RoboSSM: Scalable In-context Imitation Learning via State-Space Models

Anonymous Author(s)
Affiliation
Address
`email`

Abstract: In-context imitation learning (ICIL) enables robots to learn tasks from prompts consisting of just a handful of demonstrations. By eliminating the need for parameter updates at deployment time, this paradigm supports few-shot adaptation to novel tasks. However, recent ICIL methods rely on Transformers, which have computational limitations and tend to underperform when handling longer prompts than those seen during training. In this work, we introduce RoboSSM, a scalable recipe for in-context imitation learning based on state-space models (SSM). Specifically, RoboSSM replaces Transformers with Longhorn – a state-of-the-art SSM that provides linear-time inference and strong extrapolation capabilities, making it well-suited for long-context prompts. We evaluate our approach on the LIBERO benchmark and compare it against strong Transformer-based ICIL baselines. Experiments show that RoboSSM extrapolates effectively to varying numbers of in-context demonstrations, yields high performance on unseen tasks, and remains robust in long-horizon scenarios. These results of RoboSSM highlight the potential of SSMs as an efficient and scalable backbone for ICIL.

Keywords: Imitation Learning, In-context Learning, State-Space Model

1 Introduction

Imitation Learning (IL) is a powerful framework that enables robots to learn behaviors from demonstrations without explicit programming or reward design [1, 2]. While IL has achieved notable success in manipulation and navigation tasks, a key limitation of conventional imitation learning lies in its restricted adaptation capability, particularly when faced with new tasks. Even with models trained on large multi-task datasets [3, 4, 5, 6], adapting to novel tasks still requires collecting a large amount of task-specific data and retraining, which can be computationally costly and often unstable [7, 8]. To address this challenge, In-Context Imitation Learning (ICIL) introduces a new paradigm, inspired by the success of large language models (LLMs) in adapting to unseen language tasks through few-shot learning [9]. ICIL integrates the concept of prompting into imitation learning, allowing the model to infer and perform tasks based on a prompt composed of demonstrations, with no post-demonstration training.

Given that ICIL formulates imitation learning as a sequence modeling problem, recent ICIL approaches have naturally adopted Transformer-based models as their primary architecture [10, 11, 12]. Although Transformers are the dominant architecture for sequence modeling, their time complexity scales quadratically with sequence length, and they struggle to extrapolate beyond training lengths [13, 14]. To enable the use of long prompts at test time in ICIL, it is essential to adopt alternatives to Transformers that enhance scalability with input length. In this paper, we introduce **RoboSSM**, a scalable in-context learning framework that replaces Transformers with state-space models (SSMs). Specifically, RoboSSM utilizes Longhorn [15], a state-of-the-art SSM with linear inference time and strong extrapolation capability for long-context sequences. Leveraging these properties, RoboSSM can process substantially longer prompts at test time compared to previous

Transformer-based ICIL methods. On the LIBERO [16] benchmark, RoboSSM uniquely benefits from using more in-context examples, maintaining high success rates on unseen tasks when trained with only a few demonstrations. For instance, RoboSSM achieves a higher success rate as more demonstrations are provided for the task *pick up the plate and place it in the tray*, even though the plate *object* was never seen in the training data. Furthermore, our framework performs well on unseen long-horizon tasks, which we simulate by repeating frames in the demonstrations to create time-dilated scenarios. Consequently, we observe that RoboSSM can handle prompts up to 16x longer than those used during training, whereas Transformer-based ICIL methods exhibit a performance collapse once the test prompt length exceeds the training length. These results demonstrate that RoboSSM can execute unseen tasks by leveraging long-range in-context information without any task-specific parameter updates, while outperforming Transformer-based ICIL methods.

2 Related Work

In this section, we provide an overview of prior ICIL methods and state-space models (SSMs), along with their recent applications to robotics.

2.1 In-Context Imitation Learning

Imitation learning has long been a foundation for imparting skills to robots by learning from demonstration data. Standard behavior cloning approaches [17, 18] typically train a separate policy for each task or rely on large multi-task datasets to acquire broader skills. While multi-task imitation learning [3, 4] can handle diverse tasks, these methods still struggle to perform completely unseen tasks without additional data collection for fine-tuning.

Inspired by the in-context learning paradigm in large language models (LLMs) [9], recent imitation learning methods aim to eliminate parameter updates at test time, instead prompting a multi-task policy with a few demonstrations of unseen tasks. Keypoint Action Tokens [11] introduce an ICIL framework that converts the visual observations and actions into tokens, which are fed into a pre-trained large language model. ICRT [10] performs in-context learning using a causal Transformer that predicts actions with next-token prediction, conditioned on a prompt consisting of a sequence of encoded teleoperated demonstrations. LipVQ-VAE [12] is an action tokenizer that uses vector quantization to address the lack of temporal smoothness in existing tokenizers and enable ICIL.

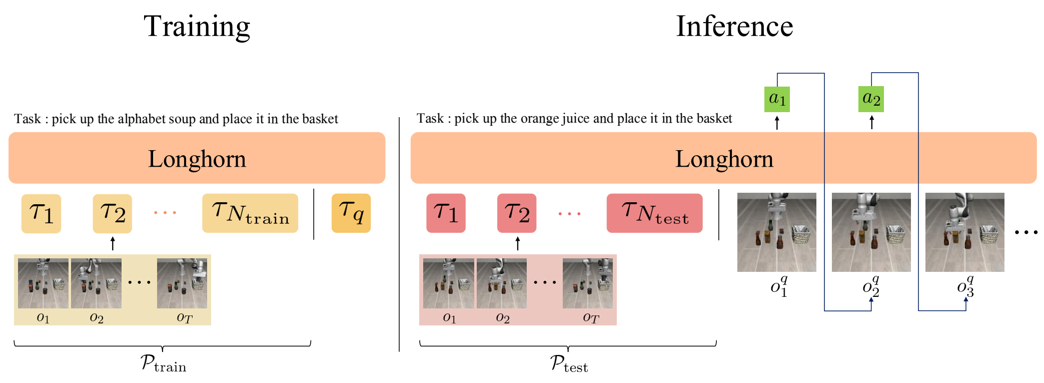
Although ICIL methods have been extensively studied and have achieved significant progress, they are typically trained and evaluated on test prompts that closely match the training prompt distribution in terms of length. For instance, ICRT learns from inputs containing five demonstrations and masks the first random k of them as the prompt, then at inference time is evaluated with three demonstration prompts. LipVQ-VAE is trained and evaluated using only a single full demonstration as the prompt. In contrast, RoboSSM explicitly aims to handle prompts that significantly deviate from the training distribution, such as those containing a larger number of demonstrations or long-horizon tasks.

2.2 State-Space Models

State-space models (SSMs) have emerged as a promising alternative to Transformers for sequence modeling in language tasks, addressing the quadratic time complexity of Transformers and their limitations in handling long contexts. SSMs originate from classical control theory and are particularly inspired by continuous-time linear dynamical systems. By discretizing the continuous-time formulation, SSMs can be expressed as discrete-time models that update the hidden state S_t via a linear recurrence:

$$s_t = As_{t-1} + Bx_t, \tag{1}$$

where x_t is the input and A, B are state transition matrices. Recent SSMs aim to design the transition matrices A, B and the recurrence formulation. S4 [19], H3 [20], S5 [21], Mamba [22], and Longhorn [15] have introduced structured state transition matrices and parallel computation schemes



Training
Inference
Task : pick up the alphabet soup and place it in the basket
Longhorn
τ_1
τ_2
$\tau_{N_{\text{train}}}$
τ_q
o_1
o_2
o_T
$\mathcal{P}_{\text{train}}$
Task : pick up the orange juice and place it in the basket
Longhorn
τ_1
τ_2
$\tau_{N_{\text{test}}}$
o_1
o_2
o_T
$\mathcal{P}_{\text{test}}$
a_1
a_2
o_1^q
o_2^q
o_3^q

Figure 1: **Overview of RoboSSM**. **Training** (left): Longhorn receives N_{train} trajectories from $\mathcal{P}_{\text{train}}$ and a query trajectory for the same task. **Inference** (right): Given N_{test} trajectories from $\mathcal{P}_{\text{test}}$ containing unseen tasks, the model predicts actions and updates the environment iteratively from the initial observation embedding.

to enhance efficiency and capability. In particular, recent SSM architectures enable linear-time inference and demonstrate strong extrapolation capabilities over long-range contexts, while achieving comparable performance to Transformers in language modeling tasks.

As SSMs have evolved, their applications have expanded beyond language modeling to various other domains, including robotics. For instance, S5 is applied to reinforcement learning by allowing hidden state resets within a trajectory [23]. MAIL [24] proposes a novel imitation learning policy by leveraging Mamba. Building on these extensions, we explore using Longhorn, a recent state-of-the-art model, to perform in-context imitation learning.

3 Method

Our objective is to learn an ICIL policy π_θ that maximizes success rate on unseen tasks when conditioned on few-shot demonstrations. In this section, we describe how RoboSSM implements π_θ with Longhorn [15] and how it processes trajectory prompts during training and inference.

3.1 Architecture

RoboSSM first processes the observations with multimodal encoders. The per-step encoded observation embeddings are then passed through the Longhorn state-space block to generate actions.

Input Encoding RoboSSM encodes multimodal observations at each time step of a demonstration. At each time step, the observation includes front-view and hand-view RGB images, the robot's joint angles, and the gripper state. To prevent the model from trivially copying the actions in the prompt, we exclude actions from the input representation. We further exclude any task language instructions to force the model to attend to the in-context demonstrations. Visual data are processed by convolutional neural networks (CNNs), and proprioceptive data are embedded using multi-layer perceptrons (MLPs). The per-step features from each modality are concatenated and projected through an MLP to produce an observation embedding.

Longhorn state-space block The sequence of observation embeddings $\{x_t\}_{t=1}^T$ is fed into Longhorn, which recurrently updates a memory state matrix $s_t \in \mathbb{R}^{d\times m}$. At each time step, the input is interpreted as a key–value pair (k_t, x_t), with $x_t \in \mathbb{R}^d$ and $k_t \in \mathbb{R}^m$ obtained via a linear projection of x_t, analogous to how Transformers use keys in the attention mechanism. Longhorn then performs a recurrent update:

$$s_t = A_t \odot s_{t-1} + B_t, \tag{2}$$

where $\odot$ denotes the element-wise product, and $A_t, B_t : \mathbb{R}^d \to \mathbb{R}^{d\times m}$ are functions of x_t, defined as

$$A_t = (\mathbf{1}_{d\times m} - \varepsilon_t \otimes k_t^{\odot 2}), \quad B_t = (\varepsilon_t \otimes \mathbf{1}_m) \otimes k_t, \quad \varepsilon_{t,i} = \frac{\beta_{t,i}}{1 + \beta_{t,i} k_t^\top k_t}, \tag{3}$$

where $\otimes$ denotes the outer product and $\beta_t \in \mathbb{R}^d$ is a weighting vector.

From the updated state, we compute a context vector:

$$r_t = s_t q_t \in \mathbb{R}^d, \tag{4}$$

where $q_t \in \mathbb{R}^m$ is a query vector derived from a linear projection of x_t.

Finally, this context vector is passed through an output head to produce the corresponding action a_t.

Longhorn as an Online Learning Problem From the perspective of online learning, the recurrent form in (2) can be derived as the solution to the following online convex programming objective [25]:

$$s_t = \arg\min_{s\in\mathbb{R}^{d\times m}} \left\{ \|s - s_{t-1}\|_{\mathrm{F}}^2 + \|sk_t - x_t\|_{\mathrm{diag}(\beta_t)}^2 \right\}, \tag{5}$$

where $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm and $\beta_t \in \mathbb{R}^d$ is a weighting vector. This objective balances two competing goals inherent in online learning: the first term encourages the updated state s_t to remain close to the previous state s_{t-1}, promoting consistency over time and preventing forgetting, while the second term enforces that the current state s_t accurately reflects the new input–target pair (k_t, x_t), allowing the model to incorporate newly observed information. Together, these terms enable the model to integrate new data while retaining useful prior knowledge. The weighting vector β_t modulates this trade-off by controlling the relative importance of the current observation embedding; it is obtained by applying a sigmoid activation to a linear projection of the input x_t. In this online regression view, the use of β_t in equation (5) naturally mitigates forgetting while integrating new information, thereby enabling efficient in-context learning with long context.

3.2 In-Context Imitation Learning

Following the standard ICIL formulation, RoboSSM conditions on a context of demonstration trajectories during both training and inference. At test time, the policy adapts to new tasks based solely on the provided demonstrations, without any parameter updates.

Each input to the policy consists of a prompt and a query trajectory. The prompt $\mathcal{P}$ contains N trajectories that provide task context:

$$\mathcal{P} = [\tau_1, \tau_2, \ldots, \tau_N], \tag{6}$$

where each demonstration trajectory τ_i is a sequence of T_i observation embeddings:

$$\tau_i = \left\{ o_1^{(i)}, o_2^{(i)}, \ldots, o_{T_i}^{(i)} \right\}, \tag{7}$$

and $o_t^{(i)}$ denotes the t-th observation embedding from the i-th demonstration.

Figure 1 illustrates the training and inference procedure of the policy π_θ. At each time step t, given a prompt $\mathcal{P}$ and the sequence of query observation embeddings $o_{1:t}^q$, the policy predicts the next action for the query trajectory as:

$$a_t = \pi_\theta(\mathcal{P}; o_1^q, \ldots, o_t^q), \quad t = 1, \ldots, T_q. \tag{8}$$

Both the prompt and the query trajectory are sampled from demonstrations of the same task. During training, the output actions of both the prompt and the query are supervised using the ground-truth actions. We adopt a multi-task learning approach following ICRT [10], enabling the policy to infer the task intent from the prompt and to generalize to unseen tasks.

At inference time, the query trajectory is initialized with the first observation embedding o_0^q. The policy then iteratively outputs the next action using the contextual information in $\mathcal{P}$ and applies the action to the environment, gradually building on the resulting observations until the task is complete.

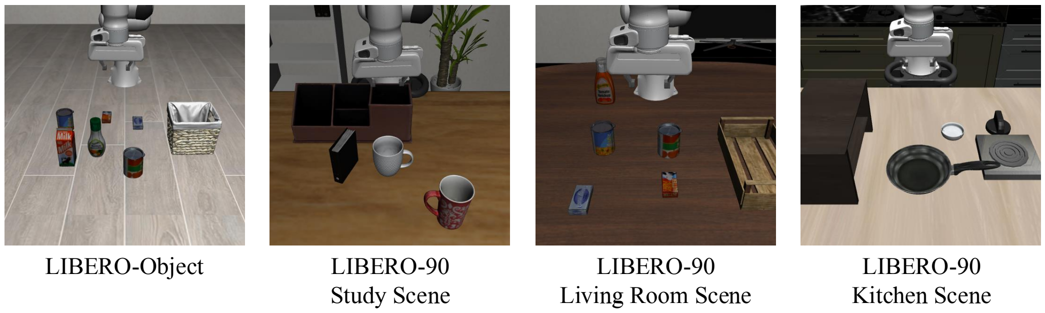


Figure 2: We evaluate RoboSSM on challenging manipulation tasks from the LIBERO benchmark. In LIBERO-90, the Study and Living Room suites each comprise 4 different scenes, while the Kitchen suite comprises 10 difference scenes.

4 Empirical Results

In this section, we evaluate whether RoboSSM can execute unseen tasks based on demonstration prompts, comparing it to previous state-of-theart ICIL methods. Section 4.1 describe the experimental setup, including dataset construction. We consider two regimes: out-of-distribution to assess length generalization with $|\mathcal{P}_{\text{test}}| > |\mathcal{P}_{\text{train}}|$ (Sec. 4.2), and in-distribution where $|\mathcal{P}_{\text{test}}| = |\mathcal{P}_{\text{train}}|$ (Sec. 4.3). We find that RoboSSM benefits from longer prompts that contain more demonstrations at test time by leveraging additional in-context information. Additionally, we compare RoboSSM to a multi-task learning policy in terms of its capability on unseen tasks (Sec. 4.4).

4.1 Experimental Setup

We conduct experiments on the **LIBERO** benchmark [16], a challenging benchmark for visuomotor robot manipulation. LIBERO consists of five task suites: *LIBERO-Object*, *LIBERO-Goal*, *LIBERO-Spatial*, *LIBERO-Long*, and *LIBERO-90*. *LIBERO-90* consists of 90 tasks, while each of the other suites contains 10 tasks. Each task contains 50 demonstration trajectories.

In our experiments, we use the full *LIBERO-Object* suite and divide *LIBERO-90* into three task suites based on scene type, including *kitchen*, *living room*, and *study* scenes, as shown in Figure 2. For all experiments, the test set $\mathcal{D}_{\text{test}}$ contains tasks that are completely disjoint from the training set $\mathcal{D}_{\text{train}}$, ensuring that models are evaluated on entirely unseen tasks. We set $|\mathcal{D}_{\text{test}}| = 2$ for all task suites, and $|\mathcal{D}_{\text{train}}| = 8$ except for the *living room* suite, where $|\mathcal{D}_{\text{train}}| = 14$.

We compare **RoboSSM** with **ICRT** [10], a Transformer-based in-context imitation learning method that employs LLaMA2-Base as its backbone. To ensure a fair comparison, we configure both backbones to have a similar number of parameters. We also implement multi-task learning (MTL) for each backbone, where **MTL-TF** uses LLaMA2-Base and **MTL-SSM** uses Longhorn. Unlike RoboSSM and ICRT, these policies take language instructions as input to specify the task.

During evaluation, we execute 20 rollouts per task, generate trajectories of up to 200 time steps, and compute the average success rate over all tasks in the suite across 6 random seeds.

We design our experiments to answer the following research questions:

- **Q1:** Can RoboSSM extrapolate to prompts composed of demonstrations that are longer than those used in training?
- **Q2:** How many training demonstrations are required for RoboSSM to effectively infer with long prompts?
- **Q3:** Can RoboSSM achieve comparable performance to Transformer-based baselines when the test-time prompt length is equal to that used in training?

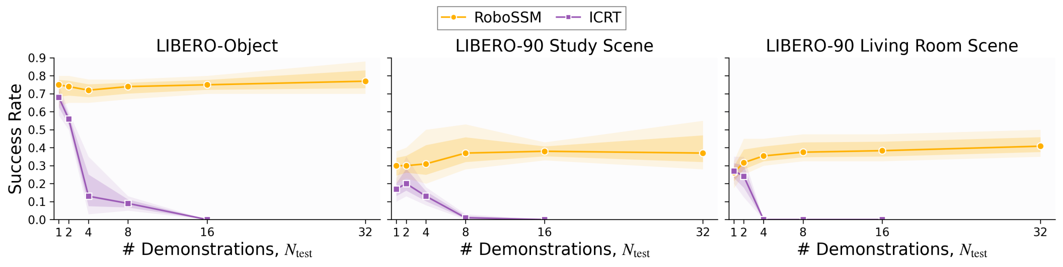


Figure 3: Comparison of RoboSSM and ICRT across test-time demonstrations (N_{test}), with both models trained with N_{train}=2. ICRT's performance drops sharply once N_{test}>N_{train}.

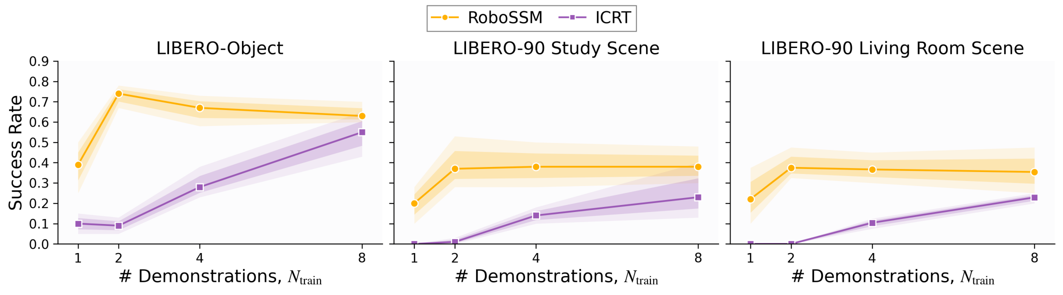


Figure 4: Results with a fixed test-time prompt $N_{\text{test}} = 8$ across models trained with different numbers of demonstrations (N_{train}).

- **Q4:** Can RoboSSM achieve superior performance on unseen tasks compared to multi-task learning?

4.2 Prompt Length Generalization

We investigate long-range in-context imitation learning with RoboSSM, focusing on prompt-length extrapolation to out-of-distribution context. In this section, we consider two approaches to making the test prompt substantially longer than the training prompt ($|\mathcal{P}_{\text{test}}| > |\mathcal{P}_{\text{train}}|$): (1) increasing the number of demonstrations and (2) applying temporal dilation to demonstrations. These experiments are conducted on LIBERO-Object, LIBERO-90 Study, and the Living Room Scene.

4.2.1 Number of demonstrations

To measure how performance changes when the number of test-time demonstrations differs from that of training, we train models with a small number of demonstrations $N_{\text{train}} = 2$. We then evaluate them on test prompts with $N_{\text{test}} \in \{1, 2, 4, 8, 16, 32\}$.

Additionally, to understand how many training demonstrations are needed for long-range in-context learning, we train with $N_{\text{train}} \in \{1, 2, 4, 8\}$ and evaluate under a fixed test-time prompt of $N_{\text{test}} = 8$.

As shown in Figure 3, RoboSSM maintains or even slightly improves its success rates as N_{test} increases beyond N_{train}. On LIBERO-Object, RoboSSM achieves its best performance at $N_{\text{test}} = 32$, which is 16x longer than the training prompt length $|\mathcal{P}_{\text{train}}|$, despite never having observed such long prompts during training. This result answers **Q1** affirmatively, showing that RoboSSM extrapolates effectively to prompts far longer than the training horizon. In contrast, ICRT degrades sharply once $N_{\text{test}} > N_{\text{train}}$ and collapses at the long prompts. This result suggests that ICRT fails to generalize to longer prompts, performing reliably only when N_{test} is equal to or shorter than N_{train}. Moreover, according to Figure 4, RoboSSM achieves strong long-range in-context learning

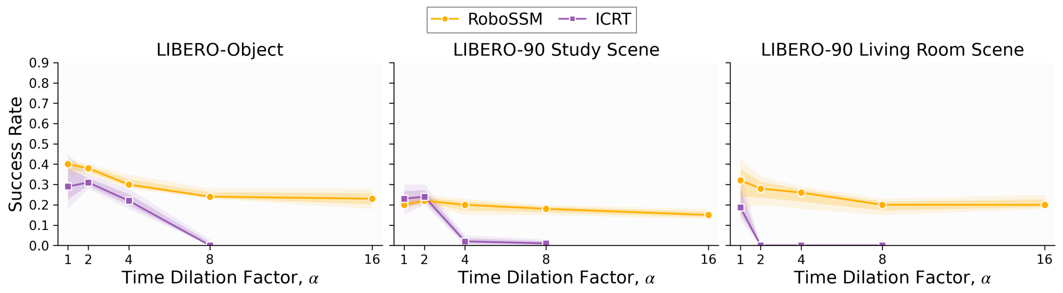


Figure 5: Comparison of RoboSSM and ICRT when evaluated on test-time prompts with temporally dilated demonstrations.

even when trained with few task examples, addressing **Q2**. ICRT performs comparably to RoboSSM only when trained with as many demonstrations as provided at test time.

4.2.2 Time dilation

In real-world scenarios, robot demonstrations may vary in execution speed due to factors such as operator latency, hardware variability, or differing task conditions. To simulate such temporal variability, we evaluate whether the model can generalize to time-dilated demonstrations at test time. We create temporally stretched demonstrations by repeating each observation embedding in the original trajectory α times, resulting in a new trajectory of length $\alpha \cdot T$, where T is the original trajectory length and $\alpha \in \{1, 2, 4, 8, 16\}$ is the dilation factor. Although models are trained with the original prompt length ($\alpha = 1$), we evaluate their robustness under extended test-time prompts, where $|\mathcal{P}_{\text{test}}| = \alpha \cdot |\mathcal{P}_{\text{train}}|$, with α up to 16.

Figure 5 indicates that RoboSSM sustains competitive success rates across increasing dilation factors α, highlighting robustness to temporal stretching and long-context extrapolation (**Q1**). By contrast, ICRT exhibits a consistent performance decay with α, culminating in failure on the longest prompts.

4.3 In-distribution ICIL

We evaluate RoboSSM and ICRT on in-distribution tasks, where $N_{\text{train}} = N_{\text{test}} \in \{1, 2, 4, 8\}$, covering varying numbers of demonstrations. This evaluation addresses **Q3** when both models operate under distributional conditions similar to those in training. Given that Longhorn exhibits performance parity with Transformers in language modeling tasks, we expect it to demonstrate comparable performance to Transformers in this setting.

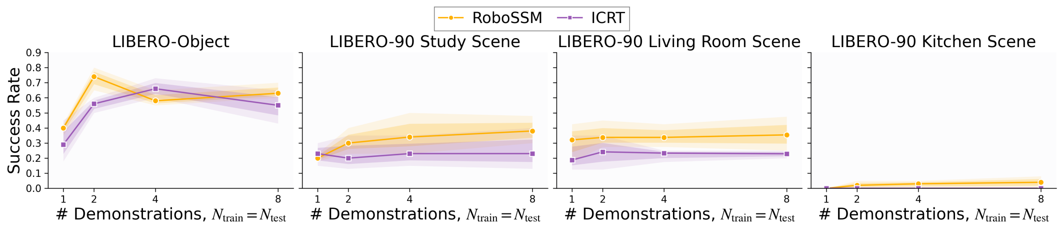


Figure 6: Results of RoboSSM and ICRT on in-context imitation learning, where the prompt consists of the same number of demonstrations during training and testing ($N_{\text{train}} = N_{\text{test}}$).

Nevertheless, as shown in Figure 6, RoboSSM consistently achieves higher success rates than ICRT across most scenarios, particularly in the LIBERO-90 Study and Living Room scene. This performance is likely due to the Longhorn architecture, whose formulation as an online regression problem enhances its in-context learning capability. However, both models struggle in the LIBERO-90 Kitchen scene, likely due to the inherent difficulty of the tasks in that suite.

Table 1: Comparison of RoboSSM and ICRT with multi-task learning policies using their respective backbones. *w/ lang* denotes that language instructions are included in the input, and *w/o lang* denotes that language instructions are excluded. Both ICIL frameworks consistently outperform the MTL baselines.

		LIBERO-90		
Method	LIBERO-Object	Study Scene	Living Room Scene	Kitchen Scene
MTL-TF	24.6 ± 1.7	14.6 ± 6.4	2.5 ± 3.5	0.0 ± 0.0
MTL-SSM	20.4 ± 3.3	15.0 ± 4.3	0.4 ± 0.9	0.0 ± 0.0
ICRT [10]				
w/o lang	**66.3 ± 3.8**	22.5 ± 6.1	23.3 ± 2.8	0.0 ± 0.0
w/ lang	48.3 ± 5.9	21.2 ± 3.8	27.1 ± 3.7	0.0 ± 0.0
RoboSSM				
w/o lang	57.9 ± 3.5	34.2 ± 3.7	**33.8 ± 4.5**	**3.8 ± 1.3**
w/ lang	45.4 ± 2.3	**35.4 ± 4.7**	29.2 ± 5.3	0.0 ± 0.0

4.4 Comparison to Multi-Task Learning

We compare RoboSSM against multi-task learning baselines, MTL-TF and MTL-SSM. During training, we set $N_{\text{train}} = 4$, and for evaluation, $N_{\text{test}} = 0$ for MTL baselines, while $N_{\text{test}} = 4$ for RoboSSM and ICRT. To enable a fair comparison with language-conditioned MTL, we additionally train and evaluate RoboSSM and ICRT with language instructions. In response to **Q4**, Table 1 shows that RoboSSM and ICRT reliably surpass the MTL baselines across their respective backbone architectures. However, the inclusion of language does not lead to improved performance on unseen tasks, as the language instructions serve to identify tasks. These results demonstrate that RoboSSM effectively handles unseen tasks without any parameter updates, and that RoboSSM achieves stronger performance compared to prior methods.

5 Conclusion

In this work, we introduce RoboSSM, a method that leverages SSMs for scalable in-context imitation learning. RoboSSM executes unseen tasks and exhibits strong prompt-length extrapolation, handling many-shot prompts and time-dilated long-horizon tasks. Across the LIBERO benchmarks, it processes prompts up to $16\times$ longer than those seen in training and outperforms Transformer-based ICIL methods, highlighting SSMs as a promising backbone for long-context robotics. Additionally, RoboSSM can support continual adaptation for lifelong learning by simply being fed demonstration prompts for new tasks, without any task-specific parameter updates Despite this potential, our study remains constrained in its ability to handle complex tasks. Moreover, comprehensively addressing novel tasks will require broader and more diverse training corpora. Future work will explore scaling datasets in both size and diversity to enable more effective generalization to novel tasks and complex task suites.

References

[1] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

[2] A. Billard, S. Calinon, R. Dillmann, and S. Schaal. Robot programming by demonstration. In *Springer handbook of robotics*, pages 1371–1394. Springer, 2008.

[3] O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[4] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[5] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

[6] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

[7] M. J. Kim, C. Finn, and P. Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

[8] J. Hu, R. Hendrix, A. Farhadi, A. Kembhavi, R. Martín-Martín, P. Stone, K.-H. Zeng, and K. Ehsani. Flare: Achieving masterful and adaptive robot policies with large-scale reinforcement learning fine-tuning. *arXiv preprint arXiv:2409.16578*, 2024.

[9] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[10] L. Fu, H. Huang, G. Datta, L. Y. Chen, W. C.-H. Panitch, F. Liu, H. Li, and K. Goldberg. In-context imitation learning via next-token prediction. *arXiv preprint arXiv:2408.15980*, 2024.

[11] N. Di Palo and E. Johns. Keypoint action tokens enable in-context imitation learning in robotics. *arXiv preprint arXiv:2403.19578*, 2024.

[12] A. D. Vuong, M. N. Vu, D. An, and I. Reid. Action tokenizer matters in in-context imitation learning. *arXiv preprint arXiv:2503.01206*, 2025.

[13] Y. Zhou, U. Alon, X. Chen, X. Wang, R. Agarwal, and D. Zhou. Transformers can achieve length generalization but not robustly. *arXiv preprint arXiv:2402.09371*, 2024.

[14] T. Dao, D. Fu, S. Ermon, A. Rudra, and C. Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35: 16344–16359, 2022.

[15] B. Liu, R. Wang, L. Wu, Y. Feng, P. Stone, and Q. Liu. Longhorn: State space models are amortized online learners. *arXiv preprint arXiv:2407.14207*, 2024.

[16] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

[17] M. Bain and C. Sammut. A framework for behavioural cloning. In *Machine intelligence 15*, pages 103–129, 1995.

[18] F. Torabi, G. Warnell, and P. Stone. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954*, 2018.

[19] A. Gu, K. Goel, and C. Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

[20] D. Y. Fu, T. Dao, K. K. Saab, A. W. Thomas, A. Rudra, and C. Ré. Hungry hungry hippos: Towards language modeling with state space models. *arXiv preprint arXiv:2212.14052*, 2022.

[21] J. T. Smith, A. Warrington, and S. W. Linderman. Simplified state space layers for sequence modeling. *arXiv preprint arXiv:2208.04933*, 2022.

[22] A. Gu and T. Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[23] C. Lu, Y. Schroecker, A. Gu, E. Parisotto, J. Foerster, S. Singh, and F. Behbahani. Structured state space models for in-context reinforcement learning. *Advances in Neural Information Processing Systems*, 36:47016–47031, 2023.

[24] X. Jia, Q. Wang, A. Donat, B. Xing, G. Li, H. Zhou, O. Celik, D. Blessing, R. Lioutikov, and G. Neumann. Mail: Improving imitation learning with mamba. *arXiv preprint arXiv:2406.08234*, 2024.

[25] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 928–936, 2003.