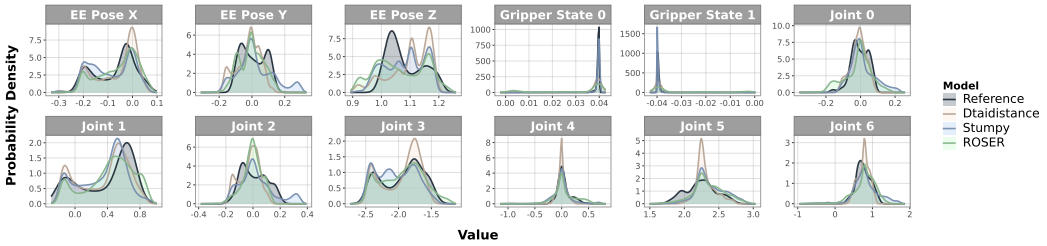

Figure 14: Feature-level distribution visualization for top drawer close task in the LIBERO benchmark.

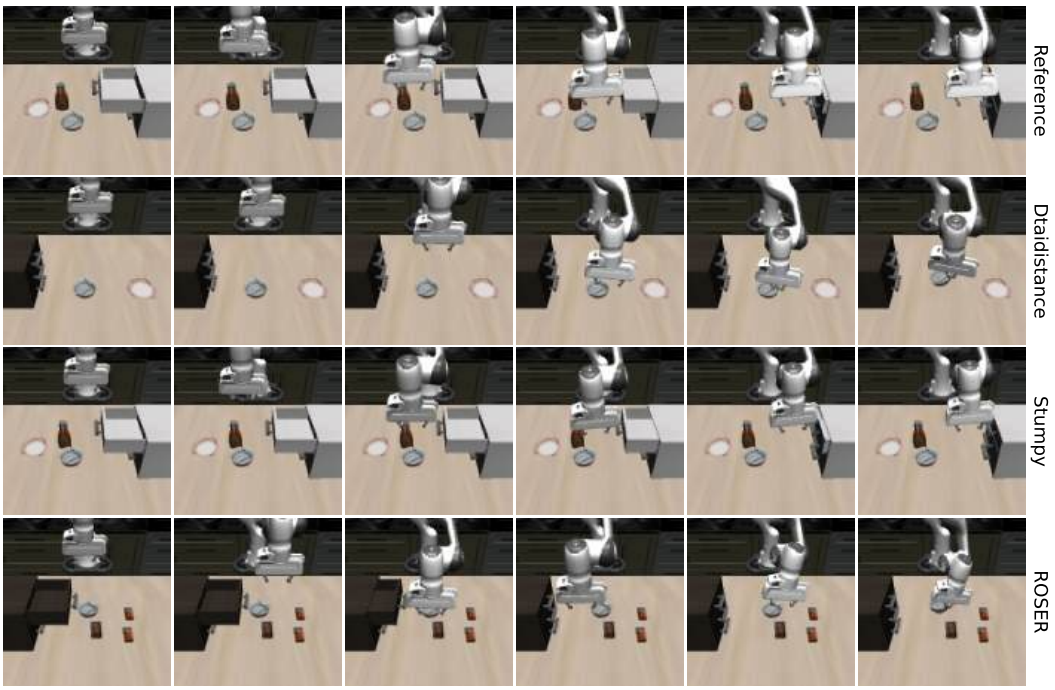

Figure 15: Libero qualitative results for top drawer close. ROSER and Stumpy successfully retrieve the task while Dtaidistance retrieves pick and place task.

C.1.7 BOTTOM DRAWER OPEN

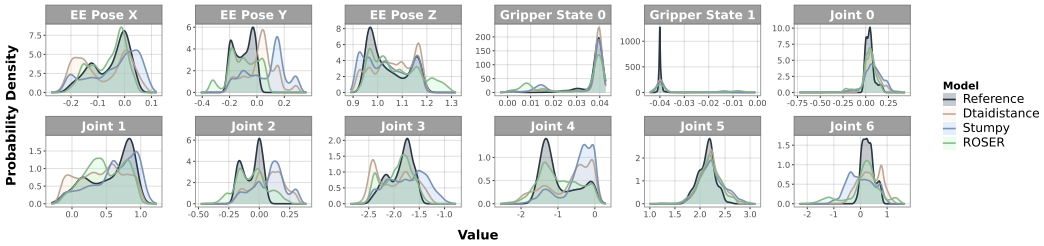

Figure 16: Feature-level distribution visualization for bottom drawer open task in the LIBERO benchmark.

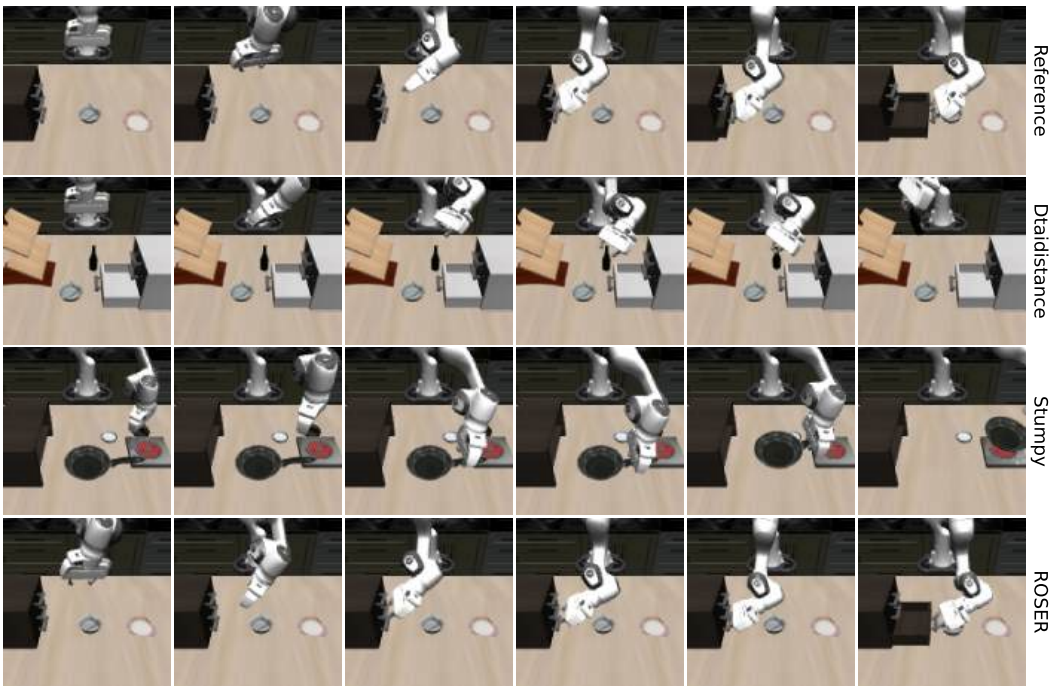

Figure 17: LIBERO qualitative results for bottom drawer open. This task is challenging since robot arms need to avoid collding with top drawer handle to reach bottom drawer handle. ROSER successfully retrieves similar task while Stumpy and Dtaidistance retrieve other tasks.

C.1.8 Bottom Drawer Close

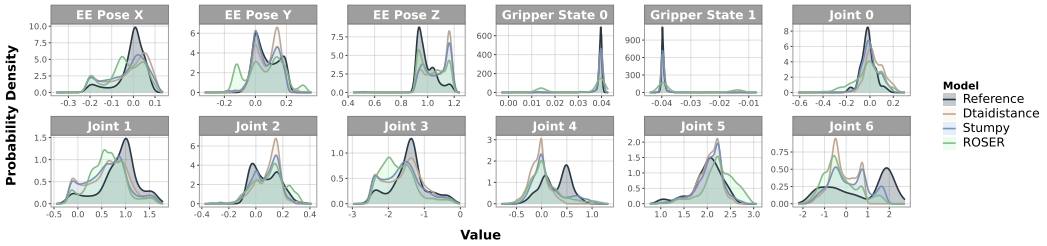

Figure 18: Feature-level distribution visualization for bottom drawer close task in the LIBERO benchmark.

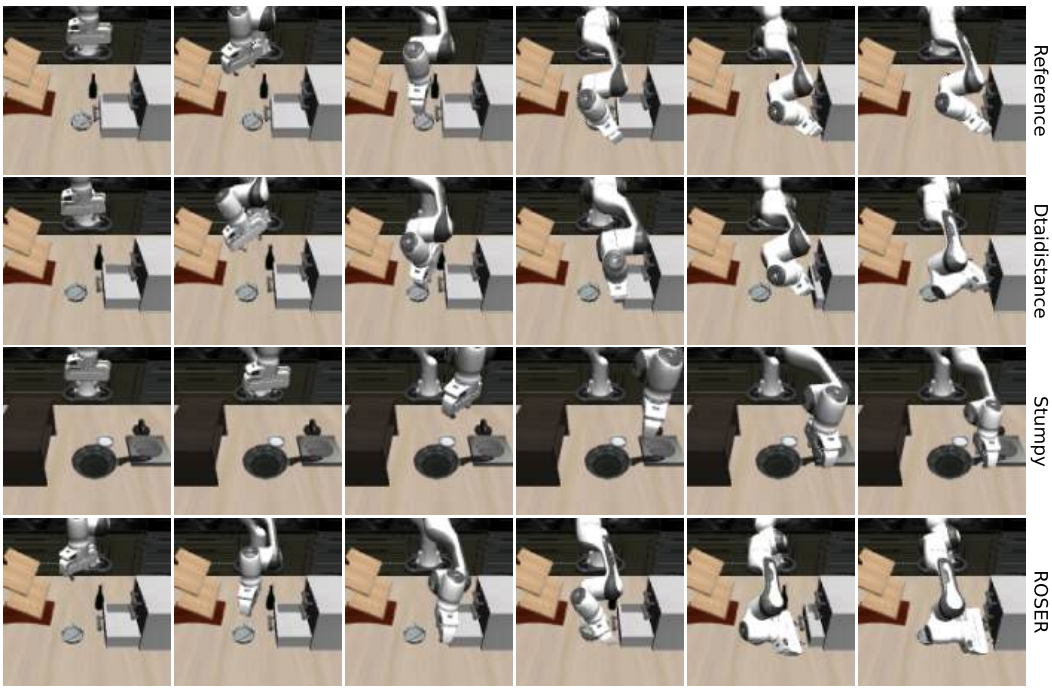

Figure 19: LIBERO qualitative results for bottom drawer close. ROSER and Dtaidistance successfully retrieves similar task while Stumpy retrieves pick and place task.

C.1.9 PICK AND PLACE (PNP)

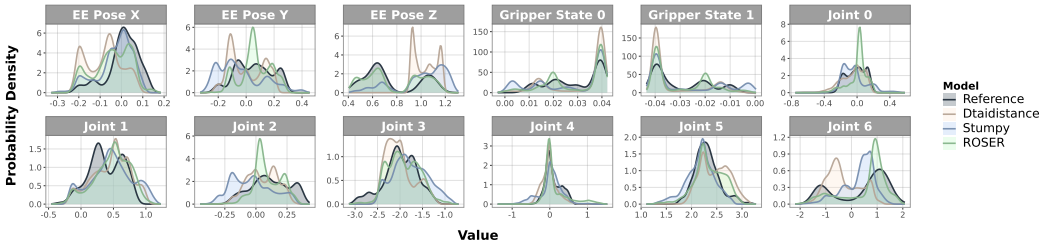

Figure 20: Feature-level distribution visualization for pick and place task in the LIBERO benchmark.

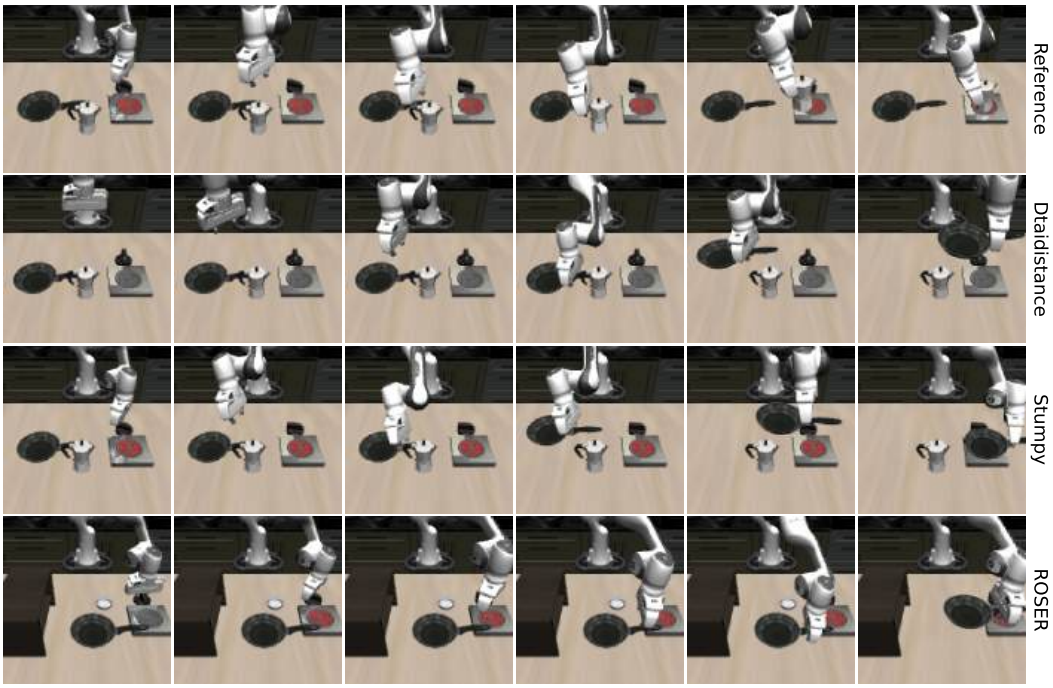

Figure 21: LIBERO qualitative results pick and place. All models can successfully retrieve pick an place tasks

C.2 DROID BENCHMARK

C.2.1 CLOSE DRAWER

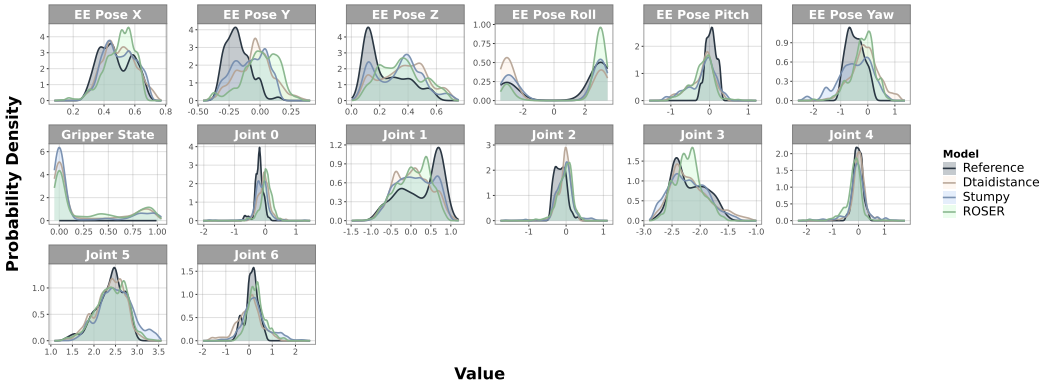

Figure 22: Feature-level distribution visualization for drawer close task in the DROID benchmark.

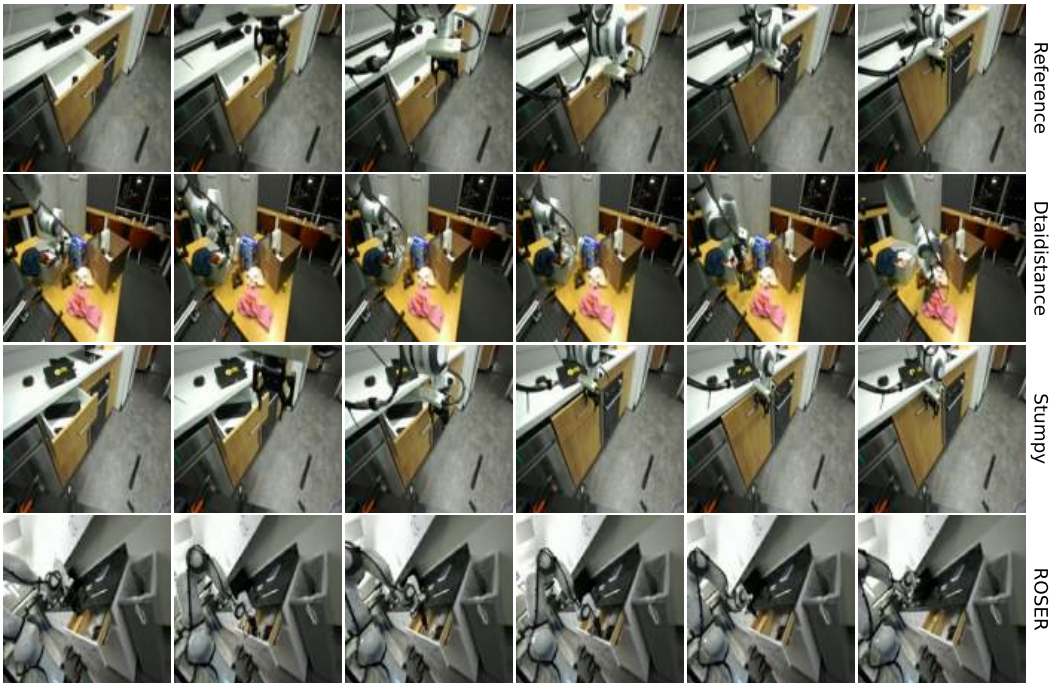

Figure 23: DROID qualitative results for drawer close. ROSER can successfully retrieve similar task even when target environment is different.

C.2.2 PICK AND PLACE (PNP)

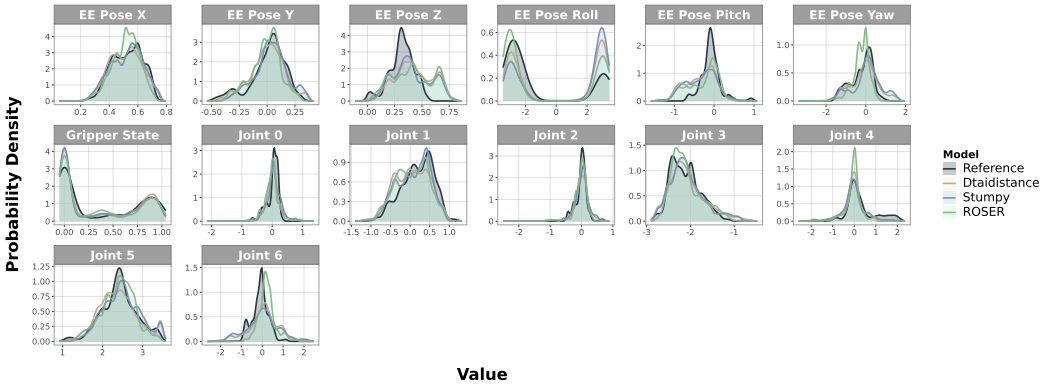

Figure 24: Feature-level distribution visualization for pick and place task in the DROID benchmark.

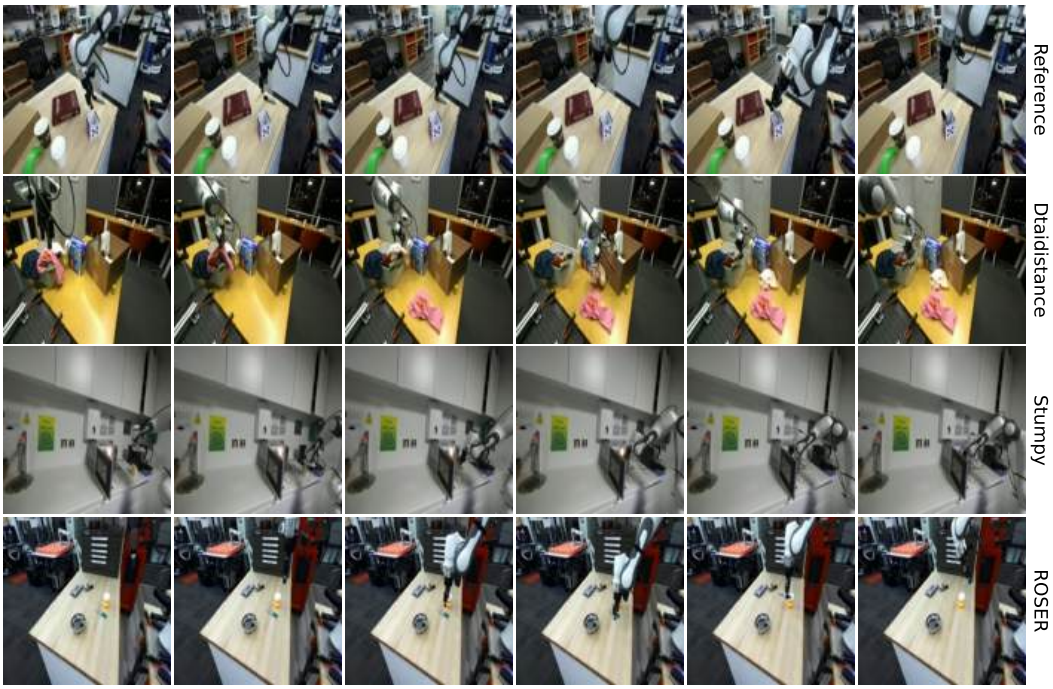

Figure 25: DROID qualitative results for pick and place task. All models successfully retrieve similar tasks.

C.2.3 TURN FAUCET

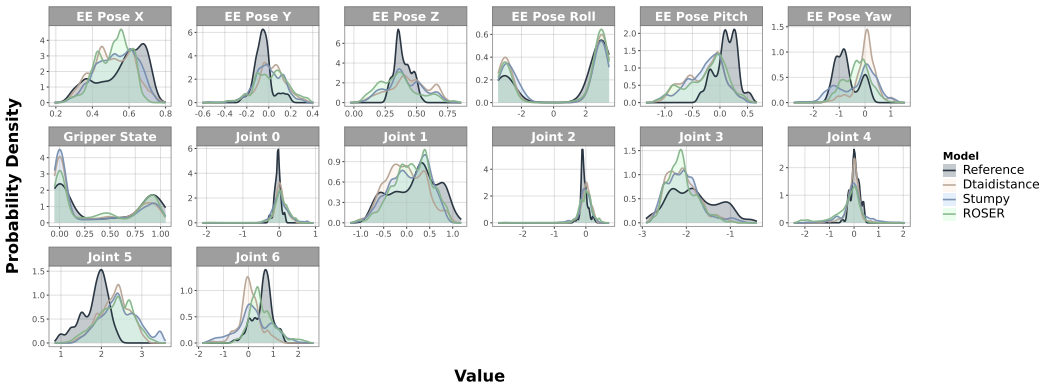

Figure 26: Feature-level distribution visualization for turn faucet task in the DROID benchmark.

Figure 27: DROID qualitative results on turn faucet. All models successfully retrieve similar tasks.

C.2.4 CLOSE CABINET

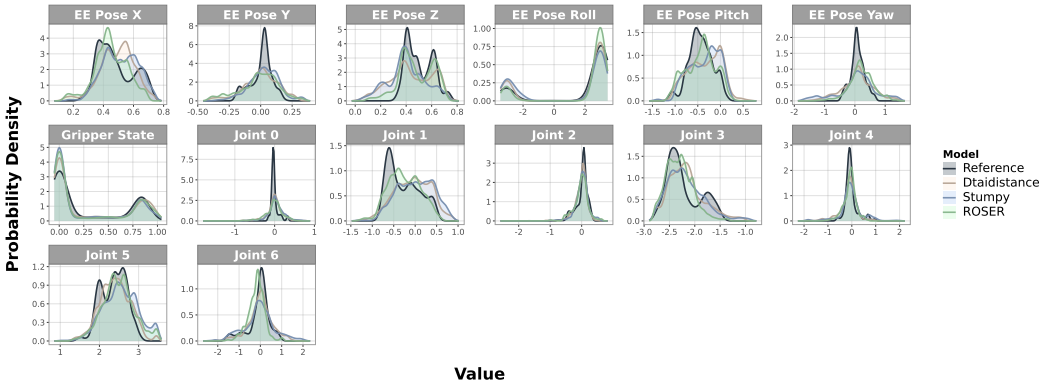

Figure 28: Feature-level distribution visualization for close cabinet task in the DROID benchmark.

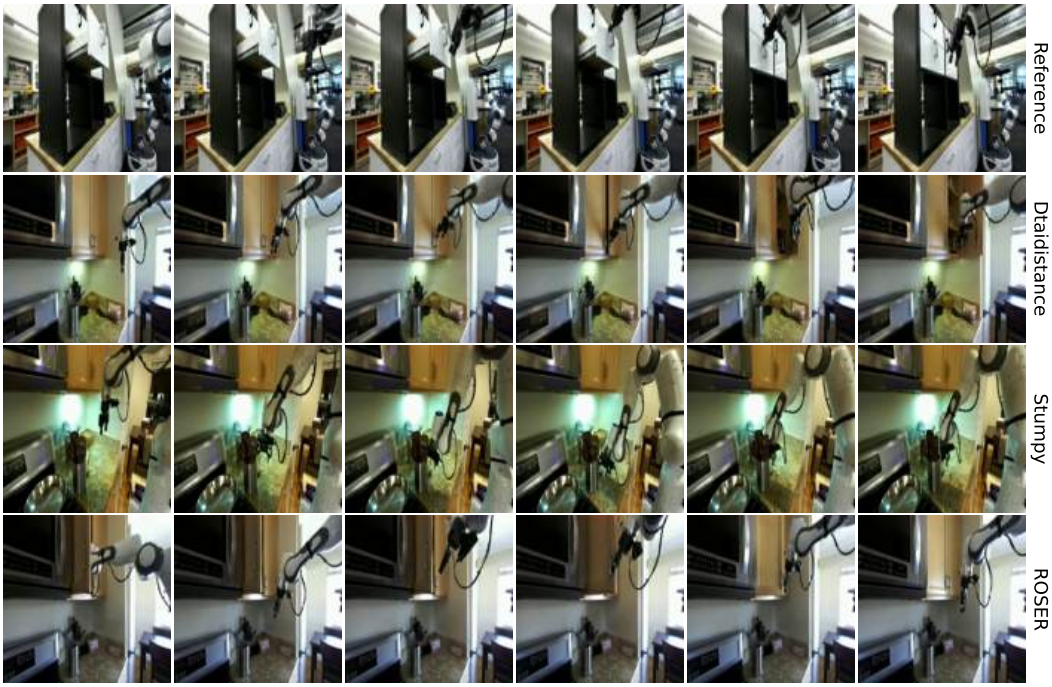

Figure 29: DROID qualitative results for close cabinet task. ROSER successfully retrieves similar task from a different environment while Stumpy retrieves a pick and place task and Dtaidistance retrieves a close cabinet task.

C.2.5 OPEN CABINET

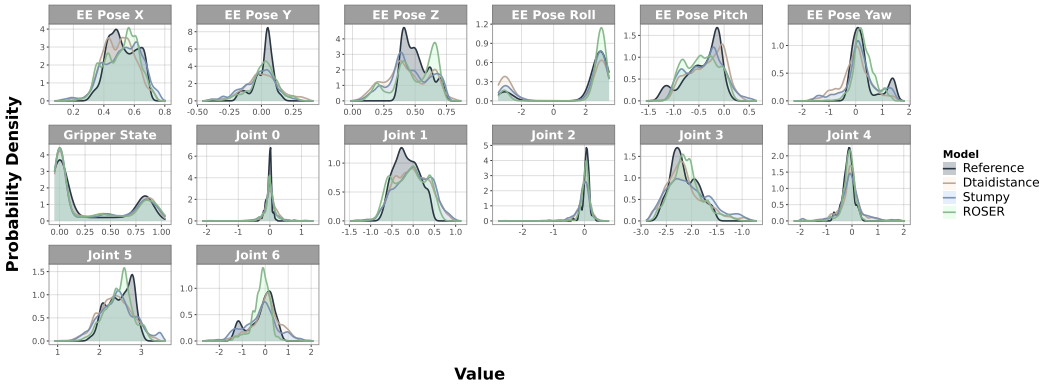

Figure 30: Feature-level distribution visualization for open cabinet task in the DROID benchmark.

Figure 31: DROID qualitative results for open cabinet. ROSER and Dtaidistance successfully retrieve similar task while Stumpy retrieves a pick and place task.

C.3 nuScenes Benchmark

C.3.1 Left Turn

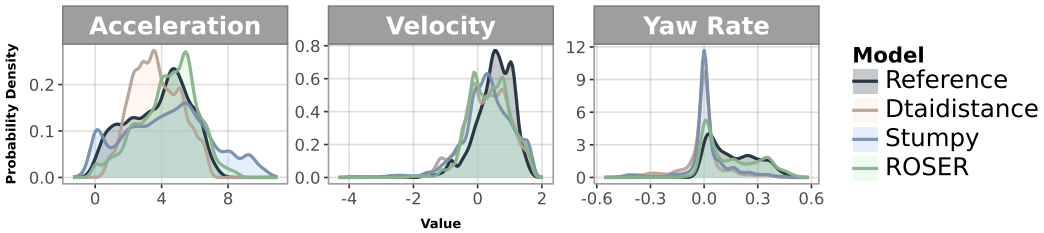

Figure 32: Feature-level distribution visualization for left turn maneuver in the nuScenes benchmark.

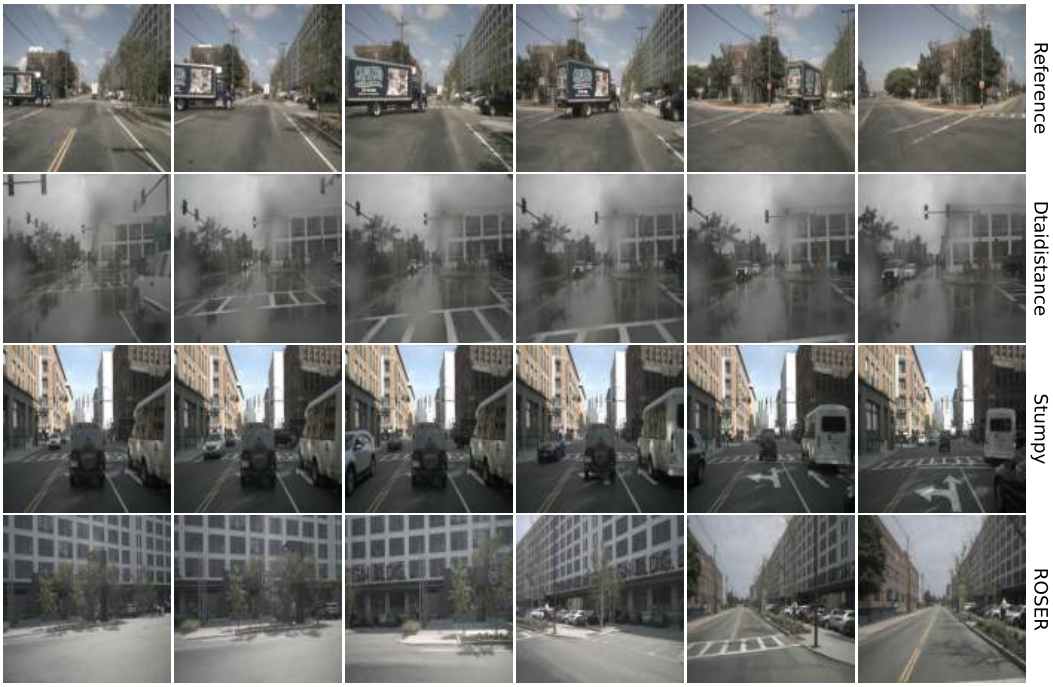

Figure 33: nuScenes qualitative results for left turn maneuver. ROSER successfully retrieves similar task while Dtaidistance retrieves a regular stop and Stumpy retrieves a straight driving scene.

C.3.2 RIGHT TURN

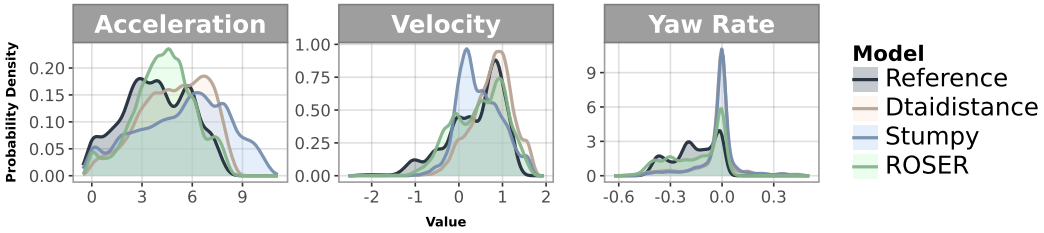

Figure 34: Feature-level distribution visualization for right turn maneuver in the nuScenes benchmark.

Figure 35: nuScenes qualitative results for right turn maneuver. ROSER and Dtaidistance successfully retrieves similar task while Stumpy retrieves a straight driving scene.

C.3.3 STRAIGHT DRIVING

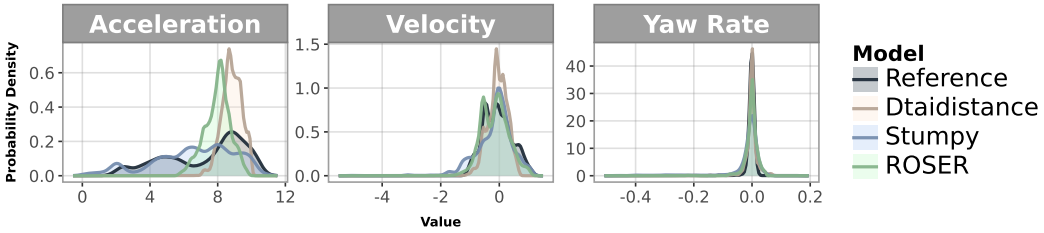

Figure 36: Feature-level distribution visualization for straight driving maneuver in the nuScenes benchmark.

Figure 37: nuScenes qualitative results for straight driving maneuver. All models successfully retrieve similar scene for this task.

C.3.4 REGULAR STOP

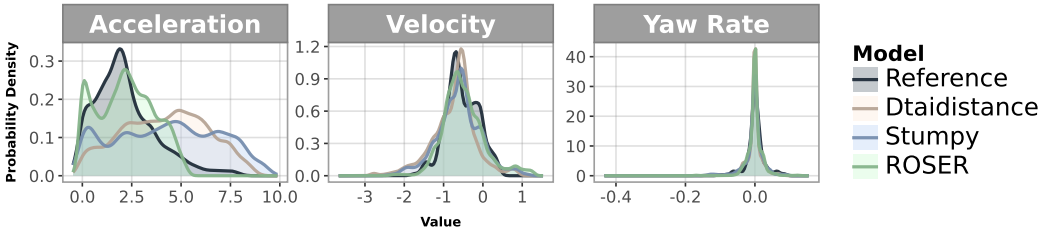

Figure 38: Feature-level distribution visualization for regular stop maneuver in the nuScenes benchmark.

Figure 39: nuScenes qualitative results for regular stop maneuver. ROSER successfully retrieves similar task as the vehicle stops at the crosswalk while Dtaidistance and Stumpy retrieve straight driving scenes.