

EFFICIENT ROBOTIC POLICY LEARNING VIA LATENT SPACE BACKWARD PLANNING

Anonymous authors

Paper under double-blind review

ABSTRACT

Current robotic planning methods often rely on predicting multi-frame images with full pixel details. While this fine-grained approach can serve as a generic world model, it introduces two significant challenges for downstream policy learning: substantial computational costs that hinder real-time deployment, and accumulated inaccuracies that can mislead action extraction. Planning with coarse-grained subgoals partially alleviates efficiency issues. However, their forward planning schemes can still result in off-task predictions due to accumulation errors, leading to misalignment with long-term goals. This raises a critical question: Can robotic planning be both efficient and accurate enough for real-time control in long-horizon, multi-stage tasks? To address this, we propose a **Latent space Backward Planning** scheme (**LBP**), which begins by grounding the task into final latent goals, followed by recursively predicting intermediate subgoals closer to the current state. The grounded final goal enables backward subgoal planning to always remain aware of task completion, facilitating on-task prediction along the entire planning horizon. The subgoal-conditioned policy incorporates a learnable token to summarize the subgoal sequences and determines how each subgoal guides action extraction. Through extensive simulation and real-robot long-horizon experiments, we show that LBP outperforms existing fine-grained and forward planning methods, achieving SOTA performance. Project Page: <https://lbp-authors.github.io>

1 INTRODUCTION

Accurately predicting future states is crucial for many robotic planning methods in solving long-horizon, multi-stage tasks, where models must anticipate outcomes over extended temporal sequences. However, this requires balancing two conflicting objectives: (1) capturing sufficiently rich and accurate future information for task completeness, and (2) maintaining computational efficiency for real-time decision-making. Current methods face a trade-off between these objectives—those focused on long-term performance often predict multi-step future states for detailed guidance where errors accumulate rapidly while suffering from excessive computational costs, while efficiency-oriented methods compromise the semantic richness necessary for solving complex long-horizon tasks. This creates a fundamental trilemma—balancing efficiency, adequate future guidance, and future prediction accuracy—that remains unresolved and presents a significant challenge in robotic planning for real-world deployment.

To model future outcomes, one category of existing robotic planning methods (Du et al., 2024; Ajay et al., 2024; Hu et al., 2024) resorts to predicting an episode of future video as policy guidance. However, predicting consecutive frames can lead to the propagation of inaccuracies that compound over time, resulting in significant deviations from the intended final goal or generating physically inconsistent frames that confuse the downstream policies. Furthermore, modeling entire future videos requires high computational costs and puts a heavy burden on real-time inference. Obviously, predicting every detail in the future is often unnecessary for task execution, while also at the cost of computation efficiency and task-oriented consistency in predictions.

The second category of robotic planning methods focuses on predicting future subgoals (Nair & Finn, 2020; Huang et al., 2024). These coarse-grained subgoals improve planning efficiency and reduce the computational burden. However, it still adheres to the forward planning paradigm, which often leads to plans that are less aligned with distant goals, resulting in prediction errors that accumulate over time and cause off-task behavior (Kang & Kuo, 2024). To address this, recent methods have introduced reachability or optimality checks (Eysenbach et al., 2019; Nasiriany et al., 2019; Fang et al., 2022;

054 [Huang et al., 2024](#)) to correct deviations and improve on-task accuracy. However, these post-hoc
055 adjustments also add lots of complexities and do not really address the fundamental challenges.
056

057 The aforementioned two categories of methods both have some pros and cons. The video planning
058 methods provide rich future guidance but suffer from heavy training demands and inefficient inference.
059 The subgoal planning methods enjoy efficient planning but trade off long-horizon task progress
060 guidance. Apart from failing to strike a desirable balance across different considerations, all previous
061 efforts fall short in maintaining on-task prediction accuracy. How can we address the above limitations
062 and enable robots to plan efficiently and effectively through long-horizon tasks?

063 In this paper, we propose a **Backward Planning** approach in **Latent space (LBP)** for language-guided
064 robotic control as in Figure 1. LBP first trains a latent goal predictor that maps the current state and
065 language description to a distant final goal, grounding the task objective in latent image space to
066 enforce task progression. Second, LBP recursively predicts intermediate subgoals that are closer to
067 the current state, ensuring that each subgoal remains aligned with the task progression and completion.
068 These two steps mirror how humans plan in complex tasks: we begin by envisioning the desired
069 outcome based on the task objective, and then break it down into smaller, gradually manageable
070 subgoals that are closer to the present stage. The subgoal sequences in LBP track the path to the goal
071 with less redundancy, providing denser guidance in near terms while preserving task progression
072 information over the entire planning horizon. Lastly, LBP incorporates a subgoal fusion technique
073 that enables the subgoal-conditioned policy to adaptively determine how to best utilize subgoals
074 at varying distances. Collectively, LBP effectively addresses the triplet of challenges of off-task
075 planning, guidance sufficiency, and high computational costs inherent in previous methods.

076 LBP provides a lightweight planning framework for robotic policy learning with on-task subgoal
077 generation guarantee. It combines the strengths of latent planning ([Wang et al., 2023](#); [Wen et al.,
078 2023](#)) and coarse-grained subgoal planning ([Black et al., 2024](#)), drastically reducing computational
079 costs and enabling real-time deployment. Unlike previous methods that struggle with subgoal horizon
080 selection, LBP provides an informative subgoal sequence spanning the entire planning horizon
081 toward the final goal. This offers flexibility that allows the downstream policy to leverage subgoal
082 signals at varying distances. The backward planning approach further enhances on-task accuracy by
083 ensuring that the predicted subgoal sequence remains aligned with the overall task progression and
084 the ultimate objective. Through extensive evaluations in both simulation and real-robot experiments,
085 we demonstrate that LBP significantly outperforms existing methods, especially excelling on long-
086 horizon, multi-stage tasks.

087 2 RELATED WORKS

088 **Video Planning.** A significant body of research has explored video generation as planners for
089 visuomotor control ([Pertsch et al., 2020](#); [Du et al., 2024](#); [Ajay et al., 2024](#); [Hu et al., 2024](#); [Wu et al.,
090 2024](#); [Bharadhwaj et al., 2024](#)). Approaches such as UniPi ([Du et al., 2024](#)) and HiP ([Ajay et al.,
091 2024](#)) generate actions using inverse dynamics models from predicted consecutive frames, while
092 Seer ([Tian et al., 2024b](#)) and GR-1 ([Wu et al., 2024](#)) jointly predict actions and subsequent image
093 frames. Although some methods operate in latent space ([Nair et al., 2020](#); [Hu et al., 2024](#)), this line
094 of work faces significant challenges, including high computational demands and limited real-time
095 capabilities, primarily due to the need to generate every consecutive frame of the future. Most of these
096 methods operate in a forward autoregressive manner ([Wu et al., 2024](#); [Tian et al., 2024b](#)), which are
097 prone to rapid error accumulation over time, significantly complicating policy learning. In summary,
098 these approaches attempt to plan with excessive detail that is often unnecessary to visuomotor control,
099 resulting in computational inefficiency, compounded prediction errors, and challenges in effective
100 action extraction.

101 **Coarse-grained Planning.** Coarse-grained planning approaches focus on predicting intermedi-
102 ate subgoals ([Nair & Finn, 2020](#); [Wang et al., 2023](#); [Black et al., 2024](#); [Hatch et al., 2024](#)), im-
103 proving computational efficiency by avoiding the need to predict every frame of details. Goal-
104 conditioned supervised (GCSL) ([Ghosh et al., 2021](#); [Emmons et al., 2022](#)) and reinforcement learning
105 (GCRL) ([Chane-Sane et al., 2021](#); [Park et al., 2024](#)), have demonstrated that planning with intermedi-
106 ate goals can alleviate downstream policy learning burdens while enhancing long-horizon capabilities
107 in simulation benchmark tasks. However, this paradigm faces unresolved challenges, particularly
in subgoal selection, such as 1) determining appropriate prediction horizons, and 2) balancing the
number of subgoals for effective policy guidance ([Levy et al., 2019](#); [Nachum et al., 2020](#)). Distant

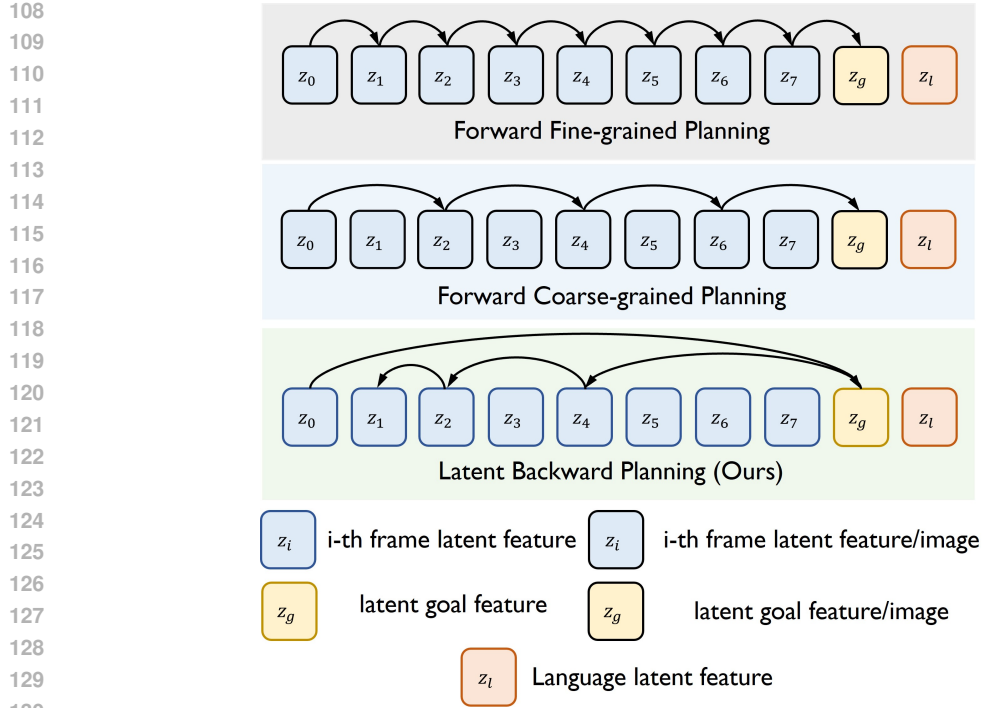


Figure 1: Illustration of latent space backward planning.

133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150

subgoals provide limited actionable information, while nearby subgoals may misalign with final task objectives. Similarly, excessive subgoals increase model complexity, whereas sparse subgoals fail to capture task progression semantics. Existing methods lack principled treatment to balance planning efficiency and long-term reliability. Furthermore, the forward planning paradigm inherently suffers from error accumulation over time, leading to off-task behavior (Kang & Kuo, 2024). Recent attempts to mitigate this through post-hoc corrections, such as reachability or optimality checks (Eysenbach et al., 2019; Nasiriany et al., 2019; Fang et al., 2022; Huang et al., 2024), which add extra complexity without addressing the fundamental limitations of forward planning.

151
152

Summary. Both fine-grained (video) and coarse-grained (subgoal) planning approaches fail to resolve the fundamental trilemma of robotic planning: achieving computational efficiency, maintaining long-horizon consistency, and ensuring prediction accuracy. These limitations highlight the need for a novel approach that appropriately balances these objectives in long-horizon, multi-stage visuomotor tasks. Inspired by the “coarse-to-fine” paradigm in computer vision (Tian et al., 2024a) and natural language processing (Wei et al., 2022), we propose a backward planning framework that predicts subgoals in reverse temporal order—from coarse to fine horizons—starting from the final goal. At every control step, this approach generates a subgoal sequence that spans the entire task horizon, providing sufficient actionable guidance efficiently while minimizing error accumulation by ensuring consistent task alignment.

153 3 PRELIMINARIES

154
155
156
157
158
159
160

We consider the problem of learning a visuomotor policy conditioned on different contexts c that reflect task objective or completeness. These contexts c can include goal images $I_g \in \mathcal{G} \subset \mathcal{I}$, language descriptions $l \in \mathcal{L}$, intermediate subgoals $w_i \in \mathcal{W} \subset \mathcal{I}$, and etc. Each video segment is defined as $\tau_i = \{(I_t, a_t)\}^{H_i}$ with H_i frames, where $I_t \in \mathcal{I}$ represents the image observation and $a_t \in \mathcal{A}$ denotes the action at time step t . Given a dataset of video segments $\mathcal{D} = \{\tau_1, \tau_2, \dots, \tau_N\}$ and a distribution over contexts $f(c|\tau)$, a conditioned policy $\pi_\theta(a_t|I_t, c)$ is trained to generate control signals in a closed-loop manner, achieving the desired behaviors align with the task description or future goals. The policy can be optimized using the following objective:

$$161 \max_{\theta} \sum_{\tau \in \mathcal{D}} \sum_{1 \leq t \leq H} \mathbb{E}_{c \sim f(c|\tau)} [\log \pi_\theta(a_t|I_t, c)] \quad (1)$$

We use the expectation over all contexts because some video segments are annotated with multiple types of task-relevant information, which can be utilized as guidance during policy learning. For instance, $f(l|\tau)$ represents the distribution of language descriptions, $f(I_g, l|\tau) = f(I_g|I_t, l)f(l|\tau)$ models the joint distribution of goal images and language descriptions, and $f(w, I_g, l|\tau) = f(w = I_{t+k}|I_t, I_g, l)f(I_g|I_t, l)f(l|\tau)$ captures the distribution of k -step future subgoals, goal images, and language descriptions. Each context provides a different level of guidance: language serves as a basic task identifier, goal images indicate task completeness, and subgoals reflect task progression toward completion. By exploring different combinations of these contexts, we can adapt the level of guidance to meet varying demands for granularity in policy learning.

4 LATENT BACKWARD PLANNING

Overview. We propose latent space backward planning (LBP), an efficient and robust planning framework for long-horizon visuomotor tasks, built upon the idea of backward subgoal prediction. We observe that existing long-horizon planning with predicted subgoals struggles with (1) planning inefficiency and (2) off-task prediction. Generating high-dimensional subgoal images poses significant challenges of computation loads, while modeling every future frame sequentially further deteriorates temporal efficiency, collectively hindering real-time real-world planning.

Thus, one of our key insights is planning in latent space with coarse-grained subgoals, enhancing planning efficiency in both spatial and temporal dimensions. While latent subgoal planning has been explored in existing works (Veerapaneni et al., 2020), they typically adopt forward planning that often fail to align subgoals with ultimate task objectives. Without accounting for task completion, subgoals can easily deviate from desired task progression, causing downstream policies to suffer from compounding errors snowballing along the planning process. Existing approaches have to introduce additional subgoal quality checks on reachability or optimality to combat the error accumulation (Nair & Finn, 2020; Eysenbach et al., 2019; Nasiriany et al., 2019; Fang et al., 2022; Huang et al., 2024), at the cost of adding unnecessary complexity and trading off efficiency, but without fundamentally resolve the underlying off-task issues.

Another key insight of ours is that we can learn a final goal predictor that grounds the ultimate task objective (i.e. language description) into latent image space (Section 4.1). Latent image space encapsulates much richer task progression information than language space, enabling backward planned subgoal sequences grounded on the predicted final goal to ensure on-task consistency (Section 4.2). The subgoal sequences can effectively capture task progression and provide flexibility for downstream policy learning to leverage envisioned subgoals at varying distances. To facilitate efficient policy training, we introduce a subgoal fusion technique that non-trivially compresses subgoal information and adaptively determines how to best utilize subgoals across different distances (Section 4.3).

4.1 GROUNDING TASK OBJECTIVE AS LATENT GOALS

Previous research suggests that visual instructions can complement language descriptions, significantly enhancing guidance performance in conditioned visuomotor policy learning (Shah et al., 2023; Li et al., 2024a; Radford et al., 2021). This synergy proves to be crucial in long-horizon, multi-stage tasks, where language descriptions often reduce to task identifiers due to their limited semantic information. In contrast, latent visual representations provide richer information about task progression, with latent visual goals offering precise specifications of the desired final scenario. However, while latent visual goals can be easily obtained through hindsight labeling during training, their test-time specification presents challenges (Lynch & Sermanet, 2021): it inherently depends on the current scenario configurations—for example, in the task “place the brown cup in front of the white cup”, the precise goal state variably depends on the initialized relative spatial locations of the two cups. Crucially, this relationship is not fixed: if the position of the white cup changes at test time, the semantic meaning of “in front of” must be re-evaluated, requiring corresponding adjustments to the target visual goal. To address this, we learn a goal prediction model f_g that estimates the latent visual goal z_g from the current observation I_t and language instruction l . Given a dataset of video segments $\mathcal{D}_z = \{\tau_i\}^N$, with latent visual state z_t in $\tau_i = \{(z_t, a_t)\}^{H_i}$ and language feature ϕ_l encoded by some pre-trained language-grounded visual encoder $(z_t, \phi_l) = \Phi(I_t, l)$, we optimize:

$$\max_{f_g} \sum_{\tau \in \mathcal{D}_z} \sum_{1 \leq t \leq H} \mathbb{E}_{p(z_g, \phi_l | \tau)} \log f_g(z_g | z_t, \phi_l) \quad (2)$$

where $p(z_g, l|\tau)$ represents the conditional context distribution ($p(c|\tau)$) of latent goals and language instructions derived from trajectory τ . This approach enables dynamic goal specification while ensuring the planning process operates within the semantically rich latent visual space.

4.2 PREDICTING SUBGOALS WITH A BACKWARD SCHEME

While the final visual goal specifies the condition of task completion, it provides limited guidance about task progression—the sequence of states required to achieve the ultimate objectives. To better capture long-horizon task progression, we predict intermediate subgoals. However, subgoal selection presents a fundamental dilemma (Park et al., 2024; Levy et al., 2019): balancing the sufficiency of subgoals for task progression against the accuracy of their prediction. Sparse subgoal predictions fail to adequately reflect task progression, while long subgoal sequences are prone to compounding prediction errors that lead to off-task behaviors deviating from the intended task goals. To address this, we begin by predicting the first subgoal w_1 from the current state z_t , final goal z_g , and language instruction ϕ_l in latent space, with the optimization objective:

$$\max_{f_w^1} \sum_{\tau \in \mathcal{D}_z} \sum_{1 \leq t \leq H} \mathbb{E}_{p(w_1, z_g, \phi_l | \tau)} \log f_w^1(w_1 | z_t, z_g, \phi_l) \quad (3)$$

To ensure sufficient long-term information, we set the first subgoal relatively close to the final goal, which maintains better alignment with task objectives yet provides less immediate guidance for policy learning. To bridge this gap, we recursively predict intermediate subgoals closer to the current state. Specifically, each subsequent subgoal w_i is predicted from the previous subgoal w_{i-1} , current state z_t , and instruction ϕ_l , forming a backward chain from coarse to fine temporal resolutions. The optimization objective for predicting subgoal w_i is given by:

$$\max_{f_w^i} \sum_{\tau \in \mathcal{D}_z} \sum_{1 \leq t \leq H} \mathbb{E}_{p(w_i, w_{i-1}, \phi_l | \tau)} \log f_w^i(w_i | z_t, w_{i-1}, \phi_l) \quad (4)$$

For convenience, let $\Gamma(w_i)$ denote the corresponding time step of subgoal w_i in the trajectory. We can thus define a fixed recursive planning coefficient $\lambda = \frac{\Gamma(w_i) - t}{\Gamma(w_{i-1}) - t}$, ($w_0 = z_g$), to govern the recursive subgoal generation for $i = 1, 2, \dots$, which represents the ratio of the temporal distance between the predicted subgoal and the current state z_t relative to the distance between the previous-level subgoal w_{i-1} and the current state z_t .

By inspecting Eq. (3) and (4), we can observe that it is possible to use a single unified model f_w for all different levels of subgoal predictors f_w^i , as they all share the same structure. This unified model is expected to predict the intermediate subgoal $z_\lambda := z_{\lceil (1-\lambda)t + \lambda k \rceil}$ between any start latent state z_t and final latent state z_k , where $1 \leq t < k$. The objective is given by:

$$\begin{aligned} \max_{f_w} \frac{1}{2} \sum_{\tau \in \mathcal{D}_z} \sum_{1 \leq t < H} \mathbb{E}_{p(z_t, z_k, \phi_l | \tau), p(\{z_{\lambda^i}\}_{i=1}^n | \tau)} & \left[\sum_{i=1}^n \log f_w(z_{\lambda^i} | z_t, z_{\lambda^{i-1}}, \phi_l) \right] \\ + \frac{1}{2} \sum_{\tau \in \mathcal{D}_z} \sum_{1 \leq t < H} \mathbb{E}_{p(z_t, z_k, \phi_l | \tau), p(\{z_{\lambda^i}\}_{i=1}^n | \tau)} & \left[\sum_{i=1}^n \log f_w(z_{\lambda^i} | z_t, \hat{z}_{\lambda^{i-1}}, \phi_l) \right] \end{aligned} \quad (5)$$

where $z_{\lambda^i} := z_{\lceil (1-\lambda^i)t + \lambda^i H \rceil} \subset \tau$, $i \in [1, \dots, n]$ denotes the ground truth subgoal and \hat{z}_{λ^i} denotes its predicted counterpart by f_w . The first term in Eq. (5) fits subgoal prediction with the ground truths z_{λ^i} in τ , capturing the actual task progression. The second term optimizes subgoal predictor f_w given its own previous predictions $\hat{z}_{\lambda^{i-1}}$ as inputs, ensuring the consistency of the recursive prediction of f_w at test-time. This recursive mechanism will suffer from much less compounding error as the training of f_w is completely supervised with ground truth data and the λ -recursion effectively reduces the planning steps from the goal state towards the current state.

As illustrated in Figure 1, this backward planning scheme generates asymmetric coarse-to-fine grained latent subgoal sequences spanning the entire task horizon, offering three key advantages over conventional methods: (1) comprehensive task progression information in subgoal sequences, providing rich and flexible guidance for policy learning; (2) improved prediction consistency with task objectives in a backward manner, reducing error accumulation compared to forward planning; and (3) computational efficiency by adopting recursion, avoiding the need for fine-grained frame-by-frame prediction.

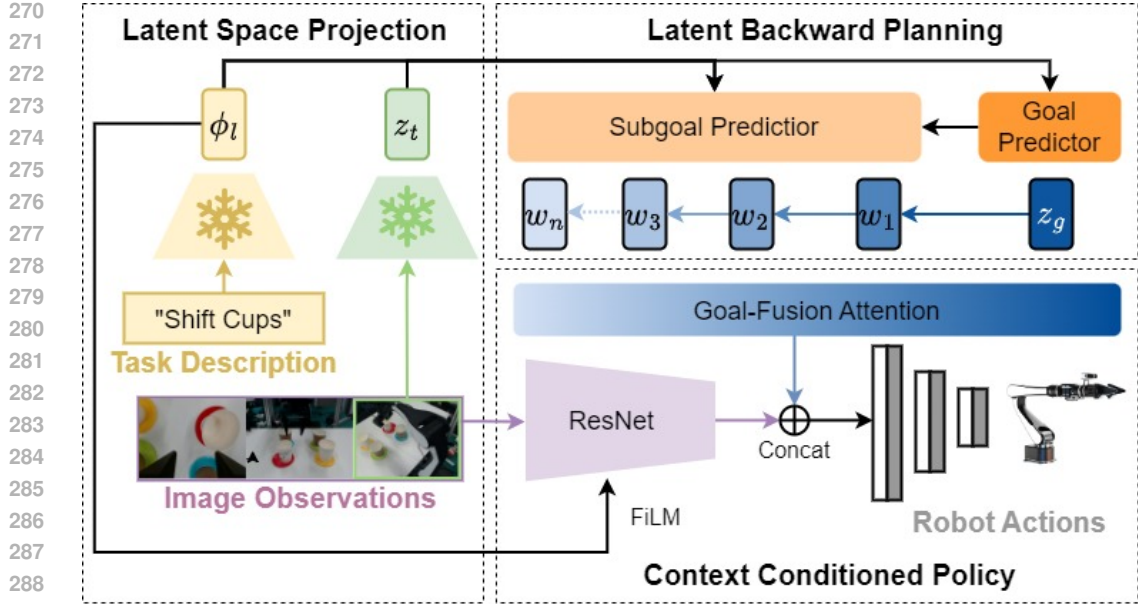


Figure 2: Overall framework architecture of LBP.

4.3 LEARNING CONTEXT CONDITIONED POLICY

The generated subgoal sequence provides rich contextual information for policy learning. Given the complete context set $c = \{w_n, \dots, w_1, z_g, \phi_l\} \in \mathbb{R}^{(n+2) \times N_z}$ derived from dataset \mathcal{D}_z , we optimize the conditioned policy through:

$$\max_{\pi} \sum_{\tau \in \mathcal{D}_z} \sum_{1 \leq t \leq H} \mathbb{E}_{c \sim p(c|\tau)} \log \pi_{\theta}(a_t | z_t, c) \quad (6)$$

However, even in latent space, the aggregated context dimensions can burden policy learning. Moreover, the policy should adaptively leverage (sub)goal information rather than treating all predictions equally, as different task execution stages require varied focus between short-term and long-term guidance. For instance, tasks requiring large movements intuitively benefit more from distant subgoals to prevent actions that hinder future progress, while precision-oriented tasks require stronger emphasis on nearby subgoals.

To address these challenges, we introduce a goal-fusion module with a Perceiver-style cross-attention (Jaegle et al., 2021) that performs both correlation discovery and dimensionality reduction. Specifically, the contexts c are queried by a trainable latent vector of size D_z : $z \in \mathbb{R}^{1 \times N_z}$, which outputs the context embeddings z_c . This design compresses all contextual tokens c into a lower-dimensional token z_c while enabling the adaptive extraction of the most relevant context information. This enables dynamic balancing of short- and long-term guidance throughout task execution, maximally leveraging the flexibility of predictions at varying distances and granularities.

4.4 PRACTICAL ALGORITHM

In the training phase, we learn a final goal predictor f_g with Eq. 2, backward subgoal predictor f_w with Eq. 5, and a conditioned policy π with Eq. 6. At each step t at test time, LBP processes the current observation I_t and language instruction l into latent state z_t and language feature ϕ_l , and then generates latent (sub)goal plans $\{w_n, \dots, w_2, w_1, z_g\}$ by f_g and f_w . Then we use the contexts c including predicted goal plans and language features ϕ_l to condition the policy $\pi(a_t | s_t, c)$ for action extraction.

The detailed architecture of our model is present in Figure 2. We implement the goal predictor f_g and the subgoal predictor f_w using two-layer MLPs and employ a cross-attention block to realize the goal-fusion model. Compared to recent planning-based methods that rely on complex pixel-level generative models, LBP demonstrates significant efficiency. For the low-level policy, we use a shared ResNet-34 (He et al., 2016a) as the backbone to extract visual features from images of different

Table 1: **LIBERO-LONG results.** For each task, we present the average performance of top-3 checkpoints. The metric ‘‘Avg. Success’’ measures the average success rate across 10 tasks. LBP outperforms baselines with higher Avg. Success and better results on most tasks. The best results are **bolded**. LIBERO-LONG tasks include: (1) put soup and sauce in basket; (2) put box and butter in basket; (3) turn on stove and put pot; (4) put bowl in drawer and close it; (5) put mugs on left and right plates; (6) pick book and place it in back; (7) put mug on plate and put pudding to right; (8) put soup and box in basket; (9) put both pots on stove; (10) put mug in microwave and close it.

Method \ Task ID	1	2	3	4	5	6	7	8	9	10	Avg. Suc \uparrow
MTACT	0.00	50.0	75.0	85.0	20.0	75.0	0.00	30.0	10.0	65.0	41.0
MVP	78.3	90.0	80.0	88.3	46.6	63.3	45.0	83.3	60.0	46.6	68.2
MPI	86.6	86.6	96.6	95.0	83.3	83.3	56.6	66.6	40.0	78.3	77.3
OpenVLA	45.0	95.0	65.0	45.0	40.0	80.0	60.0	35.0	20.0	55.0	54.0
Seer	88.3	90.0	91.6	81.6	85.0	65.0	86.7	80.0	51.6	66.6	78.6
SuSIE [†]	83.3	63.3	96.6	100.0	83.3	83.3	83.3	39.9	53.3	76.6	76.3
LBP _{SigLIP}	100.0	93.3	93.3	96.6	83.3	53.3	66.6	90.0	63.3	70.0	81.0
LBP _{DecisionNCE}	80.0	100.0	100.0	100.0	93.3	66.6	80.0	53.3	60.0	90.0	82.3

[†] Since the original SuSIE only supports single-view input, we incorporate a wrist view to reproduce it for fair comparison.

camera views, where the language embeddings are injected via FiLM conditioning layers (Perez et al., 2018a). The current visual features encoded by ResNet are then integrated with the contexts to generate actions. More details are provided in Appendix A.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

We conduct extensive experiments to evaluate the effectiveness of the proposed LBP. Specifically, we assess LBP on both the LIBERO-LONG simulation benchmark and a real-robot environment with long-horizon, multi-stage tasks. For all methods involving subgoal prediction, the planning process is solely applied to the third-person view in the LIBERO-LONG experiments and the top view in the real-world experiments. Unless otherwise stated, we adopt a three-step planning scheme (predicting a final goal and two intermediate subgoals) of LBP and set the planning coefficient $\lambda = 0.5$. Ablation studies on key framework designs and different choices of λ are provided in Section 5.3.

LIBERO-LONG experiments. LIBERO-LONG (Liu et al., 2024) consists of 10 distinct long-horizon robotic manipulation tasks that require diverse skills such as picking up objects, turning on a stove, and closing a microwave. These tasks involve multi-stage decision-making and span a variety of scenarios, making them particularly challenging. All models are trained on 50 unique expert demonstrations for each task. More details of LIBERO-LONG benchmark are provided in Appendix B.

Real-world experiments. To investigate the effectiveness of LBP in real world, we specifically design four long-horizon tasks: **Stack 3 cups**, **Move cups**, **Stack 4 cups** and **Shift cups**. Each task is decomposed into multiple sequential stages, as illustrated in Figure 3, requiring the robot to perform fundamental pick-and-place operations. These tasks establish a critical dependency where progress in subsequent stages is contingent on successful execution of preceding ones. We assess task performance using a stage-based scoring system with discrete values **{0, 25, 50, 75, 100}** for each stage, where each score corresponds to the completion progress of the current stage. A stage is assigned **100** only upon successful completion of the entire stage. All experimental evaluations are conducted with a 6 DoF AIRBOT robotic arm, together with three different views provided by Logitech C922PRO cameras. The overall environmental setups and task illustrations are shown in Figure 3. All models are trained with 200 expert demonstrations for the task *Move cups* and *Shift cups*, and 200 expert demonstrations total for *Stack 3/4 cups*. More details of experimental setups can refer to Appendix C.

Baselines. For the LIBERO-LONG benchmark, we implement the multi-task policy MTACT (Zhao et al., 2023), the general image-based pre-trained policy MVP (Xiao et al., 2022), the interaction-

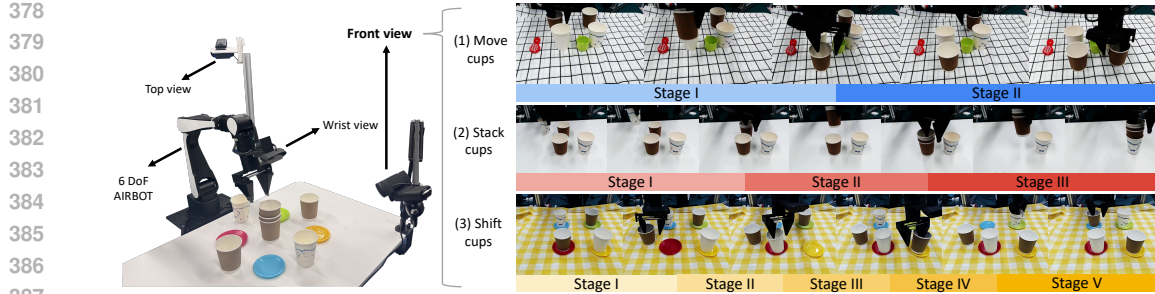


Figure 3: **Left:** the entire desktop environment setups of real-world experiments contains a 6 DoF AIRBOT arm and three Logitech C922PRO cameras with different views; **Right:** (1) *Move cups*: move both brown cups in front of the white ones; (2) *Stack cups*: stack all paper cups together; (3) *Shift cups*: shift all the paper cups to another plate, in a clockwise direction.

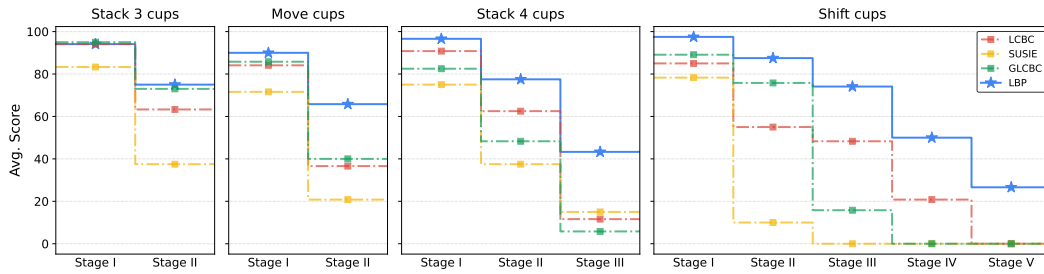


Figure 4: **Real-world main results.** We evaluate LCBC, GLCBC, SuSIE and LBP in aforementioned 4 tasks. The metric "Avg. Score" measures the average score for each stage. We observe that while LBP slightly outperforms other strong baselines at the early stages, LBP wins by a fairly large margin at the final stages of all tasks. This shows LBP significantly excels in handling long-horizon tasks.

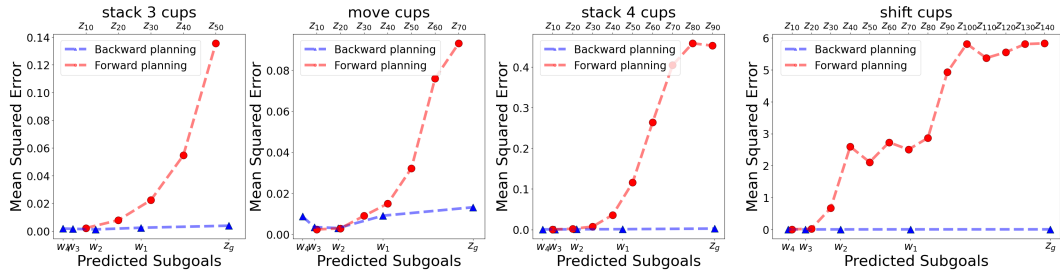


Figure 5: Mean Squared Errors (MSE) between predicted subgoals and corresponding ground truths in forward and backward planning.

oriented representation learning method MPI (Zeng et al., 2024), large-scale pretrained vision-language-action policy OpenVLA (Kim et al., 2024), an image-editing based subgoal planner SuSIE (Black et al., 2024), and the end-to-end predictive inverse dynamics model Seer (Tian et al., 2024b). For real-world experiments, we compare our LBP with SuSIE, one of the most competitive methods against LBP according to LIBERO-LONG benchmark. Also, we deploy vanilla Language Conditioned Behavior Cloning (LCBC) and Goal-and-Language Conditioned Behavior Cloning (GLCBC) for comprehensive comparison. We do not implement Seer and MPI in real-world experiments due to their inherent limitations in processing multi-view inputs. OpenVLA, MVP and MTACT are excluded from real-world evaluations because of their incompetent performance in simulation.

Metrics for long-horizon multi-stage tasks. Following Seer (Tian et al., 2024b), we report the average performance of the top three checkpoints, evaluated over 10 rollouts for each task on the LIBERO-LONG benchmark. For real-world experiments, we evaluate the last three checkpoints, with

each checkpoint being tested across 10 rollouts per task to provide an average score at each stage, offering a thorough evaluation of long-horizon capabilities.

5.2 MAIN RESULTS

Simulation Experiment Results. Table 1 presents the quantitative comparison on the LIBERO-LONG benchmark. LBP outperforms all baselines, achieving higher success rates across the majority of tasks. Specifically, LBP attains an average success rate of 81.0% in SigLIP (Zhai et al., 2023) latent space and 82.3% in DecisionNCE (Li et al., 2024b) latent space, demonstrating its flexibility in leveraging different latent representations. Compared to SuSIE and Seer, which rely on heavy generative models for high-level planning, LBP demonstrates that lightweight MLPs can achieve comparable or even better performance in long-horizon tasks. This improvement stems from the backward planning paradigm adopted in LBP, which maintains long-horizon consistency by recursively generating subgoals that preserve alignment with the final objective, ultimately enhancing both overall performance and computational efficiency on long-horizon multi-stage tasks.

Real-world experiment results. In Figure 4, we present the quantitative comparison on the real-world AIRBOT tasks. LBP consistently achieves the best performance at each stage in long-horizon tasks. Notably, in the early stages, the performance gap between different methods is relatively small. However, as the task progresses, other methods struggle due to insufficient and inconsistent guidance, leading to failures in later stages, whereas LBP maintains strong performance throughout. The results also show that GLCBC sometimes initially outperforms LCBC by incorporating additional visual goal features but suffers a sharp decline in later stages. This drop is likely due to misalignment between the given final goals and current states, which misguides the policy in long-horizon tasks, highlighting the importance of dynamically predicting the final latent goal in LBP. Additionally, we observe that SuSIE often generates hallucinated and incorrect subgoal images that confuse the low-level policy. While this issue may be less pronounced in relatively deterministic simulation environments, it significantly impacts performance in real-world settings with inherently complex disturbances and stochasticity. In contrast, LBP enables easy prediction and efficient planning in latent space with its backward philosophy.

Comparison to forward planning. To verify the effectiveness of the backward planning approach, we train conventional forward planners $f(z_{t+k}|z_t, \phi_l)$ and $f(z_{t+nk}|z_t, z_{t+(n-1)k}, \phi_l)$, $n = 2, \dots$ with the same network architecture for fair comparison. While our planners progressively predict subgoals $z_g, w_1, w_2, \dots, w_n$ in a backward manner, the forward planners predict the subgoal $k = 10$ steps ahead at a time, sequentially generating latent subgoals $z_{10}, z_{20}, \dots, z_{10n}$. We randomly sample one trajectory from each real-world AIRBOT task and calculate the Mean Squared Error (MSE) between the predicted subgoals and their corresponding ground truths for both planners as presented in Figure 5. It can be observed that the compounding errors of forward planning grow rapidly across all tasks; for the most challenging task “Shift cups”, the error becomes unbearable when predicting distant subgoals. Even more concerning is that some existing methods attempt to predict continuous frames of future images, which would exacerbate these issues further. In contrast, our method maintains an extremely low error throughout the entire planning horizon. These results highlight the advantage of our backward planning strategy, which enables accurate subgoal predictions using only a few subgoals throughout the entire trajectory. LBP not only ensures efficient prediction but also effectively reduces compounding errors, in contrast to forward planning, which requires many predictions and suffers from significant accumulated errors.

5.3 ABLATION STUDIES

In this section, we conduct ablation studies to evaluate the impact of different design choices of LBP on long-horizon performance. All models adopt DecisionNCE latent space and are tested on LIBERO-LONG.

Ablation on key model designs. We ablate the impact of the LBP planner and goal-fusion strategy, with results presented in Table 2. Removing the planner and relying solely on the low-level policy reduces the model to LCBC, resulting in a 9.3% performance drop, underscoring the necessity of subgoals predicted by LBP. Besides, replacing our goal-fusion strategy with simple average pooling causes a 10.7% decline in performance, showing that naively compressing subgoals across different horizons undermines the low-level policy. This highlights the role of our goal-fusion strategy in adaptively leveraging subgoals at different distances in a way that effectively enhances planning performance.

Table 2: Ablations on key design components of LBP on LIBERO-LONG.

	Variant	Avg. Suc \uparrow
Effectiveness of the planner	w/o planner	73.0
	ours	82.3
The strategy of goal-fusion	average pooling	71.6
	ours	82.3

Table 3: Ablations on different hyperparameter choices of LBP on LIBERO-LONG.

λ	(Sub)Goals	Avg. Suc \uparrow
-	-	73.0
-	z_g	78.6
0.5	z_g, w_1	79.6
0.5	z_g, w_1, w_2	82.3
0.5	z_g, w_1, w_2, w_3	81.5
0.75	z_g, w_1	81.6
0.75	z_g, w_1, w_2	80.0
0.75	z_g, w_1, w_2, w_3	82.3

Ablations on key hyperparameters. We perform an ablation study on two key hyperparameters: planning steps and the recursive planning coefficient λ in Table 3. We test different numbers of planning steps, where more steps correspond to predicting more subgoals. Additionally, we vary the planning coefficient λ , which controls the temporal sparsity of the subgoal sequence—larger values result in more densely packed subgoals, closer to the final goal. The main findings are: (1) Without grounding the task objective in the latent visual goal z_g , the approach reduces to LCBC, achieving an average success rate of 73%. When z_g is provided as additional context, the variant shows a significant improvement of 5.6%, demonstrating the effectiveness of leveraging the visual goal. (2) Adding subgoals w as additional contexts leads to an obvious performance improvement since it provides downstream policy with more about the future, but it is unnecessary to predict a large number of subgoals to achieve optimal results. This reflects the efficiency of our approach—unlike many planning methods that rely on generating numerous continuous waypoints, our method achieves high performance with fewer subgoals. This advantage likely arises from the backward planning philosophy of LBP, where subgoals are predicted recursively in reverse from the final goal, providing efficient yet relevant planning information closely aligned with task progression. (3) We test λ with 0.5 and 0.75, observing that LBP is robust and relatively insensitive to this hyperparameter choice.

6 CONCLUSION AND FUTURE DIRECTION

We present LBP, a novel and efficient robotic planning framework that features backward planning in the latent space to break the critical trilemma among planning efficiency, long-horizon temporal consistency, and prediction accuracy. By leveraging visual latent space for planning, LBP achieves computational efficiency while maintaining sufficient information to capture task progression. Moreover, by adopting the recursive “coarse-to-fine” backward prediction paradigm, LBP fundamentally mitigates the compounding prediction errors inherent in traditional forward planning approaches, particularly addressing the challenges of off-task prediction in long-horizon scenarios. Extensive evaluations across diverse simulated and real-world environments, including complex long-horizon and multi-stage robotic tasks, consistently demonstrate LBP’s superior performance and robustness. One promising research direction is the integration of advanced subgoal selection mechanisms, such as key-frame detection methods, to enhance the identification of informative subgoals. Another is the

540 incorporation of more sophisticated robotic encoders to construct better-structured latent spaces for
541 more efficient and accurate planning.
542

543 REFERENCES

544 Anurag Ajay, Seungwook Han, Yilun Du, Shuang Li, Abhi Gupta, Tommi Jaakkola, Josh Tenenbaum,
545 Leslie Kaelbling, Akash Srivastava, and Pulkit Agrawal. Compositional foundation models for
546 hierarchical planning. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 2
547

548 Homanga Bharadhwaj, Debidatta Dwibedi, Abhinav Gupta, Shubham Tulsiani, Carl Doersch, Ted
549 Xiao, Dhruv Shah, Fei Xia, Dorsa Sadigh, and Sean Kirmani. Gen2act: Human video generation
550 in novel scenarios enables generalizable robot manipulation. In *1st Workshop on X-Embodiment*
551 *Robot Learning*, 2024. 2

552 Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Rich Walke, Chelsea Finn, Aviral Kumar,
553 and Sergey Levine. Zero-shot robotic manipulation with pre-trained image-editing diffusion
554 models. In *The Twelfth International Conference on Learning Representations*, 2024. URL
555 <https://openreview.net/forum?id=c0chJTSbci>. 2, 8, 14
556

557 Elliot Chane-Sane, Cordelia Schmid, and Ivan Laptev. Goal-conditioned reinforcement learning
558 with imagined subgoals. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th*
559 *International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning*
560 *Research*, pp. 1430–1440. PMLR, 18–24 Jul 2021. URL <https://proceedings.mlr.press/v139/chane-sane21a.html>. 2
561

562 Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and
563 Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in Neural*
564 *Information Processing Systems*, 36, 2024. 1, 2
565

566 Scott Emmons, Benjamin Eysenbach, Ilya Kostrikov, and Sergey Levine. Rvs: What is essential for
567 offline RL via supervised learning? In *International Conference on Learning Representations*,
568 2022. URL <https://openreview.net/forum?id=S874XAIpkR->. 2

569 Ben Eysenbach, Russ R Salakhutdinov, and Sergey Levine. Search on the replay buffer: Bridging
570 planning and reinforcement learning. *Advances in neural information processing systems*, 32,
571 2019. 1, 3, 4
572

573 Kuan Fang, Patrick Yin, Ashvin Nair, and Sergey Levine. Planning to practice: Efficient online
574 fine-tuning by composing goals in latent space. In *2022 IEEE/RSJ International Conference on*
575 *Intelligent Robots and Systems (IROS)*, pp. 4076–4083. IEEE, 2022. 1, 3, 4

576 Dibya Ghosh, Abhishek Gupta, Ashwin Reddy, Justin Fu, Coline Manon Devin, Benjamin Eysenbach,
577 and Sergey Levine. Learning to reach goals via iterated supervised learning. In *International*
578 *Conference on Learning Representations*, 2021. URL <https://openreview.net/forum?id=rALA0Xo6yNJ>. 2
579

580 Philippe Hansen-Estruch, Ilya Kostrikov, Michael Janner, Jakub Grudzien Kuba, and Sergey Levine.
581 Idql: Implicit q-learning as an actor-critic method with diffusion policies. *arXiv preprint*
582 *arXiv:2304.10573*, 2023. 14
583

584 Kyle Beltran Hatch, Ashwin Balakrishna, Oier Mees, Suraj Nair, Seohong Park, Blake Wulfe, Masha
585 Itkina, Benjamin Eysenbach, Sergey Levine, Thomas Kollar, et al. Ghil-glu: Hierarchical control
586 with filtered subgoal images. In *CoRL 2024 Workshop on Mastering Robot Manipulation in a*
587 *World of Abundant Data*, 2024. 2

588 Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image
589 recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*,
590 pp. 770–778, 2016a. 6
591

592 Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image
593 recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*,
pp. 770–778, 2016b. 14

- 594 Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in*
595 *neural information processing systems*, 33:6840–6851, 2020. 14
- 596
- 597 Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil
598 Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with
599 predictive visual representations. *arXiv preprint arXiv:2412.14803*, 2024. 1, 2
- 600
- 601 Zixuan Huang, Yating Lin, Fan Yang, and Dmitry Berenson. Subgoal diffuser: Coarse-to-fine
602 subgoal generation to guide model predictive control for robot manipulation. In *IEEE International*
603 *Conference on Robotics and Automation (ICRA)*. IEEE, 2024. 1, 2, 3, 4
- 604 Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira.
605 Perceiver: General perception with iterative attention. In *International conference on machine*
606 *learning*, pp. 4651–4664. PMLR, 2021. 6
- 607
- 608 Xuhui Kang and Yen-Ling Kuo. Incorporating task progress knowledge for subgoal generation in
609 robotic manipulation through image edits. *arXiv preprint arXiv:2410.11013*, 2024. 1, 3
- 610 Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael
611 Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel,
612 Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An
613 open-source vision-language-action model. In *8th Annual Conference on Robot Learning*, 2024. 8
- 614
- 615 Andrew Levy, Robert Platt, and Kate Saenko. Hierarchical reinforcement learning with hindsight. In
616 *International Conference on Learning Representations*, 2019. URL [https://openreview.net/](https://openreview.net/forum?id=ryzEC0AcY7)
617 [forum?id=ryzEC0AcY7](https://openreview.net/forum?id=ryzEC0AcY7). 2, 5
- 618 Jianxiong Li, Zhihao Wang, Jinliang Zheng, Xiaoi Zhou, Guanming Wang, Guanglu Song, Yu Liu,
619 Jingjing Liu, Ya-Qin Zhang, Junzhi Yu, et al. Robo-mutual: Robotic multimodal task specification
620 via unimodal learning. *arXiv preprint arXiv:2410.01529*, 2024a. 4
- 621
- 622 Jianxiong Li, Jinliang Zheng, Yinan Zheng, Liyuan Mao, Xiao Hu, Sijie Cheng, Haoyi Niu, Jihao
623 Liu, Yu Liu, Jingjing Liu, et al. Decisionnce: Embodied multimodal representations via implicit
624 preference learning. *arXiv preprint arXiv:2402.18137*, 2024b. 9, 14
- 625
- 626 Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero:
627 Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information*
628 *Processing Systems*, 36, 2024. 7
- 629
- 630 Corey Lynch and Pierre Sermanet. Language conditioned imitation learning over unstructured data.
631 *Robotics: Science and Systems (RSS)*, 2021. 4
- 632
- 633 Ofir Nachum, Haoran Tang, Xingyu Lu, Shixiang Gu, Honglak Lee, and Sergey Levine. Why does
634 hierarchy (sometimes) work so well in reinforcement learning?, 2020. URL [https://openreview.](https://openreview.net/forum?id=rJgSk04tDH)
635 [net/forum?id=rJgSk04tDH](https://openreview.net/forum?id=rJgSk04tDH). 2
- 636
- 637 Suraj Nair and Chelsea Finn. Hierarchical foresight: Self-supervised learning of long-horizon tasks
638 via visual subgoal generation. In *International Conference on Learning Representations*, 2020.
639 URL <https://openreview.net/forum?id=H1gzR2VKDH>. 1, 2, 4
- 640
- 641 Suraj Nair, Silvio Savarese, and Chelsea Finn. Goal-aware prediction: Learning to model what
642 matters. In *International Conference on Machine Learning*, pp. 7207–7219. PMLR, 2020. 2
- 643
- 644 Soroush Nasiriany, Vitchyr Pong, Steven Lin, and Sergey Levine. Planning with goal-conditioned
645 policies. *Advances in neural information processing systems*, 32, 2019. 1, 3, 4
- 646
- 647 Seohong Park, Dibya Ghosh, Benjamin Eysenbach, and Sergey Levine. Hiql: Offline goal-conditioned
rl with latent states as actions. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 5
- Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018a. 7

- 648 Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual
649 reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial*
650 *intelligence*, volume 32, 2018b. 14
- 651 Karl Pertsch, Oleh Rybkin, Frederik Ebert, Shenghao Zhou, Dinesh Jayaraman, Chelsea Finn,
652 and Sergey Levine. Long-horizon visual planning with goal-conditioned hierarchical predictors.
653 *Advances in Neural Information Processing Systems*, 33:17321–17333, 2020. 2
- 654 Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal,
655 Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual
656 models from natural language supervision. In *International conference on machine learning*, pp.
657 8748–8763. PMLR, 2021. 4, 14
- 658 Rutav Shah, Roberto Martín-Martín, and Yuke Zhu. Mutex: Learning unified policies from
659 multimodal task specifications. In *7th Annual Conference on Robot Learning*, 2023. URL
660 <https://openreview.net/forum?id=PwqiaaEzJ>. 4
- 661 Keyu Tian, Yi Jiang, Zehuan Yuan, BINGYUE PENG, and Liwei Wang. Visual autoregressive
662 modeling: Scalable image generation via next-scale prediction. In *The Thirty-eighth Annual*
663 *Conference on Neural Information Processing Systems*, 2024a. URL [https://openreview.net/](https://openreview.net/forum?id=gojL67CfS8)
664 [forum?id=gojL67CfS8](https://openreview.net/forum?id=gojL67CfS8). 3
- 665 Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Pre-
666 dictive inverse dynamics models are scalable learners for robotic manipulation. *arXiv preprint*
667 *arXiv:2412.15109*, 2024b. 2, 8, 14
- 668 Rishi Veerapaneni, John D. Co-Reyes, Michael Chang, Michael Janner, Chelsea Finn, Jiajun Wu,
669 Joshua Tenenbaum, and Sergey Levine. Entity abstraction in visual model-based reinforcement
670 learning. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura (eds.), *Proceedings of*
671 *the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*,
672 pp. 1439–1456. PMLR, 30 Oct–01 Nov 2020. URL [https://proceedings.mlr.press/v100/](https://proceedings.mlr.press/v100/veerapaneni20a.html)
673 [veerapaneni20a.html](https://proceedings.mlr.press/v100/veerapaneni20a.html). 4
- 674 Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima
675 Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. *arXiv preprint*
676 *arXiv:2302.12422*, 2023. 2
- 677 Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny
678 Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in*
679 *neural information processing systems*, 35:24824–24837, 2022. 3
- 680 Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point
681 trajectory modeling for policy learning. *arXiv preprint arXiv:2401.00025*, 2023. 2
- 682 Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu,
683 Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot
684 manipulation. In *The Twelfth International Conference on Learning Representations*, 2024. 2
- 685 Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for
686 motor control. *arXiv preprint arXiv:2203.06173*, 2022. 7
- 687 Jia Zeng, Qingwen Bu, Bangjun Wang, Wenke Xia, Li Chen, Hao Dong, Haoming Song, Dong Wang,
688 Di Hu, Ping Luo, et al. Learning manipulation by predicting interaction. In *Robotics: Science and*
689 *Systems (RSS)*, 2024. 8
- 690 Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language
691 image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*,
692 pp. 11975–11986, 2023. 9, 14
- 693 Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual
694 manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023. 7
- 695
696
697
698
699
700
701

A IMPLEMENTATION DETAILS

LBP (ours) For high-level planner, we implement the goal predictor f_g and the subgoal predictor f_w using two-layer MLPs and employ two cross-attention blocks to realize the goal-fusion attention model. We employ DecisionNCE (Li et al., 2024b) and SigLIP (Zhai et al., 2023) as frozen encoders to project language instructions and images to latent space.

For the low-level policy, we use a shared ResNet-34 (He et al., 2016b) as the backbone to extract visual features from all camera view images, where the language embeddings are injected via FiLM conditioning layers (Perez et al., 2018b). The visual features, goal-fused feature, and current proprioception are then concatenated and fed into a residual MLP (Hansen-Estruch et al., 2023) to generate actions. The policy is optimized with diffusion loss to model complex distributions (Ho et al., 2020), with the denoising step fixed at 25.

For training the high-level planner, we use a batch size of 64 and train for 100k steps with the AdamW optimizer. For the low-level policy on LIBERO-LONG, we set the batch size to 64 and train for 200k steps. In the case of the low-level policy for real-world robot experiments, we increase the batch size to 128 and train for 400k steps.

SuSIE (Black et al., 2024) The high-level image-editing diffusion model is trained on video data using four A6000 GPUs. We utilize the official codebase with minimal modifications, altering only the datasets. For the LIBERO setting, we adopt a training strategy inspired by CALVIN and perform fine-tuning on the LIBERO dataset. In the Airbot setting, our approach is guided by the training paradigm of BridgeData. However, we conduct fine-tuning exclusively on the Airbot dataset. Regarding the low-level policy, we align its architecture to our model and introduced an additional channel to accept subgoal image inputs. Notably, language instructions are removed to maintain consistency with the downstream training in SuSIE.

LCBC. We implement it by directly removing our high-level planner from the architecture of LBP. The language is projected to the latent space by CLIP (Radford et al., 2021), and images are projected by a ResNet-34 (He et al., 2016b), then the semantic features is captured by a FiLM (Perez et al., 2018b) module. The low-level policy takes in these semantics, together with current proprioception, then out put a predicted diffusion-based noise.

GLCBC. The only difference between GLCBC and LCBC is the part before entering FiLM module. We chose a predefined image as final goal, then project it to a latent space with DecisionNCE image encoder. We concat the language embeddings with the final goal image embeddings, then pass this combination into the FiLM module.

Others. For LIBERO-LONG benchmark, since our experimental settings and evaluation metrics are the same with Seer, we obtain the scores of MTACT, MVP, MPI, OpenVLA and Seer from (Tian et al., 2024b).

B LIBERO-LONG BENCHMARK DETAILS

The detailed language instructions and average demonstration lengths for each task of LIBERO-LONG is shown in Table 4.

C REAL ROBOT EXPERIMENT DETAILS

Real robot dataset. We collect 200 expert demonstrations each for tasks *Move cups*, *Stack 3/4 cups* and *Shift cups*. To enhance the robustness of model trained on this dataset, we manually add some augmentation metrics (Table 5), including *Distractor augmentation*, *Target augmentation*, *Background augmentation* and *View augmentation*. Examples from the side view can be seen at Table 6. **View augmentation** always exists because the side view camera is not a fixed-position view. **Distractor augmentation** means placing various unrelated objects on the table. **Target augmentation** refers to replacing the paper cups with cups of different materials. **Background augmentation** is placing various colors of tablecloth above the clean white table.

Scoring metrics We design a scoring metric for compare the performance of different models at handling long-horizon tasks. Since every task has multiple stages, we simply make an unified scoring metric for judging within a single stage. Concretely, if the robot shows intention to catch the correct target object, a score **25** will be obtained. If the target is successfully picked up, the score will reach

Table 4: Language instructions and average lengths of LIBERO-LONG.

Task ID	Task name	Language instruction	Average demonstration length (frames)
1	put soup and sauce in basket	put both the alphabet soup and the tomato sauce in the basket	294
2	put box and butter in basket	put both the cream cheese box and the butter in the basket	260
3	turn on stove and put pot	turn on the stove and put the moka pot on it	266
4	put bowl in drawer and close it	put the black bowl in the bottom drawer of the cabinet and close it	249
5	put mugs on left and right plates	put the white mug on the left plate and put the yellow and white mug on the right plate	258
6	pick book and place it in back	pick up the book and place it in the back compartment of the caddy	189
7	put mug on plate and put pudding to right	put the white mug on the plate and put the chocolate pudding to the right of the plate	255
8	put soup and box in basket	put both the alphabet soup and the cream cheese box in the basket	270
9	put both pots on stove	put both moka pots on the stove	416
10	put mug in microwave and close it	put the yellow and white mug in the microwave and close it	305

Table 5: Dataset settings of real robot experiments.

Task name	Language instruction	w/ distractor augmentation?	w/ view augmentation?	w/ target augmentation?	w/ background augmentation?
Move cups	first put the right brown cup in front of the right white cup then put the left brown cup in front of the left white cup	✓	✓		
Stack 3/4 cups	stack the paper cups	✓	✓		
Shift cups	move each cup to a new position in a clockwise direction	✓	✓	✓	✓










50. Once the robot carries the object towards the correct destination, **75** points will be made. Last, when the object is successfully put in the desired place, a full score of **100** has been achieved. For multiple stages, a critical rule is strictly obeyed, that only when a preceding stage achieves **100** points, can the robot go to next stage. This rule makes our real robot experiments a challenging one to judge model’s performance on handling long-horizon tasks.

D ADDITIONAL RESULTS.

Except for the overview result of real robot experiments at Figure 4, we also present the whole numerical results of each stage for each task in Table 7-10.

810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863

Table 6: Demonstrations of augmentation metrics.

Augmentation metrics	Examples		
Distractor augmentation			
Target augmentation			
Background augmentation			

864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917

Table 7: Numerical results of task: *Stack 3 cups*.

Method	Stage I	Stage II	Avg. Score \uparrow
LCBC	94.1	63.3	78.7
GLCBC	95.0	74.1	84.6
SuSIE	83.3	37.5	60.4
LBP	94.1	75.0	84.6

Table 8: Numerical results of task: *Move cups*.

Method	Stage I	Stage II	Avg. Score \uparrow
LCBC	84.1	36.6	60.4
GLCBC	85.8	40.0	62.9
SuSIE	71.6	20.8	46.2
LBP	90.0	65.8	77.9

Table 9: Numerical results of task: *Stack 4 cups*.

Method	Stage I	Stage II	Stage III	Avg. Score \uparrow
LCBC	90.8	62.5	11.6	55.0
GLCBC	82.5	48.3	5.8	45.5
SuSIE	75.0	37.5	15.0	42.5
LBP	96.6	77.5	43.3	72.5

Table 10: Numerical results of task: *Shift cups*.

Method	Stage I	Stage II	Stage III	Stage IV	Stage V	Avg. Score \uparrow
LCBC	85.0	55.0	48.3	20.8	0.0	41.8
GLCBC	89.1	75.8	15.8	0.0	0.0	36.1
SuSIE	78.3	10.0	0.0	0.0	0.0	17.7
LBP	97.5	87.5	74.1	50.0	26.6	67.1