LESS IS MORE: ON DATA REDUNDANCY IN VLA TRAINING

Kevin Yang* †
Department of Computer Science
Brown University
Providence, RI 02912, USA
kevin_c_yang@brown.edu

Tony Yang* †
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104, USA
tyang08@seas.upenn.edu

ABSTRACT

Recent work suggests robotic benchmarks contain significant data redundancy. In this work, we empirically verify and quantify this on the popular LIBERO benchmark by evaluating two distinct data selection methods: random sampling and difficulty-based sampling. We first evaluate random sampling via random frame downsampling, finding that a sparse 30% coverage of the dataset is sufficient to match the performance of the full dataset. We then investigate whether difficulty-based sampling can improve this further by implementing a temporal surprise score (TSS). TSS identifies high-volatility action frames and selects them as dense, continuous clusters (including temporal neighbors) while discarding low-information transit frames. We find that random sampling matches, and in some cases exceeds, full dataset performance, while difficulty-based sampling underperforms both. This suggests that maintaining broad coverage across diverse scenarios is more crucial than targeting difficult frames for VLA training. This work characterizes this diversity vs. difficulty trade-off, providing an initial empirical analysis of sparse versus dense data selection in VLA training.

1 INTRODUCTION

In robot learning, the rise of large-scale Generalist Robot Policies (GRPs) powered by Vision-Language-Action (VLA) architectures (Black et al., 2024; Kim et al., 2024; NVIDIA, 2024) has shown promising results in generalizing across diverse tasks and environments. However, despite the curation of massive datasets (Collaboration, 2023; Khazatsky et al., 2024) for training, the assumption that "more data means better" overlooks the possibility that not all robot data are equally important.

So what makes the data high-quality for VLA training? Typically, a robot demonstration mostly consists of frames with low-entropy movements such as moving an arm across open areas to reach an object. While these frames are important for execution, they often contain redundant information that is not as difficult to "learn" as frames with more critical moments of interaction, such as grasping a cup or manipulating some tool. Because these critical actions are usually short in duration, a majority of frames in datasets are overwhelmed with these "transit" frames, potentially diluting the model's attention on critical frames that dictate task success. As datasets grow from thousands to millions of trajectories (Walke et al., 2023; Collaboration, 2023), training VLAs with possibly redundant data becomes costly and raises concerns. Thus, we ask the question: Can we achieve better performance with less, but higher-quality, data?

This paper investigates and highlights the extent of data redundancy in established benchmarks, specifically LIBERO (Liu et al., 2023). We quantify this redundancy through extensive empirical testing on both SmolVLA (Shukor et al., 2025) and GR00T N1.5 NVIDIA (2024) by comparing two data selection strategies: random sampling (random downsampling) and difficulty-based sampling. We hypothesize that an effective data selection strategy should prioritize "temporal surprise", or moments where action trajectories undergo significant shifts. To investigate this, we propose Temporal

*Equal contribution.
†Work performed while at AI2 Robotics.

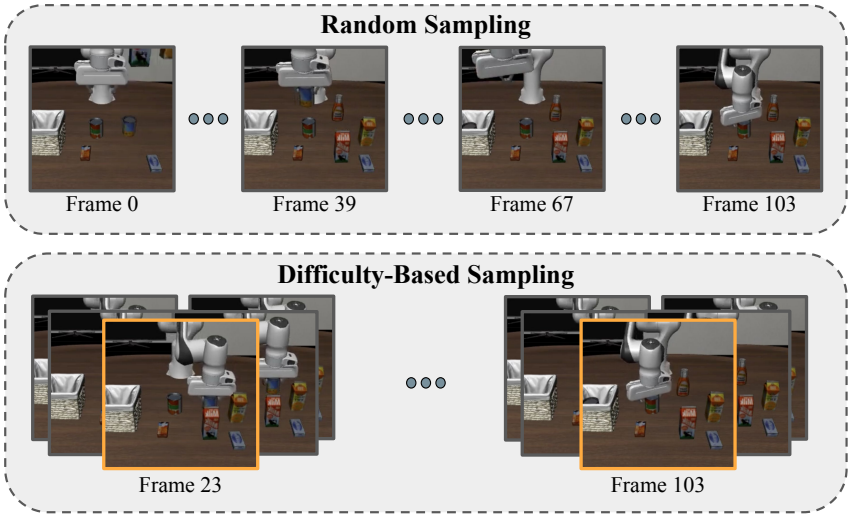


Figure 1: **Method Comparison.** Visualization for random sampling and difficulty-based sampling. Center frames (bottom) are selected using TSS.

Surprise Score (TSS), a metric that scores frames based on a weighted calculation of action change. TSS picks out frames with critical transitions, and our results show random selection outperforms TSS, demonstrating the importance of data diversity compared to data difficulty.

2 Related Works

Recent works have been mainly focused on redundancy in VLA processing during inference. **VLA-Pruner** (Liu et al., 2025) and **SpecPrune-VLA** (Wang et al., 2025) show that a large portion of visual tokens can be pruned during generation without hurting performance via temporal continuity and attention scores. Additionally, **Action-aware Dynamic Pruning (ADP)** (Pei et al., 2025) addresses this redundancy problem to task phases and observes that aggressive pruning is safe during coarse manipulation (transit) but detrimental during fine-grained interactions. Although these methods focus on mitigating data redundancy during *inference*, the models used are still trained on the full, original dataset. We extend this insight to the training stage, asking: if the model need not attend to these frames during inference, does it need to learn from them at all?

3 Data Selection Methods

3.1 Random Sampling

First, we empirically test our episode redundancy hypothesis by training GR00T N1.5 and SmolVLA on different randomly selected coresets from the full LIBERO dataset. To create the coreset, we randomly sample $p\%$ of the total 273,465 frames uniformly across all episodes. We hypothesize that if there truly is redundancy within the data, random coreset performance should match that of the full dataset.

3.2 Difficulty-based sampling

Having established that random selection works surprisingly well, we investigate whether targeted selection of difficult frames can improve performance. We consider imitation learning from a dataset of expert demonstrations $\mathcal{D} = \{(o_t, a_t)\}_{t=1}^{N}$, where o_t denotes the observation (e.g., RGB image)

and $a_t \in \mathbb{R}^d$ denotes the action at timestep t. The dataset consists of K episodes, where each episode k contains a trajectory $\tau_k = \{(o_t^k, a_t^k)\}_{t=1}^{T_k}$ of length T_k.

For manipulation tasks, the action vector typically comprises motion components $m_t \in \mathbb{R}^{d-1}$ (position and orientation deltas) and a gripper state $g_t \in \mathbb{R}$ (open/close command):

$$a_t = [m_t; g_t] \tag{1}$$

Our goal is to assign an importance score $S(t) \in \mathbb{R}_{\geq 0}$ to each frame, where higher scores indicate frames that are more critical for learning the task. These scores are then used to select a coreset of high-importance frames.

Temporal Surprise Score We propose Temporal Surprise Score (TSS), a model-free metric that identifies important frames based on action dynamics. The key insight is that at typical recording frequencies (10-30 Hz), consecutive frames in smooth robot motion contain nearly identical action vectors. Frames where the robot's behavior changes such as direction reversals, grasp events, or motion initiation are precisely the moments that matter most for policy learning.

TSS scores each frame based on two components:

$$S(t) = D_{\cos}(m_t, m_{t-1}) + \gamma \cdot |g_t - g_{t-1}| \tag{2}$$

where $D_{\cos}(a, b) = 1 - \frac{a \cdot b}{\|a\|\|b\|}$ is the cosine distance, and γ is a hyperparameter weighting the gripper term.

Direction Change. The first term measures the change in motion direction using cosine distance rather than L2 distance. This design choice is motivated by the observation that frame-to-frame action *magnitudes* are nearly constant during smooth motion since the robot moves at roughly steady speed. L2 distance primarily captures speed variations, which are less semantically meaningful than directional changes. Cosine distance instead captures when the robot changes *what* it is doing: moving left then suddenly moving up registers as high surprise, regardless of speed.

Gripper Events. The second term explicitly detects gripper state changes. Grasp and release events are discrete, high-importance moments that would otherwise be diluted across the full action dimension. Separating this term ensures that gripper transitions always contribute to the surprise score.

Edge Cases. We handle two edge cases: (1) *Episode boundaries*: The first frame of each episode has no previous action for comparison; we assign $S(t) = 0$ to these frames. (2) *Stationary robot*: When $\|m_t\| \approx 0$ or $\|m_{t-1}\| \approx 0$, cosine similarity is undefined. We set the direction change to 0 if both frames are stationary, and to 1 if transitioning to/from motion.

Temporal Continuity Preliminary results along with previous studies indicate the importance of temporal continuity in demonstration frames. To ensure this, our resulting coreset consists of the top 20% frames with the highest score, each with a context window of $k = 5$ frames. Ablations show that $k = 5$ is the optimal context window size. Duplicate frames due to overlapping context windows are removed.

4 EXPERIMENTS

For our experiments, we compare downstream evaluation on LIBERO results from our TSS method against a random coreset of the same proportion as well as the full dataset baseline. In particular, we want to answer the following question: Can our constructed coresets yield the same, if not better performance, as the full dataset? To establish a compute-efficient baseline for academic replication, we standardize training to 30k steps with a batch size of 32. Preliminary ablations (Table 7) found that a higher batch size of 64 did not yield significant improvement while increasing computational overhead. Finally, we follow the evaluation protocol established by the LeRobot GR00T N1.5 implementation and SmolVLA implementation (Cadene et al., 2024). We report performance on the

Table 1: Performance comparison of data selection methods on LIBERO benchmark. Results show success rates (%) across four task suites using 30% of training data, except for baselines using 100%. TSS uses $k = 5$ nearest neighbors and retains approximately 64% of frames. Hardest coreset refers to the top-p% highest TSS frames without temporal neighbors.

Model	Method	Data %	Long	Object	Spatial	Goal	Average
SmolVLA	Original Baseline	100	35	68	63	73	59.75
	Hardest Coreset	30	30	67	55	59	52.75
	Random Coreset	30	**36**	**75**	**66**	**65**	**60.50**
	TSS (k=5)	∼64	34	71	58	62	56.25
GR00T	Original Baseline	100	66	**100**	75	-	80.33
	Hardest Coreset	30	61	82	74	-	72.33
	Random Coreset	30	**69**	95	**80**	-	**81.33**
	TSS (k=5)	∼64	59	92	71	-	74.00

Libero-Spatial, Libero-Object, Libero-Goal, and Libero-Long benchmarks, excluding the Goal suite for GR00T as per official LeRobot implementation due to known convergence limitations. Training details are provided in Appendix A.2.

5 Results

Random sampling reveals dramatic data redundancy in LIBERO. As seen in Figure 2, the performance of the models trained on coresets of different proportions remains relatively stable, until its decline past 5%. This demonstrates that VLAs like GR00T and SmolVLA can be effectively post-trained on down to only 5% of the full LIBERO dataset, indicating that the dataset is heavily frame-redundant.

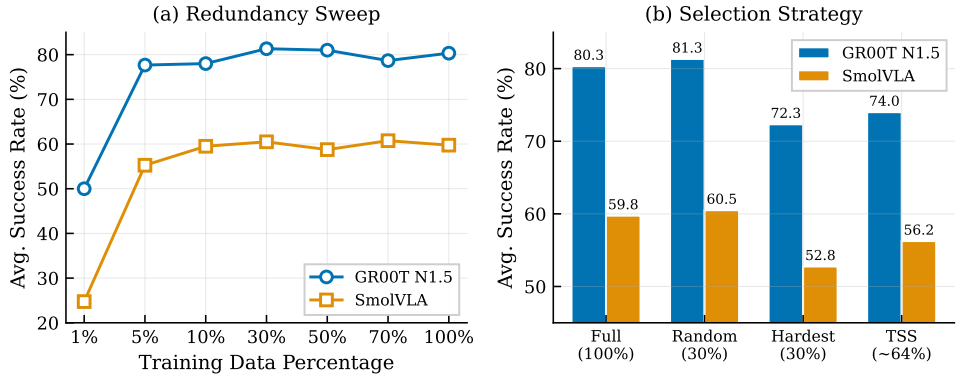


Figure 2: **Data redundancy analysis in VLA models.** (a) Average success rate across training data percentages ($p\%$). (b) Comparison of data selection strategies against the 100% baseline.

Despite using 64% of frames (vs. 30% random), TSS underperforms random selection for both models as shown in Table 1 where SmolVLA achieves 56.25% with TSS vs. 60.50% with random sampling, while GR00T achieves 74.00% vs. 81.33%. This suggests that learning from the most difficult frames does not necessarily translate to improved performance.

6 Discussion

Our findings reveal a critical inefficiency in current robot learning data collection: we are capturing and storing vast amounts of redundant information. The fact that random 5-10% coresets match full dataset performance indicates that the primary value of large datasets lies in their diversity, not their

raw size. We hypothesize that LIBERO's 1690 episodes contain extensive within-task redundancy but good across-task diversity.

Why does random selection work so well? Our results suggest that large demonstration datasets are valuable due to their *coverage* of the state-action space rather than their density. Through random subsampling, this coverage is preserved proportionally across episodes and task phases, while difficulty-based sampling (TSS, Hardest Coreset) can create gaps in coverage that hurt generalization. This aligns with recent findings that VLA models are quite robust to missing frames during inference (Liu et al., 2025; Pei et al., 2025). A 5% random sample can preserve a fair coverage of different scenarios (different object positions, initial states, approach angles) while filtering out many redundant variations of the same maneuver. This suggests VLAs learn task-level strategies rather than memorizing specific trajectories. Once the model has seen one example of grasping a cup from the left, additional examples from slightly different left angles provide diminishing returns. Thus, the success of random sub-sampling may reflect an implicit curriculum effect. By reducing the model's exposure to frames that are repetitive, easy examples will not dominate the model's gradient updates as much as it did before, possibly leading to faster convergence.

Diversity vs. Difficulty Trade-off. Interestingly, we observe a tension where random sampling maintains diversity but filters out potentially important frames, while difficulty-based sampling preserves difficult moments but in turn sacrifices coverage. It is clear that our results show favor for diversity as even at 5-10% sampling proportions, random selection still outperforms curated 30% coresets. This suggests that current VLA architectures are already capable of learning from sparse supervision and that what they need is broader coverage, not harder examples. However, we recognize that diversity vs. difficulty may not be a simple trade-off. Future work should further explore the effectiveness of the two through dynamic curriculum learning, phasing in difficult frames only after diverse coverage.

Limitations. In this work, our analysis is solely focused on LIBERO, a simulation benchmark that contains relatively short-horizon tasks. It is possible that the extent of redundancy may differ in (1) real-world datasets with more diversity, (2) longer-horizon tasks that require more temporal focus, or (3) multi-task settings where coverage across tasks is more important. Additionally, given that we only evaluate two VLA architectures, redundancy patterns should not be plainly generalized to every model.

Why didn't TSS help? We hypothesize two reasons: (1) High-TSS frames (direction changes, grasps) may be *already well-represented* in random samples since they occur regularly throughout demonstrations. (2) The temporal neighbors ($k = 5$) we add to preserve continuity may dilute the signal, effectively approximating random selection. Future work could explore adaptive k or learned importance scores.

Our findings have several practical implications, though they require further validation on larger, real-world datasets. First, **post-hoc subsampling** of densely recorded data can reduce storage and training costs by 70% while maintaining performance in our compute-efficient training setup (30k steps). However, directly collecting frames at lower frequencies can risk not capturing crucial high-speed interactions. In this case, adaptive sampling strategies that densely record interactions but sparely sample transit phases may be more appropriate.

7 Conclusion

In this paper, we compare two methods of data selection: random sampling and difficulty-based sampling. Preliminary results on the LIBERO dataset show that models trained on less data are able to achieve comparable performance to baseline results, indicating significant data redundancy. Our results suggest that, under the conditions studied, data diversity is more critical than difficulty-based selection for VLA training on LIBERO, though further investigation across other difficulty metrics and benchmarks is needed to generalize this finding. For future work, it would be worth exploring other datasets in the same fashion. We believe our work serves as an eye-opener for problems within existing data and opens the door for more efficient training.

REFERENCES

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. URL https://arxiv.org/abs/2410.24164.

Remi Cadene, Simon Alibert, Alexander Soare, Quentin Gallouedec, Adil Zouitine, Steven Palma, Pepijn Kooijmans, Michel Aractingi, Mustafa Shukor, Dana Aubakirova, Martino Russi, Francesco Capuano, Caroline Pascal, Jade Choghari, Jess Moss, and Thomas Wolf. Lerobot: State-of-the-art machine learning for real-world robotics in pytorch, 2024. URL https://github.com/huggingface/lerobot.

Open X-Embodiment Collaboration. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023. URL https://arxiv.org/abs/2310.08864.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024. URL https://arxiv.org/abs/2403.12945.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. URL https://arxiv.org/abs/2406.09246.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *NeurIPS Datasets and Benchmarks Track*, 2023. URL https://arxiv.org/abs/2306.03310.

Ziyan Liu, Yeqiu Chen, Hongyi Cai, Tao Lin, Shuo Yang, Zheng Liu, and Bo Zhao. VLA-Pruner: Temporal-aware dual-level visual token pruning for efficient vision-language-action inference. *arXiv preprint arXiv:2511.16449*, 2025.

NVIDIA. Nvidia isaac gr00t: A foundation model for humanoid robots, 2024. URL https://developer.nvidia.com/isaac/gr00t.

Xiaohuan Pei, Yuxing Chen, Siyu Xu, Yunke Wang, Yuheng Shi, and Chang Xu. Action-aware dynamic pruning for efficient vision-language-action manipulation. *arXiv preprint arXiv:2509.22093*, 2025.

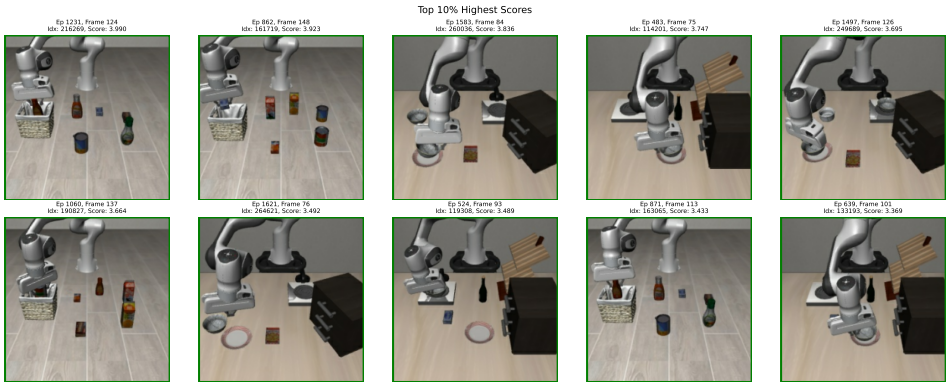
Mustafa Shukor, Merve Aubakirova, Victor Lecomte, Loubna Ben Allal, Theo Gigant, Alexis Roger, and Thomas Wolf. Smolvla: A vision-language-action model for affordable and efficient robotics, 2025. URL https://arxiv.org/abs/2506.01844.

Homer Walke, Kevin Black, Tony Z Zhao, et al. Bridgedata v2: A dataset for robot learning at scale. *Conference on Robot Learning (CoRL)*, 2023. URL https://arxiv.org/abs/2308.12952.

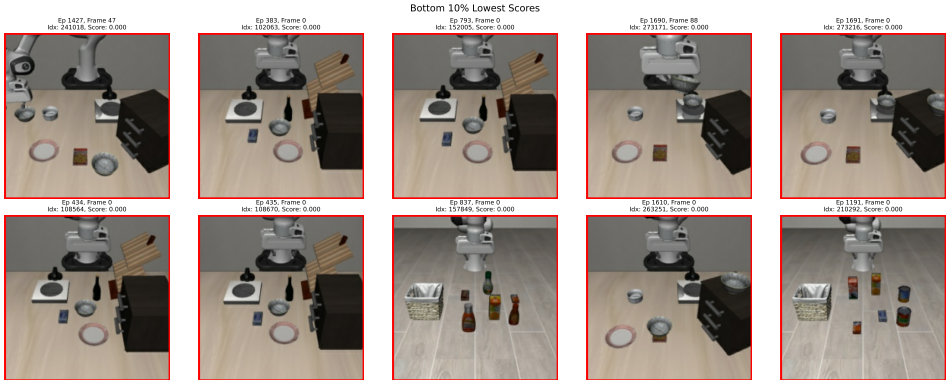
Hanzhen Wang, Jiaming Xu, Jiayi Pan, Yongkang Zhou, and Guohao Dai. SpecPrune-VLA: Accelerating vision-language-action models via action-aware self-speculative pruning. *arXiv preprint arXiv:2509.05614*, 2025.

A Appendix

A.1 Qualitative Evaluation of TSS



(a) Top 10 highest TSS scores



(b) Bottom 10 lowest TSS scores

Figure 3: Visualization of TSS scores for frames in LIBERO. (a) Frames with highest information content show diverse robot configurations and critical interaction moments. (b) Frames with lowest scores contain repetitive or low-information transitions.

A.2 Experiment Details

Table 2: Training configuration for both models.

Parameter	**SmolVLA**	**GR00T N1.5**
Base model	SmolVLM2-500M	GR00T-N1.5-3B
Learning rate	1e-4	1e-4
LR schedule	Linear decay	Cosine decay
Warmup steps	1,000	500
Weight decay	1e-10	1e-5
Batch size	32	32
Training steps	30k	30k
Gradient clip	10.0	10.0
Image size	256×256	224×224
Action dim	7	7
Precision	FP32	BF16
Eval rollouts/task	10	10

Both models trained using LeRobot. Random coresets sample frames uniformly. TSS uses γ=1, k=5.

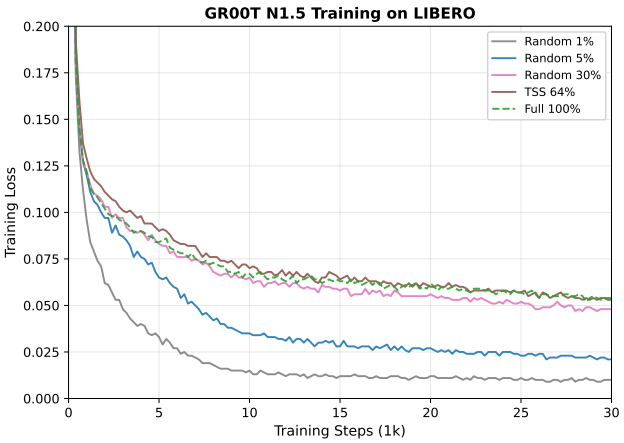


Figure 4: Training loss curves for GR00T N1.5 fine-tuning on LIBERO. Smaller coresets achieve lower training loss (1%: 0.010, 5%: 0.021), while larger coresets converge similarly (30%: 0.048, TSS 64%: 0.053, Full: 0.054).

A.3 P% Data Sweep

Table 3: Random coreset scaling for GR00T N1.5 on LIBERO. Success rates (%) across four task suites with varying training data percentages.

Data %	Long	Object	Spatial	Average
100	66	**100**	75	80.33
70	62	96	78	78.67
50	72	98	73	**81.00**
30	69	95	**80**	**81.33**
10	**74**	89	71	78.00
5	69	96	68	77.67
1	29	71	50	50.00

Table 4: Random coreset scaling for SmolVLA on LIBERO. Success rates (%) across four task suites with varying training data percentages.

Data %	Long	Object	Spatial	Goal	Average
100	35	68	63	73	59.75
70	34	70	65	74	60.75
50	33	68	57	**77**	58.75
30	**36**	**75**	**66**	65	**60.50**
10	34	70	58	76	59.50
5	32	65	52	72	55.25
1	17	27	18	37	24.75

A.4 Ablation Studies

A.4.1 Effect of k on SmolVLA Performance

Table 5 shows the effect of varying the number of nearest neighbors k in the TSS metric on SmolVLA performance.

Table 5: Ablation study: Effect of k parameter on SmolVLA with TSS metric.

k	Data %	Long	Object	Spatial	Goal	Average
1	34	28	67	49	55	49.75
3	52	27	68	51	62	52.00
5	64	**34**	**71**	58	62	**56.25**
7	73	32	64	**59**	**70**	**56.25**
9	80	30	69	51	64	53.50

A.4.2 Effect of γ on SmolVLA Performance

Table 6 shows the effect of varying the temporal weight parameter γ with fixed $k = 5$.

Table 6: Ablation study: Effect of γ parameter on SmolVLA with $k = 5$.

γ	Long	Object	Spatial	Goal	Average
0.5	33	73	57	64	**56.75**
1.0	**34**	71	58	62	56.25
2.0	24	**74**	**59**	**70**	**56.75**

The specific choice of weighting the gripper term did not matter significantly. Thus, in our experiments, $\gamma = 1$ was used.

A.4.3 Effect of Batch Size for GR00T Training

Table 7 shows the comparison for GR00T training using batch size 32 (default) vs. 64. Both configurations exhibit similar redundancy patterns, with 30% random coresets matching full-data performance. In our paper, we focus on training with batch size 32 due to the negligible difference in performance.

Table 7: Batch size ablation for GR00T N1.5 on LIBERO.

Batch	Method	Data %	Long	Object	Spatial	Avg.
32	Full	100	66	**100**	75	80.33
	Random	30	69	95	**80**	**81.33**
	Random	5	69	96	68	77.67
	TSS	$\sim$64	59	92	71	74.00
64	Full	100	**72**	95	77	**81.33**
	Random	30	65	96	76	79.00
	Random	5	56	94	70	73.33
	TSS	$\sim$64	64	96	77	79.00

A.5 Random Seed Robustness

We evaluate the robustness of our 30% random coreset selection across different random seeds to ensure our results are not due to lucky sampling.

A.5.1 SmolVLA with Different Seeds

Table 8: SmolVLA performance with 30% random coreset across different random seeds.

Seed	Data %	Steps	Long	Object	Spatial	Goal	Average
42	30	30k	36	75	66	65	60.50
123	30	30k	33	62	69	80	61.00
2024	30	30k	31	72	63	66	58.00

A.5.2 GR00T with Different Seeds

Table 9: GR00T performance with 30% random coreset across different random seeds.

Seed	Data %	Steps	Long	Object	Spatial	Goal	Average
42	30	30k	69	95	80	-	81.33
123	30	30k	72	95	77	-	81.33
2024	30	30k	76	98	72	-	82.00

Runs with different seeds show that the significance of our results were not just based on chance.