☘ Closed-Loop Visuomotor Control with Generative Expectation for Robotic Manipulation

Qingwen Bu[1,2,*] **Jia Zeng**[1,*] **Li Chen**[1,3,*] **Yanchao Yang**[3,♮] **Guyue Zhou**[4]
Junchi Yan[2] **Ping Luo**[3] **Heming Cui**[3] **Yi Ma**[3] **Hongyang Li**[1,♮]
[1] Shanghai AI Lab [2] Shanghai Jiao Tong University [3] HKU [4] Tsinghua University
[*] Equal contribution [♮] Corresponding authors

Abstract

Despite significant progress in robotics and embodied AI in recent years, deploying robots for long-horizon tasks remains a great challenge. Majority of prior arts adhere to an open-loop philosophy and lack real-time feedback, leading to error accumulation and undesirable robustness. A handful of approaches have endeavored to establish feedback mechanisms leveraging pixel-level differences or pre-trained visual representations, yet their efficacy and adaptability have been found to be constrained. Inspired by classic closed-loop control systems, we propose CLOVER, a closed-loop visuomotor control framework that incorporates feedback mechanisms to improve adaptive robotic control. CLOVER consists of a text-conditioned video diffusion model for generating visual plans as reference inputs, a measurable embedding space for accurate error quantification, and a feedback-driven controller that refines actions from feedback and initiates replans as needed. Our framework exhibits notable advancement in real-world robotic tasks and achieves state-of-the-art on CALVIN benchmark, improving by 8% over previous open-loop counterparts. Code and checkpoints are maintained at https://github.com/OpenDriveLab/CLOVER.

1 Introduction

Robotics and embodied generalists have gained enormous achievements in recent years, with the successful advancement of representation learning and visual generation in computer vision [1, 2, 3, 4], large (vision-)language models [5, 6], policy learning [7, 8], *etc*. Remarkable behavior intelligence has been demonstrated in diverse and complex single-task settings from picking up a Lego block to solving a Rubik's cube [9, 10]. However, deploying the robots for long-horizon manipulation tasks remains a long-standing challenge [11, 12].

Some literature have attempted to tackle the problem with large language models, splitting a prolonged job into detailed instructions for each minor movement [13, 14], or sub-goals. Though effective in high-level descriptions, texts could still be inadequate for detailed portrayals of the environment and robot state, leading to considerable issues under cross-morphology or multi-environments [15]. Therefore, recent efforts have started embracing vision as a universal medium to develop an embodied agent capable of planning diverse tasks through imagination and execution [16, 15, 17]. These approaches involve a generative model for predicting future videos or goal images, followed by a goal-conditioned policy for translating the visual plan into actual actions. Despite success, they adhere to an *open-loop* paradigm, *i.e.*, proceeding with a fixed sequence of actions without verifying whether the actual trajectory aligns with the planned one. For instance, when a robot is tasked to "`grab a Coke from the fridge`", current works assume that the predicted sub-goal is the visual image of the door opening, and the robot should naturally achieve the state (sub-goal) prior to grabbing the bottle. However, the lack of error measurement and real-time feedback leads to accumulative deviation, undesirable robustness, and limited adaptability—particularly inadequate for long-horizon tasks and dynamic environments [11, 18].

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-4411

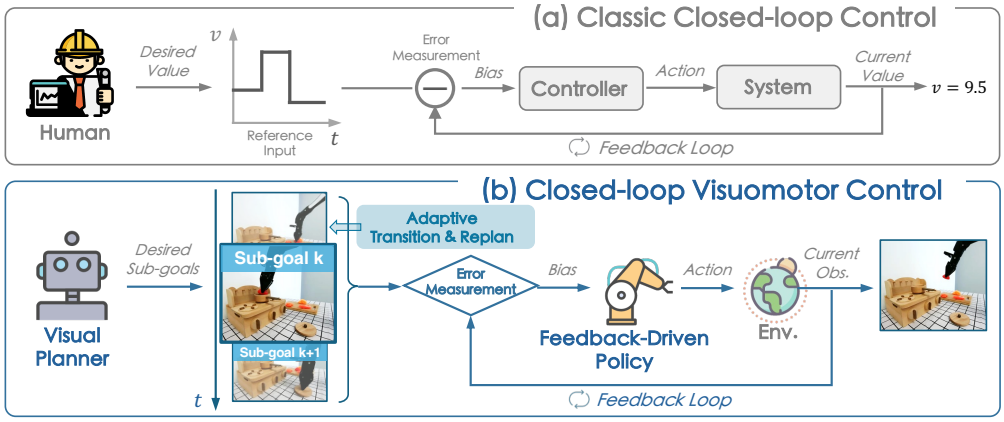


Figure 1: **Motivation.** The proposed CLOVER is inspired by the classic closed-loop control in automation systems **(a)**. Our framework **(b)** employs a visual planner to predetermine a sequence of sub-goals (Section 3.1). Then these goals guide the policy to generate actions with an error measurement strategy (Section 3.2). Within the feedback loop, it automatically replans when the sub-goal is infeasible, and adapts to to the next one upon achievement (Section 3.3).

We are inspired by the conventional closed-loop control system as depicted in Figure 1(a). It aims to regulate physical quantities such as actuator velocity by enhancing control precision via a feedback mechanism. Three major components are worth mentioning, the reference input defines desired states which could comprise multiple stages for a prolonged task; the error measurement quantifies bias between the observed state and the planned sub-goal; and the controller adjusts output to reduce the deviation (error) [19]. In fact, several works have introduced analogous feedback in robotics by measuring errors with pixel appearance [20, 21] or visual representations [22], *e.g.*, CLIP features [23], and yet their performance and adaptability is limited, leaving accurate error quantification modeling unexplored.

To this end, we propose **CLOVER**, a **CLO**sed-loop **V**isuomotor control framework with generative **E**xpectation for **R**obotic manipulation. As shown in Figure 1(b), analogy to the classic closed-loop control system, the ingredients in our version are adapted accordingly. **1) Reference inputs.** With video frames as the interface describing desired states, a text-conditioned video diffusion model generates future frames as reference inputs. To further facilitate the subsequent planning accuracy, we endow the visual planner with the ability of depth map generation, and introduce optical flow regularization to prioritize motion consistency. **2) Error measurement.** Given the limitations of pixel-wise metrics and the inadequacy of pre-trained visual representations, we propose the establishment of a measurable embedding space to realize accurate and efficient error measurement between the observed and planned states. Our state embeddings are trained using an explicit error modeling approach, which yields a strong correlation with the process of converging towards or diverging from target states. **3) Feedback-driven controller.** We present a simple yet effective control framework, comprising a controller and an error-aware adaptive control strategy. The controller is optimized via an inverse dynamics objective [24] to achieve the predefined sub-goals. To address the issue of a goal-oriented policy failing to consistently achieve the desired state, our proposed framework, CLOVER, adopts an iterative refinement strategy. It continually adjusts its actions to minimize errors, re-evaluates and replans if sub-goals are unrealistic. To sum up, our contributions are three folds:

- We introduce CLOVER, a generalizable closed-loop visuomotor control framework that incorporates a feedback mechanism to improve adaptive robotic control.
- We investigate the state-measuring attribute for latent embeddings and propose a policy by quantifying feedback errors explicitly. The error quantification settles the construction of an execution pipeline to resolve the challenge of handling uncertainties in video generation and task horizons.
- Extensive experiments in simulation and real-world robots verify the effectiveness of CLOVER. It surpasses prior state-of-the-arts by a notable margin (+8%) on CALVIN. The average length of completed tasks on real-world long-horizon manipulation nearly doubles compared to RT-1.

https://doi.org/10.52202/079017-4411

2 Related Work

Closed-loop mechanisms for robotics. Model-predictive control (MPC) is a classic and popular approach for leveraging learned dynamics for robotic control and has gained great success in learning robust closed-loop policies [25, 26, 27]. Nonetheless, many of these prior works require knowledge about the system state in the planned future which is often infeasible. Visual foresight [20, 28] integrates an auxiliary register network to calculate pixel distances between the current and goal images, consequently providing feedback. And some methods [21, 29] utilize an additional detection model with preset rules to estimate the current state. However, the adaptability of these methods is constrained to single pick-and-pull tasks and does not cover long-horizon, multi-object manipulation tasks. HiP [30] investigates the feasibility of decomposed language sub-goals and the consistency of generated video plans through training with feedback. Like other visual planners [16, 15, 17], it lacks a quantitative assessment of trajectory achievement during test-time execution. Inner Monologue [18] leverages vision-language model to provide linguistic feedback for task success detection, but the substantial size of this model hinders the efficiency during test-test execution. In contrast, CLOVER constructs a measurable latent space from pixel observations and qualitatively measures deviations from planned goals for each action, thereby incorporating real-time feedback. Furthermore, the feedback mechanism is incorporated into long-horizon manipulation tasks.

Diffusion model as a visual planner. Recently, it is trending to utilize diffusion models as visual planners to generate goal states. UniPi [15] seminally leverages internet data to train a text-conditioned video generator and uses an inverse dynamics model to estimate ultimate actions. UniSim [17] creates a universal video diffusion model for simulating interactions and training policies through generative modeling. Ajay *et al.* [30] propose compositional foundation models for hierarchical planning, including task decomposition, visual planning, and action inference. They also utilize an additional classifier to deal with the uncertainty of generation quality. SuSIE [16] utilizes an image-editing model as a high-level planner to set achievable sub-goals for a low-level controller, while ADVC [31] infers actions from predicted video content with dense correspondences. Though prior techniques can synthesize visually reasonable future sub-goals, one challenge is that the lack of consistency constraints related to geometry and motion potentially diminishes the fidelity of generated videos for policy prediction and increases generation instability. In our work, an RGB-D video prediction model is introduced and constrained by the optical flow to enhance sub-goals' reliability. Moreover, the instability of visual plans from diffusion models is rarely discussed in the aforementioned literature. Contrarily, at the core of CLOVER, we adopt an policy state estimation to detect unreachable plans by measuring the distance between consecutive frames.

3 Methodology

We aim at building a generalizable framework that integrates the closed-loop philosophy into robotic visuomotor control. The overall system is illustrated in Figure 1. Accordingly, in order to set the desired value before execution, we introduce a visual planner that generates consecutive sub-goals (Section 3.1). In Section 3.2, we detail the structure of our feedback-driven policy to decode actions, and demonstrate how to measure the deviation from the current to the goal states. Finally, the overall test-time execution pipeline of CLOVER is leveraged in Section 3.3.

3.1 Visual Planner

The visual planner is to produce a reliable sequence of future plans based on the initial observation O_0 and task descriptions c_l. Inspired by previous successful attempts [15, 31, 17], we also employ the prevailing conditional diffusion model to realize text-conditioned video generation [32]. Derived from the image diffusion model of Imagen [33, 31], our model is designed to generate future videos (*i.e.*, predicted sub-goals) spanning predetermined time frames, denoted as $\{\hat{O}_1, \hat{O}_2, ..., \hat{O}_K\}$, with $K = 8$. However, different from targets of high-resolution and meticulous structures for general generative models, visual planners for robotic manipulations highlight the need to understand spatial environments and robot movements. Therefore, designs of effectively integrating depth information and leveraging optical flow's regularization are introduced in CLOVER to generate geometry-aware and temporally coherent futures, which we describe below.

Text-conditioned RGB-D video generation. In the framework of amalgamating video prediction and goal-conditioned policy modules, generating a visual plan that precisely corresponds to the

 https://doi.org/10.52202/079017-4411

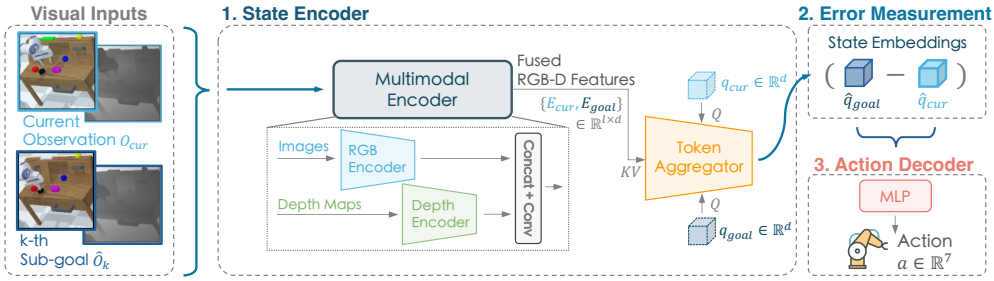


Figure 2: **Architecture of our feedback-driven policy. 1)** The state encoder takes in both current observation along with the synthesized sub-goal. A shared multimodal encoder generates fused RGB-D features, followed by two queries extracting informative features as the current and goal embeddings respectively. **2)** The discrepancy of the two state embeddings is explicitly modeled as errors. **3)** The resultant residual in error measurement is ultimately decoded to the final action.

task description is a prerequisite for accomplishing manipulation tasks. To encode language inputs, we employ the tokenizer and encoder from CLIP [23] as the basis, following [34]. In addition to the condition injection techniques outlined in Imagen [33], which integrate language embeddings into the latent space of the diffusion model directly, our model further incorporates cross-attention-based conditioning to enhance its language-following ability. Moreover, utilizing classifier-free guidance [35], the visual planner demonstrates encouraging controllability and generalization, being able to produce diverse and reasonable plans based on task descriptions (See analysis in Section 4.2).

For the vision inputs, robots operating in the 3D space face great challenges in learning from 2D observations directly [36]. Therefore, considering the ease of acquisition of depth sensory nowadays in robotics and its accurate spatial depiction of the environment, we incorporate geometric information from depth maps to assist in manipulation. To predict RGB-D videos, we adopt a simple yet effective way. Specifically, the RGB image and depth map are concatenated on the channel dimension and embedded into a unified latent space throughout all layers of the model. Compared to devising distinct branches for each modality, this yields satisfactory generation results with high consistency between modalities in practice. Moreover, the straightforward approach opens the potential for pre-training the diffusion model on large-scale RGB-only datasets to further enhance its capabilities [37, 38].

Latent regularization with optical flow. Besides the easy acquisition of the depth modality, the robotic manipulation tasks also feature in their interaction dynamics, *i.e.*, the moving robot arm and interacted objects in the environment. Though existing works [16, 30] have utilized video diffusion models for visual plan generation, they fall short in considering the essential gaps between robot manipulation and general video data adequately, particularly the static camera position and robot-initiated movements [39]. Drawing inspiration from the importance of motion cues in robot manipulation, we propose to incorporate optical flow as an explicit regularization term to further foster the classic video diffusion models for manipulation tasks. Specifically, following the end-to-end optical flow estimation framework, RAFT [40], we first build the pixel-wise correspondence map between the diffusion latent of two consecutive frames. This map is then utilized by subsequent modules to iteratively refine the optical flow estimation through lookup and update operations. Given the final estimation, our flow-based regularization term is formulated as:

$$L_{\text{reg}} = \frac{1}{K-1} \sum_{k=1}^{K-1} \|O_{k+1} - \mathcal{W}(O_k, \hat{F}_{k \to k+1})\|, \tag{1}$$

where $\hat{F}_{k \to k+1}$ is the estimated optical flow, $\mathcal{W}$ represents the wrapping function and $\{O_k, O_{k+1}\}$ are two consecutive frames in the ground-truth video. More details are provided in Appendix B.

3.2 Feedback-Driven Policy

As depicted in Figure 2, from current and desired visual inputs to the ultimate action output, our policy can be divided into the following components: 1) State Encoding: Deriving informative features from visual inputs and producing compact state embeddings that encode current and desired sub-goal states; 2) Error Measurement: Formulating the deviation from current to goal state; 3) Action Decoding: Decoding the deviation signal into the action a robot can actuate.

https://doi.org/10.52202/079017-4411

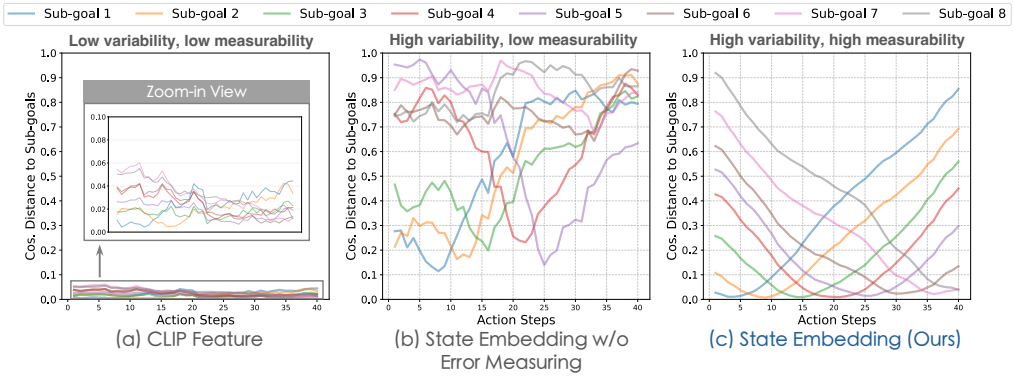


Figure 3: **Comparison on the measurement ability of different embeddings.** We visualize the cosine distance between embeddings of observations and generated sub-goals during a roll-out process. **(a)** CLIP feature [23] and **(b)** state embeddings trained without error measuring do not hold clear interrelations among frames. While **(c)** state embeddings obtained from our policy distribute reasonably in the latent space which benefits measuring the errors in feedback loops.

State encoder. To begin with, we employ a multimodal encoder to transform raw pixel inputs into enriched visual representations, which comprises two ViT-based [41] encoders for RGB and depth respectively along with a multimodal feature fusion module. The feature fusion process uses squeeze-and-excitation module [42] for channel-wise integration and selection. Subsequently, the token aggregator adaptively selects critical information pertaining to manipulation from the sequence of visual features, condensing them into a compact state embedding. Expressly, the token aggregator is built upon a multi-head attention pooling [43] with a two-layer multi-layer perceptron (MLP) performing nonlinear projection. Given fused RGB-D features $\{E_{\text{cur}},\ E_{\text{goal}}\} \in \mathbb{R}^{l\times d}$ corresponding to current and goal inputs, respectively, this process can be specified as:

$$\hat{q} = \texttt{MLP}\big[\texttt{MultiHeadAttn}(Q=q, K=V=E)\big], \tag{2}$$

where $\hat{q} \in \{\hat{q}_{\text{cur}}, \hat{q}_{\text{goal}}\}$ denotes state embeddings and $q \in \{q_{\text{cur}}, q_{\text{goal}}\}$ are queries initialized to extract visual features $E \in \{E_{\text{cur}},\ E_{\text{goal}}\}$. Here d is the hidden dimension size and l represents the visual token length. We employ shared weights for encoding both states in parallel, with the exception that two separately initialized queries are utilized to extract each state embedding. State encoder gives rise to an information bottleneck, prioritizing the encoding of manipulation-relevant features while filtering out irrelevant background details to ensure informativeness.

Error measurement. In conventional closed-loop control systems, the controller generates control signals based on the error between the desired value and feedback signals [44]. Analogously, in our visuomotor control pipeline, we explicitly model the discrepancy between the current and goal states by performing element-wise subtraction of the two corresponding state embeddings. Despite its simplicity, this approach has been proven effective in practice. It induces a strong prior that latent actions formulate the transitions between latent states [8]. More importantly, the transitions can be quantified and we note that not all embeddings hold the essential characteristics.

Notably, learning representations to distinguish among diverse instructions and visual states has been a long-standing topic [45], while there exist few works exploring their quantitative metric for action. In CLOVER, the measurement capability of state embeddings is observed by learning to act from deviation signals, which is absent in pre-trained visual encoders or policy models learned based on current observations solely, *i.e.*, behavior cloning. As illustrated in Figure 3(c), the cosine distance between state embeddings decreases together with the robot approaching each predicted sub-goal. In the meantime, the distance to the previous sub-goal increases as proceeding to the next one. Besides, the numerical range of the distance spans approximately from 0 to 0.9, thereby providing a sufficient margin for distinguishing and identifying current states. On the contrary, concerning visual representations generated by pre-trained encoders (*i.e.*, CLIP features in Figure 3(a)), there is a noticeable reduction in the range of value variations (within 6e-2) with pronounced fluctuations, although the curves show similar patterns in general. This result stands for all the pre-trained visual encoders we have studied, owing to the fact that manipulation-relevant features will be overwhelmed by immaterial background information. Furthermore, we also test employing state embeddings generated by our policy model but optimized without error measuring.

https://doi.org/10.52202/079017-4411

The resulting embeddings capture the state propagation as evidenced by the significant numerical variability (Figure 3(b)); however, they lack the capability to measure interrelations among sub-goals and exhibit poor monotonicity when reaching each sub-goal. Next, we introduce how to elevate the satisfactory error measuring feature for action decoding and adaptive feedback control autonomously.

Action decoder. To keep the framework concise and generalizable, we simply adopt an MLP to decode action outputs from error signals. We consider the action space of a 7-DoF robotic arm, encompassing the position of the end-effector $a_{\text{EE}} \in \mathbb{R}^6$ and the gripper state $a_{\text{griper}} \in \mathbb{R}^1$. Our policy is learned with an Inverse Dynamics objective $\pi_\phi(a_0|O_0, O_k)$, where it infers action a_0 based on current observation O_0 and specified sub-goals O_k. To promote the transferability of our framework, we exploit third-view RGB-D images as inputs only, with action labels as the training targets. State information such as proprioception signals or gripper-view images are not applied to facilitate manipulation. Please refer to Appendix B for further architectural and training details.

Algorithm 1: CLOVER: Test-time Execution

Input: Visual planner p_θ; Policy π_ϕ; State encoding module $g_\phi(\cdot)$; Cosine distance $D_C(\cdot,\cdot)$.
Hyper parameters : Time limit T; Distance threshold for replan and sub-goal transition $\{D_R, D_S\}$.

$t \leftarrow 0, i_{\text{sub}} \leftarrow 0$ ▷ Initialize the sub-goal selection index
while $t \leq T$ **do**
  if *Replan* ***or*** $t == 0$ **then**
    $\hat{O}_{1:K} \sim p_\theta(O_{1:K} \mid O_0, c_l)$ ▷ Generate language-conditioned sub-goals (Section 3.1)
    if $\max_{k=1,\dots,K-1}\{D_C(g_\phi(\hat{O}_k),\ g_\phi(\hat{O}_{k+1}))\} > D_R$ **then**
      Replan $\leftarrow$ *True* ▷Replan if sub-goals are unreachable
    else
      Replan $\leftarrow$ *False*
    end
  end
  if $D_C(g_\phi(O_0),\ g_\phi(\hat{O}_{i_{sub}})) < D_S$ **then**
    $i_{\text{sub}} \leftarrow i_{\text{sub}} + 1$ ▷ Transition if the current sub-goal has been reached
  end
  Sample and Execute $\hat{a} \sim \pi_\phi(a_0|O_0, \hat{O}_{i_{\text{sub}}})$ ▷ Predict and execute action (Section 3.2)
  $O_0 \leftarrow \textit{Env}(\hat{a})$ ▷ Update current observation
  $t \leftarrow t + 1$
end

3.3 CLOVER

Equipped with the error quantification capability aforementioned, we have developed a closed-loop visuomotor control framework with feedback, illustrated in Algorithm 1. Notably, our framework distinguishes itself through two key aspects: 1) It can detect and address the instability of diffusion models by initiating replanning when sub-goals are unreachable; 2) It achieves adaptive transitioning between sub-goals based on the distance measurement. In contrast to previous literature such as SuSIE [16] that sets up dataset-dependent hyperparameters to manually regulate sub-goal refreshing (usually required to be consistent with the frame intervals during training) and thus potentially limit their performance and scalability, our proposed CLOVER is agnostic to training details and adaptable to visual planners with varying intra-frame intervals. We provide illustrative examples in Appendix A to demonstrate the functionality of replanning and adaptive sub-goal transitions.

4 Experiments

4.1 Experimental Setup

Simulation tasks. We conduct the majority of our experiments using CALVIN [54], an evaluation benchmark designed for long-horizon, language-conditioned manipulation. CALVIN consists of four simulated environments (designated as A, B, C, and D), which differ in textures and object positions. Each environment comprises a Franka Emika Panda robot situated beside a desk equipped with various manipulable objects. We train policy models on demonstrations collected from environments A, B, and C, and conduct zero-shot evaluations in environment D. The evaluation protocol involves assessing model performance on a comprehensive set of 1,000 unique instruction chains, each

Table 1: **Long-horizon visuomotor control on CALVIN ABC→D.** We report success rates along with the average length of completed tasks (out of the whole 5 tasks) per evaluation sequence. CLOVER outperforms all previous methods by a notable margin. *Lang* and *All* denote whether models are trained only with the subset vision-language data. *Results reported by [16].

Method	Type	Train episodes	Task completed in a row (%) ↑ 1	2	3	4	5	Avg. Len. ↑
MCIL [46]	Language-conditioned Behaviour Cloning	All	30.4	1.3	0.2	0.0	0.0	0.31
HULC [47]		All	41.8	16.5	5.7	1.9	1.1	0.67
RT-1 [48]		Lang	53.3	22.2	9.4	3.8	1.3	0.90
RoboFlamingo [49]		Lang	82.4	61.9	46.6	33.1	23.5	2.48
GR-1 [50]		Lang	85.4	71.2	59.6	49.7	40.1	3.06
3D Diffuser Actor [51]	Diffusion Policy	Lang	92.2	78.7	63.9	51.2	41.2	3.27
UniPi* [15]	Planner + Executor	All	56.0	16.0	8.0	8.0	4.0	0.92
SuSIE [16]		All	87.0	69.0	49.0	38.0	26.0	2.69
CLOVER (Ours)		Lang	**96.0**	**83.5**	**70.8**	**57.5**	**45.4**	**3.53**

Table 2: **Performances with real-world robot tasks.** CLOVER achieves the best success rate and superior generalization capability across the board.

Method	Long-horizon task Sub-task 1	Sub-task 2	Sub-task 3	Avg. Len. ↑	Single task Pour shrimp	Stack bowls
ACT [52]	46.7	13.3	0.0	0.6	33.3	46.7
R3M [53]	53.3	20.0	0.0	0.7	46.7	53.3
RT-1 [48]	66.7	40.0	0.0	1.1	**80.0**	66.7
CLOVER (Ours)	**93.3**	**86.7**	**26.7**	**2.1**	**80.0**	**86.7**

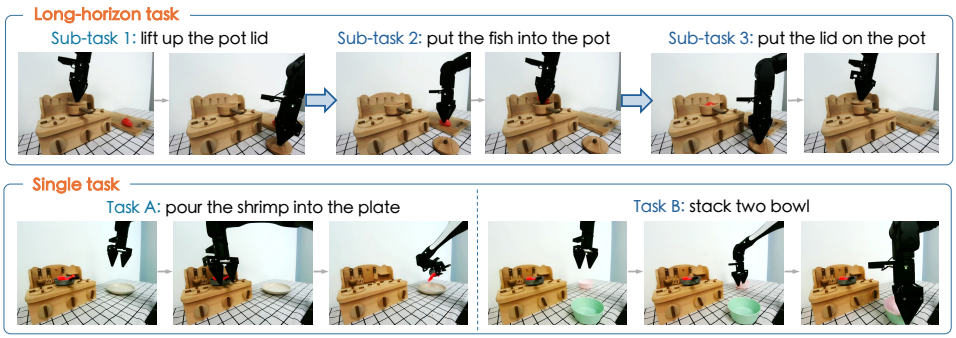


Figure 4: **Real-world robot setting.** We propose a long-horizon task encompassing three consecutive sub-tasks, where the failure of a prequel task will inevitably lead to failure of subsequent tasks. The additional single tasks are designed to validate the generalizability of CLOVER of all aspects.

comprising five distinct tasks. The CALVIN benchmark provides an extensive dataset paired with natural language annotations, thereby facilitating the training of a generalized and reliable visual planner. Detailed implementation and training protocols are provided in Appendix B.

Real-world experiments. The real-robot experiments are conducted on the AIRBOT Play robotic arm. We propose a long-horizon task comprising three consecutive sub-tasks and two additional single tasks ("Pour shrimp into plate" and "Stack two bowls", shown in Figure 4). The fish and pot lid in sub-task 2 and sub-task 3, as well as the plate and bowl in two individual tasks, are randomly placed to reflect position generalizability. All metrics are evaluated with 15 independent runs.

4.2 Main Results

Visuomotor control on CALVIN. Table 1 indicates that CLOVER achieves state-of-the-art performance on CALVIN, significantly outperforming previous methods with similar "Planner + Executor" pipelines. Without using gripper view images and proprio signals, our approach exceeds methods employing GPT-style transformers with pretraining, such as RoboFlamingo [49] and GR-1 [50]. Note that all previous methods follow the CALVIN standard evaluation protocol [54], where the simulator

https://doi.org/10.52202/079017-4411

Table 3: **Generalization evaluation.** CLOVER excels under visual distractions and dynamic scenes, while the success rate of baselines dramatically drops.

Setting	Method	Long-horizon task			
		Sub-task 1	Sub-task 2	Sub-task 3	Avg. Len. ↑
Visual Distraction	ACT [52]	13.1	0	0	0.13
	R3M [53]	20.0	0	0	0.20
	RT-1 [48]	40.0	6.7	0	0.47
	CLOVER (Ours)	**73.3**	**66.7**	**6.7**	**1.47**
Dynamic Scene	RT-1 [48]	33.0	0	0	0.33
	CLOVER (Ours)	**80.0**	**53.3**	**20.0**	**1.53**

returns the signal that marks the completion of the current task. However, such task completion signals are not accessible in real-world environments. In fact, by leveraging the advantageous properties of our state embedding, we can determine the completion of a task autonomously without the signals. We leave this exploration to Appendix A.

Figure 6 shows our diffusion model's proficiency in instruction-following, with "Slide down the switch" as a representative task from the validation set and three others randomly proposed from the CALVIN's task pool. Our planner exhibits robust generalizability, producing reliable action trajectories for the subsequent policy. More visualizations are given in Appendix D.

Manipulation with real-world robots. We present the evaluations of real-world robotic tasks in Table 2. CLOVER surpasses all baseline models by a considerable margin, especially on the long-horizon manipulation metric (+1.0 on Avg. Len.). Note that the lid knob is small and hard to grasp, which poses great challenges to policies' low-level precision. ACT [52] struggles to adjust the gripper to the right position before it should close in the first task, while CLOVER doubles the success rate. Moreover, all three baselines we test fail on the last task, which requires the robot to re-cover the pot lid that was previously placed down in Task 1. In this scenario with high uncertainty, CLOVER shows a success rate of 26.7%, indicating its stronger robustness and position generalization capability.

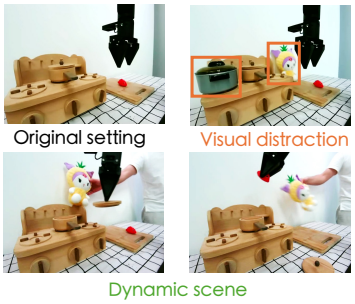


Figure 5: **Experiment setting of the generalization evaluation.** We place entirely new objects absent from training, alongside the interaction object to introduce visual distraction. We test policies under dynamic conditions by randomly placing and picking up a doll to create unpredictable visual changes.

We further study the generalizability of CLOVER under visual distractions and dynamic environments, as inllustrated in Figure 5. Table 3 lists the results of the experiment. CLVOER remains performant while manipulating under distractors, with the performance gap over baseline methods getting more pronounced. We provide qualitative analysis as shown in the Appendix Figure 17. The visual planner effectively disentangles background distractions with foreground movements and generates appropriate plans. Additionally, the feedback-driven policy proves robust to dynamic scene variations, yielding better generalizability over our leading baseline method, RT-1.

4.3 Discussion on Closed-loop v.s. Open-loop

Preliminaries. In this section, we conduct the "open-loop" experiments with the same diffusion and policy model, but do not incorporate the feedback mechanism in Algorithm 1 to facilitate adaptive replan and sub-goal transitioning, which is a common practice in previous works [16, 30, 15].

How does closed-loop work compared to open-loop? Figure 7(a) compares the manipulation performance of open- and closed-loop execution. The proposed CLOVER, which adaptively selects sub-goals by assessing the distance between current observations and given sub-goals, demonstrates a significant performance improvement of +0.44 average completed task length on CALVIN. Incorporating adaptive replanning further boosts performance by detecting unreliable visual plans and preventing error propagation. Notably, open-loop roll-out performance hinges on understanding the training specifics of both the planner and executor. Synchronizing the time interval for sub-goal transition with the frame intervals used during diffusion model training ($\Delta t = 5$ in our experiments) is needed for optimal performance. Figure 7(a) illustrates the performance deterioration when these

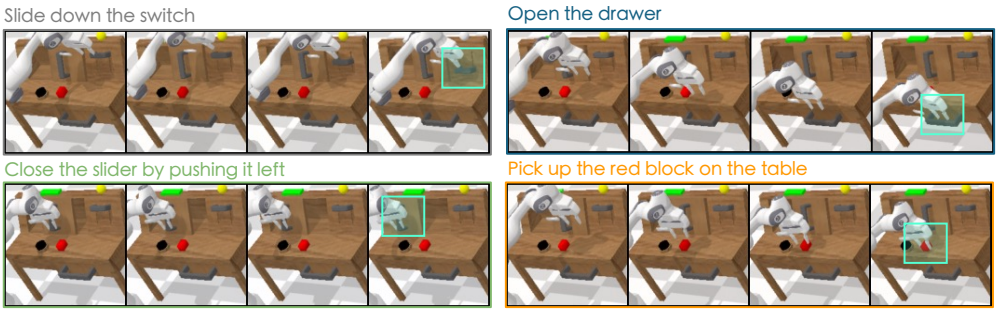


Figure 6: **Generated videos of diverse tasks conditioned on the same initial frame.** CLOVER can generate precise visual plans corresponding to the tasks, facilitating low-level executor guidance. We downsample the video by 2 and exclude depth results in visualizations for simplicity.

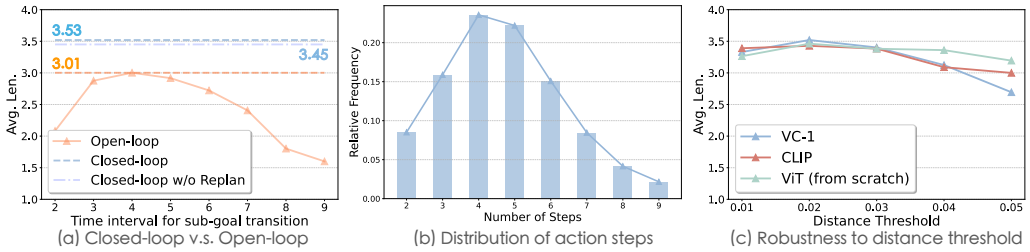


Figure 7: **Analysis and comparisons on closed-loop and open-loop roll-out on CALVIN.** (a) Comparative analysis of performance (Avg. Len.) through varying step lengths in open-loop control. Evaluations are conducted using identical models but employing different roll-out strategies. (b) The distribution of action steps taken in closed-loop roll-out to achieve each sub-goal. (c) Examination of the robustness of closed-loop control employing various visual encoders and distance thresholds.

settings are not properly aligned. Its distribution pattern aligns with the closed-loop roll-out step distribution in Figure 7(b), supporting the necessity of adaptive steps for optimal performance.

How generalizable is our feedback mechanism? We investigate the effect of different visual encoders across varying distance thresholds (D_S in Algorithm 1) used in policy models, as depicted in Figure 7(c). A larger D_S indicates a higher error tolerance. It can be seen that there is a consistent pattern across different distance thresholds for all encoders, with peak performance observed at $D_S = 0.02$. Adopting VC-1 [56] demonstrates the highest performance, while training a ViT-Base [41] encoder from scratch yields exceptional stability, with the lowest result being 3.19. The results manifest the robustness of our feedback mechanism which is independent of specific encoders and does not require customized hyperparameters for different policy models. We compare the robustness of our error measurement scheme against other representations, specifically the dense reward learning framework LIV [55] and CLIP features [23]. We maintain the same model for both the visual planner and the low-level policy, with the exception of the measurements used to determine sub-goal transitions and replanning. Our findings (as in Figure 3) indicate that CLIP features and LIV exhibit a narrower range of values, prompting us to set D_S to $2e-3$. The results presented in Table 4 show that unreliable measurements can lead to performance that is even worse than in the open-loop setting. Moreover, we do not incorporate additional contrastive objectives as in LIV, but investigate the inherent properties of the inverse dynamics-based policy.

Table 4: **Error measurments with different representations.** Our method shows exceptional cross-tasks robustness on CALVIN benchmark.

Method	Task completed in a row (%) ↑					Avg. Len. ↑
	1	2	3	4	5	
CLIP [23]	72.4	46.8	25.0	13.7	5.1	1.63
LIV [55]	70.8	48.2	29.2	18.2	10.2	1.77
CLOVER	**96.0**	**83.5**	**70.8**	**57.5**	**45.4**	**3.53**

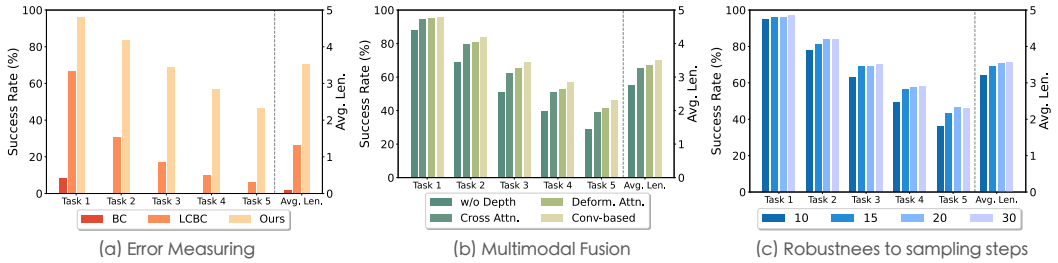
4.4 Ablation Studies

Ablations on the diffusion model (visual planner). We conduct a comparative analysis of the high-level plan (video generation) quality. Results presented in Table 5 reveal that our method outperforms AVDC [31] across all video generation metrics [57], both of which are built upon the

https://doi.org/10.52202/079017-4411

Table 5: **Comparison on video prediction.** With optical flow-based regularization and architectural modifications, CLOVER generates videos of higher quality and more accurately. †: Reproduced.

Method	SSIM ↑	PSNR ↑	LPIPS ↓	FID ↓	RMSE (Depth) ↓	Avg. Len. (CALVIN) ↑
AVDC† [31]	0.837	20.76	0.086	12.74	-	1.42
CLOVER (w/o Flow Reg.)	0.848	21.42	0.076	12.38	0.084	3.26
CLOVER (Ours)	**0.858**	**22.19**	**0.062**	**12.00**	**0.063**	**3.53**



(a) Error Measuring (b) Multimodal Fusion (c) Robustnees to sampling steps

Figure 8: **Ablations on the policy model:** (a) error measuring mechanism and (b) multimodal fusion module, with discussions on (c) robustness to different sampling steps when generating sub-goals. We report success rates and the average length of completed tasks (divided by dash lines in each plot).

Imagen framework [33], alleviating the learning and generalization burden on the policy model. Notably, AVDC struggles to generate visual plans consistent with the task description, resulting in significant performance degradation on the CALVIN benchmark. Additionally, the optical flow-based regularization not only brings a generation quality improvement across all aspects, but also significantly accelerates the training convergence. Please refer to Appendix C for further analysis.

Ablations on the policy model. Besides the ablation of feedback mechanisms in Section 4.3, we additionally evaluate the performance of CLOVER using various error measurement approaches, illustrated in Figure 8(a). Policies learned with behavior cloning (BC, Figure 3(b)) and LCBC [58] serve as baselines that do not employ error measurement, while CLOVER exceeds them by a notable margin. Figure 8(b) demonstrates that incorporating geometry information from depth data leads to a not trivial improvement on CALVIN, with convolution-based multimodal fusion modules achieving the best performance. We also examine the robustness of our policy to generation quality by varying the sampling steps of the diffusion model. As shown in Figure 8(c), increasing the sampling steps generally provides generated videos with more details but shows diminishing returns. We set the sampling step to 20 to strike a balance between performance and efficiency.

5 Conclusion

In this paper, we present a generalized closed-loop visuomotor control framework, termed CLOVER. It comprises a visual planner that specifies desired sub-goals, a policy that executes actions as planned, and a feedback-driven control strategy to realize long-horizon robotic tasks. CLOVER excels in both simulation and real-world applications, showcasing the virtue of our feedback mechanism.

Limitation and future works. We validate CLOVER for simulation and real-world experiments by training the models heavily on the corresponding data. However, emerging evidence suggests both the diffusion models and IDM-based policies exhibit out-of-distribution generalizability [59, 60, 61]. Visual planner can be trained with actionless videos, and IDM can be learned data-efficiently with random actions with corresponding observations. This points to the potential of our framework for performing few-shot and long-horizon manipulations by pre-training on web-scale datasets.

Acknowledgments

We would like to thank Jiazhi Yang, Chonghao Sima, Shenyuan Gao and all the members in OpenDriveLab for fruitful discussions and support. We thank the anonymous reviewers for their insightful feedback. Our hardware used in real-world experiments is partially supported by DISCOVER Robotics. This work was further supported by National Key R&D Program of China (2022ZD0160201), NSFC (62206172), Shanghai Committee of Science and Technology (23YF1462000), HK RIF (R7030-22),

HK ITF (GHP/169/20SZ), the Huawei Flagship Research Grants in 2021 and 2023, the HKU-SCF FinTech Academy R&D Funding Schemes in 2021 and 2022, Hong Kong RGC GRF (HKU 17208223).

References

[1] Siddharth Karamcheti, Suraj Nair, Annie S. Chen, Thomas Kollar, Chelsea Finn, Dorsa Sadigh, and Percy Liang. Language-driven representation learning for robotics. In *RSS*, 2023. 1

[2] Jia Zeng, Qingwen Bu, Bangjun Wang, Wenke Xia, Li Chen, Hao Dong, Haoming Song, Dong Wang, Di Hu, Ping Luo, Heming Cui, Bin Zhao, Xuelong Li, Yu Qiao, and Hongyang Li. Learning manipulation by predicting interaction. *arXiv preprint arXiv:2406.00439*, 2024. 1

[3] Sudeep Dasari, Mohan Kumar Srirama, Unnat Jain, and Abhinav Gupta. An unbiased look at datasets for visuo-motor pre-training. In *CoRL*, 2023. 1

[4] Yufei Wang, Zhou Xian, Feng Chen, Tsun-Hsuan Wang, Yian Wang, Katerina Fragkiadaki, Zackory Erickson, David Held, and Chuang Gan. RoboGen: Towards unleashing infinite data for automated robot learning via generative simulation. *arXiv preprint arXiv:2311.01455*, 2023. 1

[5] Yao Mu, Qinglong Zhang, Mengkang Hu, Wenhai Wang, Mingyu Ding, Jun Jin, Bin Wang, Jifeng Dai, Yu Qiao, and Ping Luo. EmbodiedGPT: Vision-language pre-training via embodied chain of thought. In *NeurIPS*, 2023. 1

[6] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023. 1

[7] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022. 1

[8] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to Control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019. 1, 5

[9] Tuomas Haarnoja, Vitchyr Pong, Aurick Zhou, Murtaza Dalal, Pieter Abbeel, and Sergey Levine. Composable deep reinforcement learning for robotic manipulation. In *ICRA*, 2018. 1

[10] Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019. 1

[11] Yafei Hu, Quanting Xie, Vidhi Jain, Jonathan Francis, Jay Patrikar, Nikhil Keetha, Seungchan Kim, Yaqi Xie, Tianyi Zhang, Zhibo Zhao, et al. Toward general-purpose robots via foundation models: A survey and meta-analysis. *arXiv preprint arXiv:2312.08782*, 2023. 1

[12] Dingzhe Li, Yixiang Jin, Hongze Yu, Jun Shi, Xiaoshuai Hao, Peng Hao, Huaping Liu, Fuchun Sun, Bin Fang, et al. What foundation models can bring for robot learning in manipulation: A survey. *arXiv preprint arXiv:2404.18201*, 2024. 1

[13] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, et al. PaLM-E: An embodied multimodal language model. In *ICML*, 2023. 1

[14] Lirui Wang, Yiyang Ling, Zhecheng Yuan, Mohit Shridhar, Chen Bao, Yuzhe Qin, Bailin Wang, Huazhe Xu, and Xiaolong Wang. GenSim: Generating robotic simulation tasks via large language models. In *ICLR*, 2024. 1

[15] Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In *NeurIPS*, 2023. 1, 3, 7, 8, 17

[16] Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Rich Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pre-trained image-editing diffusion models. In *ICLR*, 2024. 1, 3, 4, 6, 7, 8

[17] Sherry Yang, Yilun Du, Seyed Kamyar Seyed Ghasemipour, Jonathan Tompson, Leslie Pack Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *ICLR*, 2024. 1, 3

 https://doi.org/10.52202/079017-4411

[18] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner Monologue: Embodied reasoning through planning with language models. In *CoRL*, 2022. 1, 3

[19] Kuo B C and Golnaraghi M F. *Automatic control systems*. Prentice hall, 1995. 2

[20] Frederik Ebert, Sudeep Dasari, Alex X Lee, Sergey Levine, and Chelsea Finn. Robustness via retrying: Closed-loop robotic manipulation with self-supervised learning. In *CoRL*, 2018. 2, 3

[21] Connor Schenck and Dieter Fox. Visual closed-loop control for pouring liquids. In *ICRA*, 2017. 2, 3

[22] Annie S Chen, Suraj Nair, and Chelsea Finn. Learning generalizable robotic reward functions from" in-the-wild" human videos. *arXiv preprint arXiv:2103.16817*, 2021. 2

[23] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 2, 4, 5, 9, 17, 19, 21

[24] Paul Christiano, Zain Shah, Igor Mordatch, Jonas Schneider, Trevor Blackwell, Joshua Tobin, Pieter Abbeel, and Wojciech Zaremba. Transfer from simulation to real world through learning deep inverse dynamics model. *arXiv preprint arXiv:1610.03518*, 2016. 2

[25] Brijen Thananjeyan, Ashwin Balakrishna, Ugo Rosolia, Felix Li, Rowan McAllister, Joseph E Gonzalez, Sergey Levine, Francesco Borrelli, and Ken Goldberg. Safety augmented value estimation from demonstrations (saved): Safe deep model-based rl for sparse cost robotic tasks. *RA-L*, 2020. 3

[26] Zackory Erickson, Henry M Clever, Greg Turk, C Karen Liu, and Charles C Kemp. Deep haptic model predictive control for robot-assisted dressing. In *ICRA*, 2018. 3

[27] Changyeob Shin, Peter Walker Ferguson, Sahba Aghajani Pedram, Ji Ma, Erik P Dutson, and Jacob Rosen. Autonomous tissue manipulation via surgical robot using learning based model predictive control. In *ICRA*, 2019. 3

[28] Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *ICRA*, 2017. 3

[29] Monroe Kennedy, Kendall Queen, Dinesh Thakur, Kostas Daniilidis, and Vijay Kumar. Precise dispensing of liquids using visual feedback. In *IROS*, 2017. 3

[30] Anurag Ajay, Seungwook Han, Yilun Du, Shuang Li, Abhi Gupta, Tommi Jaakkola, Josh Tenenbaum, Leslie Kaelbling, Akash Srivastava, and Pulkit Agrawal. Compositional foundation models for hierarchical planning. In *NeurIPS*, 2023. 3, 4, 8

[31] Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B. Tenenbaum. Learning to Act from Actionless Videos through Dense Correspondences. In *ICLR*, 2024. 3, 9, 10, 17, 21

[32] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-A-Video: Text-to-video generation without text-video data. In *ICLR*, 2023. 3

[33] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, 2022. 3, 4, 10, 17

[34] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 4

[35] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 4

[36] David Kent, Carl Saldanha, and Sonia Chernova. Leveraging depth data in remote robot teleoperation interfaces for general object manipulation. *IJRR*, 2020. 4

[37] Tengfei Wang, Ting Zhang, Bo Zhang, Hao Ouyang, Dong Chen, Qifeng Chen, and Fang Wen. Pretraining is all you need for image-to-image translation. *arXiv preprint arXiv:2205.12952*, 2022. 4

[38] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023. 4

[39] Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, et al. Open X-Embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023. 4

[40] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. 4, 17

[41] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 5, 9, 17, 19

[42] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *ICCV*, 2018. 5, 17

[43] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set Transformer: A framework for attention-based permutation-invariant neural networks. In *ICML*, 2019. 5

[44] Kangbeom Cheon, Jaehoon Kim, Moussa Hamadache, and Dongik Lee. On replacing pid controller with deep learning controller for dc motor system. *JOACE*, 2015. 5

[45] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *ICML*, 2017. 5

[46] Corey Lynch and Pierre Sermanet. Language conditioned imitation learning over unstructured data. In *RSS*, 2021. 7

[47] Oier Mees, Lukas Hermann, and Wolfram Burgard. What matters in language conditioned robotic imitation learning over unstructured data. *RA-L*, 2022. 7

[48] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. RT-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022. 7, 8

[49] Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, Hang Li, and Tao Kong. Vision-language foundation models as effective robot imitators. In *ICLR*, 2024. 7

[50] Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *ICLR*, 2024. 7

[51] Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3D Diffuser Actor: Policy diffusion with 3d scene representations. *arXiv preprint arXiv:2402.10885*, 2024. 7

[52] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *RSS*, 2023. 7, 8, 21

[53] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3M: A universal visual representation for robot manipulation. In *CoRL*, 2022. 7, 8, 21

[54] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. CALVIN: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *RA-L*, 2022. 6, 7, 16, 18, 21

[55] Yecheng Jason Ma, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. LIV: Language-image representations and rewards for robotic control. In *ICML*, 2023. 9, 16

[56] Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Jason Ma, Claire Chen, Sneha Silwal, Aryan Jain, Vincent-Pierre Berges, Tingfan Wu, Jay Vakil, et al. Where are we in the search for an artificial visual cortex for embodied intelligence? In *NeurIPS*, 2023. 9, 17, 19, 21

[57] Qiang Fan, Wang Luo, Yuan Xia, Guozhi Li, and Daojing He. Metrics and methods of video quality assessment: a brief review. *Multimedia Tools and Applications*, 2019. 9

[58] Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. BridgeData v2: A dataset for robot learning at scale. In *CoRL*, 2023. 10

[59] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable Video Diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 10, 17

 https://doi.org/10.52202/079017-4411

[60] David Brandfonbrener, Ofir Nachum, and Joan Bruna. Inverse dynamics pretraining learns good representations for multitask imitation. In *NeurIPS*, 2024. 10

[61] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion Policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023. 10

[62] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 17

[63] Jiazhi Yang, Shenyuan Gao, Yihang Qiu, Li Chen, Tianyu Li, Bo Dai, Kashyap Chitta, Penghao Wu, Jia Zeng, Ping Luo, et al. Generalized predictive model for autonomous driving. In *CVPR*, 2024. 17

[64] Lin Sun, Kui Jia, Dit-Yan Yeung, and Bertram E Shi. Human action recognition using factorized spatio-temporal convolutional networks. In *ICCV*, 2015. 17

[65] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *ECCV*, 2018. 17

[66] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *TMLR*, 2024. 17, 19, 21

[67] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 18

[68] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 18

[69] Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via min-snr weighting strategy. In *ICCV*, 2023. 18

[70] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 18

Supplementary

A Examples of Test-time Execution with Error Measurement

In this supplemental section, we provide examples of how our proposed deviation quantification helps the autonomous test-time execution. In particular, as mentioned in Section 3.3, it can be adopted for replanning when sub-goals are infeasible or unrealistic, and transitioning to the next desired state when the robot reaches each sub-goal. In the meantime, we introduce the application of task completion assessment, though not utilized directly for performance comparisons in CALVIN.

Sub-goal replan. We first show how to identify unreachable sub-goals to mitigate the uncertainty of diffusion-based visual planners. Figure 9 gives examples of common cases showing the instability of the diffusion model. We measure the cosine distance between the state embeddings of consecutive frames generated by the model. Specifically, consistent and reasonable plans exhibit a stable intra-frame distance, typically below 0.2, whereas inconsistencies are characterized by a significant increase, often exceeding 1. Therefore, we set the distance threshold for replanning as $D_R = 1.0$ under all circumstances, which proves to work effectively in practice. The distinctive variation verifies the sensitivity and virtue of the measuring ability of state embeddings. By adaptively detecting erroneous plans and reinitializing generation, we prevent error propagation to the subsequent policy.

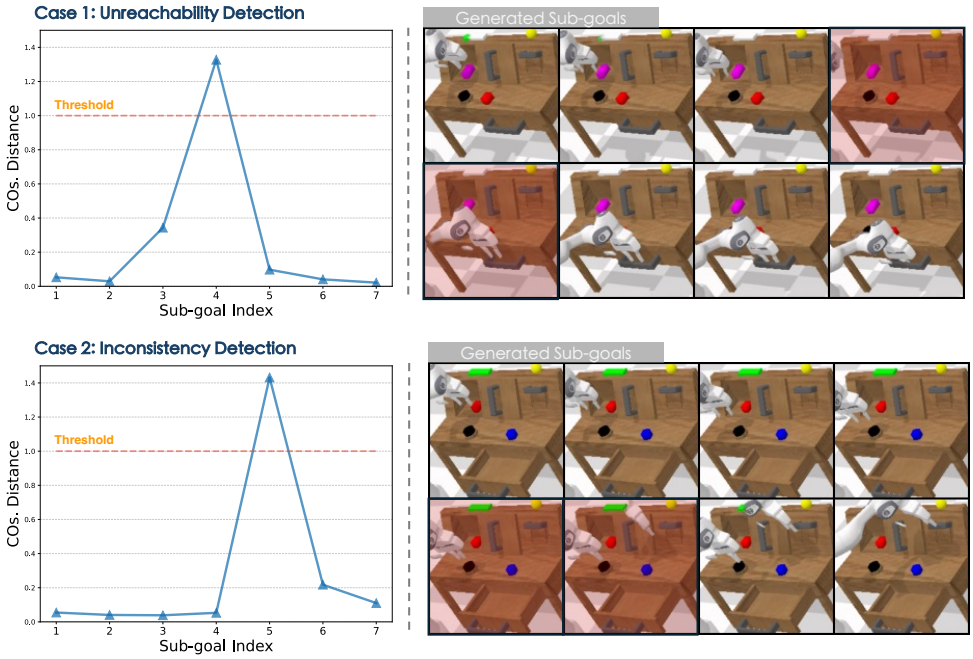


Figure 9: **Automatic identification of unreliable sub-goals generated.** We set the distance threshold for replanning as $D_R = 1.0$ under all circumstances. Ideally, the distances between generated frames remain consistent and relatively small, whereas significant variations occur during unstable generation. Examining the distance within two adjacent frames, we can detect erroneous plans generated by the diffusion model before passing them to the subsequent policy model for execution.

Sub-goal transition. Distinguished from previous works that adopt fixed action steps mentioned in Section 4.3, our framework incorporates a predefined distance threshold D_S (refer to Algorithm 1) to facilitate adaptive transitioning between sub-goals upon achievement. In Figure 10, we illustrate the distance variation between the current observation and selected goals during the roll-out process. In our experimental setup, CLOVER dynamically adjusts the number of steps for approaching each sub-goal, ranging from 1 to 7, and transitions to the next sub-goal once the distance falls below the threshold D_S. Notably, the distribution of generated goals in the latent space is uneven, and the policy does not adhere to a fixed speed in reaching targets. This is evident from the varying starting distances and changing slopes of the approach process for each sub-goal in Figure 10. These observations further underscore the necessity of a feedback mechanism to enable the adaptive sub-goal transition to mitigate error accumulation.

https://doi.org/10.52202/079017-4411

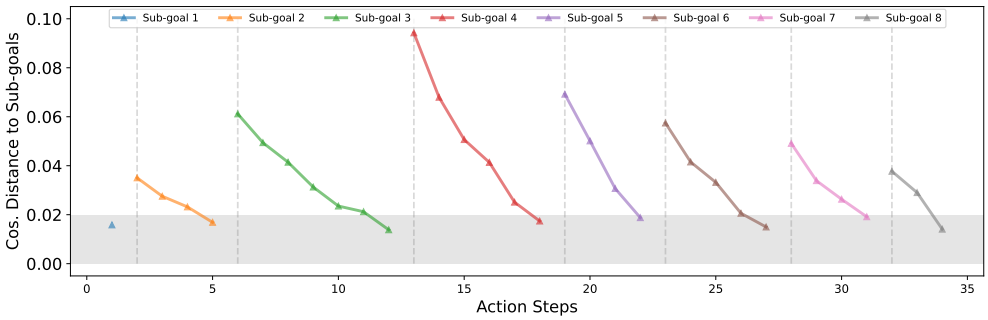


Figure 10: **Adaptive sub-goal transitions.** The cosine distance between the current observation and the selected sub-goals is plotted, with dashed gray lines indicating the transitions between sub-goals. The distance threshold D_S for sub-goal transitioning is set to 0.02. Our policy effectively reaches each assigned sub-goal and minimizes errors through an adaptive number of action steps.

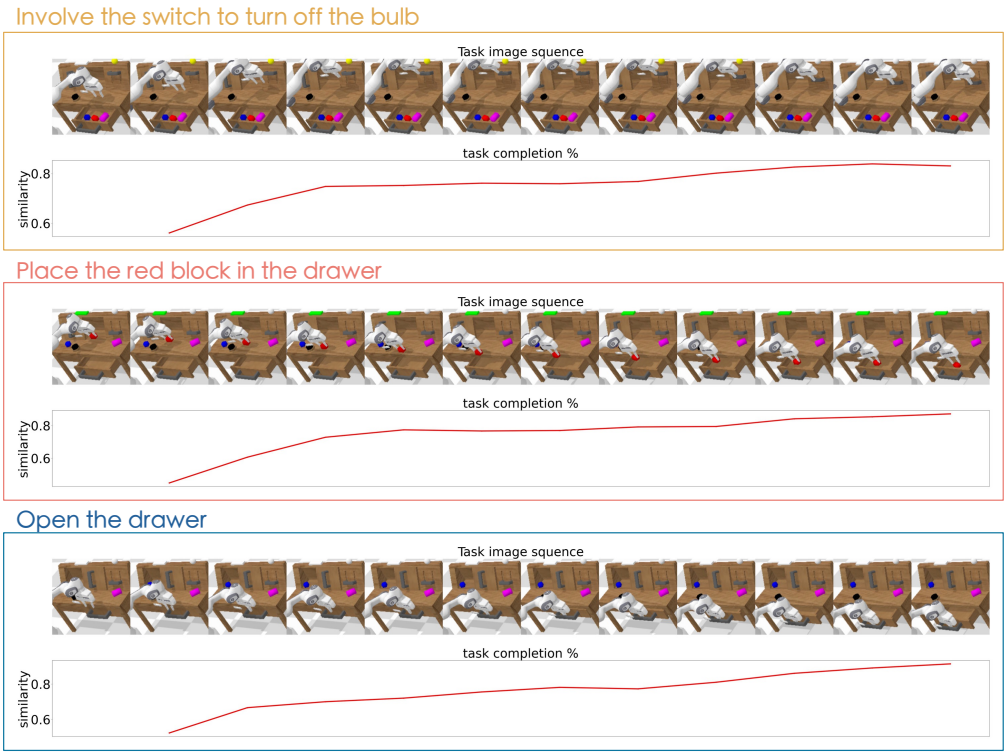


Figure 11: **Task completion score given by our value function.** As the robot approaches each sub-goal and eventually completes the task described in the text, the learned value function monotonically increments, which could be helpful for task completion assessment.

Task completion assessment. All previous methods follow the standard evaluation protocol of CALVIN [54], where the simulator updates the observation after each action roll-out and returns a signal marking the completion status of the current task. The long-horizon manipulation tasks in CALVIN consist of a sequence of five consecutive sub-tasks. Consequently, the completion status of the current sub-task serves as a valuable signal for policies to progress to the subsequent one. However, such signals are not accessible in real-world environments.

To fully exploit the advantageous properties of our state embedding as introduced in Section 3.2, we also explore using them to facilitate end-of-task assessment. Since we model latent actions for each step with the deviation of current and sub-goal embeddings, it is natural to exploit the deviation between the embeddings of the task's initial and final states to represent the latent action for the entire task. Inspired by LIV [55], we introduce a contrastive objective for language and state embeddings to establish a dense (value) reward, where a high reward for the current state represents an approaching

https://doi.org/10.52202/079017-4411

status of the entire task. Specifically, we use the subtraction of the randomly sampled intermediate state and initial state embeddings as the negative example, while the subtraction of final and initial embeddings as the positive example. We then compute the InfoNCE loss [62] with the encoded text goal to align image and language goals. Intuitively, a current state closing to the ground-truth final state resembles the overall text descriptions. Notably, the training process is built upon the pretrained policy and text encoders, with only a lightweight projector trained to align modalities. Figure 11 presents examples of using this reward design to assess a task's completion status. Such characteristic demonstrates its potential as a task completion judge.

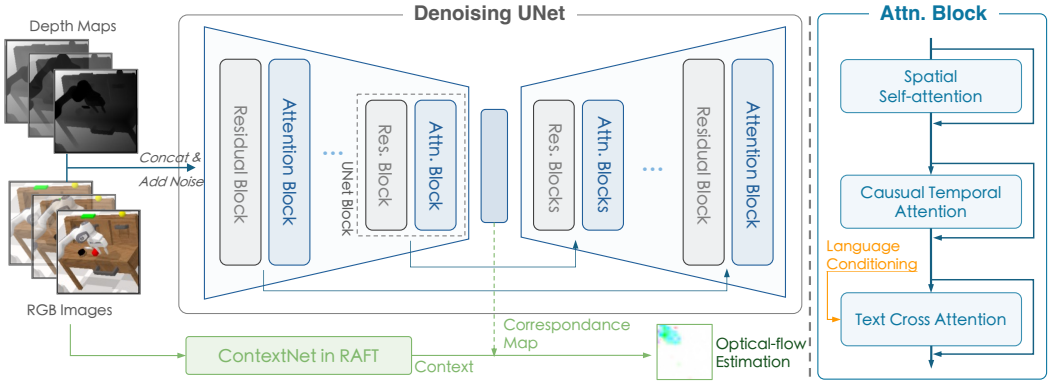


Figure 12: **The architecture of our visual planner.** We augment the original UNet proposed in Imagen [33] with casual temporal attention to improve intra-frame consistency and additional cross-attention-based language conditioning. Combining a lightweight ContextNet introduced in RAFT [40], we estimate optical flows with a correspondence map of diffusion latent embeddings.

B Implementation Details

B.1 Model Architecture

Visual planner. Figure 12 depicts the detailed architecture of our visual planner. Following [15, 31], we adopt a video diffusion model to generate synthesized sub-goals. Considering diffusion models are generally computationally expensive, we down-scale the model size to achieve a balance between the efficiency for robot manipulation tasks and high fidelity for video generation.

The visual planner needs to generate temporally coherent and consistent videos (sub-goals) to guide the subsequent executor. In line with previous studies on video generation [59, 31, 63], we concatenate the input condition frame O_0 to all the future frames $O_{1:K}$ to ensure coherency. Furthermore, to foster better temporal consistency and causal reasoning ability, causal temporal attention is adopted to encourage the full exploitation of historical information for predicting future interactions. Combining this with the factorized spatial-temporal convolution [64] in AVDC [31], our visual planner can faithfully reason about temporal causalities with improved training and inference efficiency.

To perform optical flow estimation $\hat{F}_{k\to k+1}$ with diffusion latents we employ an additional context network to encode context information from the k-th frame following RAFT [40]. In contrast to utilizing an external flow estimation model directly [31, 65], our approach incurs a minimal parameter overhead of 0.7M during training, with the optical flow estimation component being discarded during test-time control.

Feedback-driven policy. We leverage the off-the-shelf pre-trained visual encoders (*i.e.*, VC-1 [56], CLIP [23], and DINO [66]) to imbue the training process with visual priors. As for depth, we intuitively adopt a transformer-based encoder (ViT-Small [41]) to align multi-modal information in the latent space. To ensure compatibility with various pre-trained visual encoders, the input and patch size of the depth encoder are adapted to be consistent with the visual token length, thereby facilitating the widespread adoption of our policy. For fusing RGB ($\mathbf{F}_{\text{RGB}} \in \mathbb{R}^{l\times d_{\text{RGB}}}$) and depth features ($\mathbf{F}_{\text{Depth}} \in \mathbb{R}^{l\times d_{\text{Depth}}}$), we simply concatenate them on channels followed by a 3×3 convolutional layer to integrate spatial local information and reduce the channel dimension to d_{RGB}. It is then followed by a Squeeze-and-Excitation module [42] as channel attention to select important fused features.

https://doi.org/10.52202/079017-4411

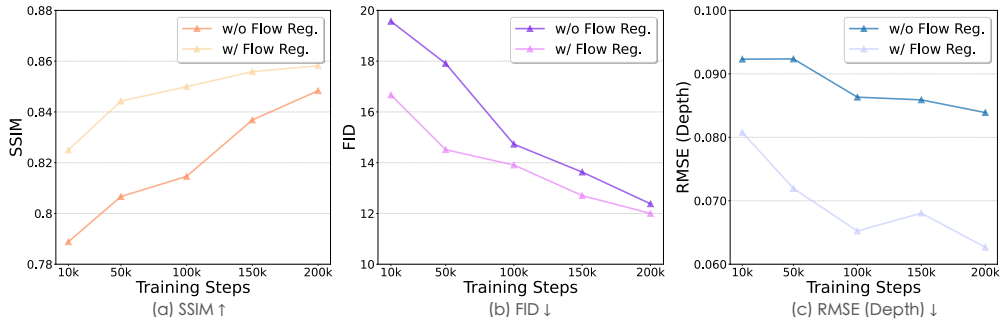


Figure 13: **Quantitative performance of video diffusion model with and without the flow-based regularization term.** The optical flow-based regularization endows the video diffusion model with more efficient training convergence and notable performance improvement on CALVIN.

For the subsequent token aggregator, we set the number of attention heads [67] to 12 in the multi-head attention module, and multiply the channel width by 4 in the MLP to enhance the representation capacity. The action decoding head is a three-layer MLP with ReLU as the activation function.

B.2 Training Protocol

Visual planner. Our diffusion model-based planner can be factorized as $p_\phi(O_{1:K} \mid O_0, c_l)$, with c_l presenting the condition given by the language. During the training process, it acts as a denoising function ϵ_ϕ predicting noises applied on future video frames $O_{1:K}$ [68]. Given the noise scheduling β_t, the training objective of the diffusion model is:

$$L_{\text{diff}} = \frac{1}{K}\sum_{k=1}^{K} \|\epsilon - \epsilon_\phi(\sqrt{1-\beta_t}\cdot O_k + \sqrt{\beta_t}\cdot\epsilon \mid t, c_l)\|^2, \tag{3}$$

where the noise $\epsilon \in \{\epsilon_{\text{RGB}}, \epsilon_{\text{Depth}}\}$ is drawn from a multivariate standard Gaussian distribution, and t represents a randomly selected diffusion step. Specifically, we sample noises separately for RGB and depth from two independent distributions. We further adopt the min-SNR weighting strategy [69] to speed up convergence. Combining the flow-based regularization term in Section 3.1, the final optimization objective of the visual planner can be formulated as:

$$L_{\text{planner}} = L_{\text{diff}} + \lambda L_{\text{reg}}, \tag{4}$$

where λ is a balancing factor and is set to 0.1 by default. In our experiments on the CALVIN [54] benchmark, we train the diffusion model for 300k iterations with a learning rate of 1e-4. Models are trained on a system equipped with 8 A100 GPUs with the batch size set as 32. We adopt the AdamW optimizer without weight decay. Besides, we track an exponential moving average (EMA) of the model parameters with a decay rate of 0.999 and use the EMA parameters at test time. For real-world experiments, we tune the diffusion model for 50,000 iterations on 50 collected demonstrations. Due to hardware limitations, we are not able to collect depth data in a real environment, so the model generates RGB images only.

For test-time execution, the DDIM sampler [70] is employed, with 20 sampling steps to strike a balance between efficiency and quality. The text guidance weight is set to 4 for generating visual plans that align with linguistic descriptions.

Feedback-driven policy. To optimize the policy model, we leverage mean squared error and binary cross-entropy loss to supervise the end-effector's position $a_{\text{EE}} \in \mathbb{R}^6$ and gripper state $a_{\text{gripper}} \in \mathbb{R}^1$, respectively. In each training episode, two frames with an interval ranging from 1 to $k_{\text{max}} = 5$ are sampled as inputs to enhance the model's robustness. We train the policy on ABC training split of CALVIN for 10 epochs with a batch size of 128. Only the relative cartesian action of a single timestamp is used for training. The training process takes around 10 hours on 8 A100 GPUs.

C Extended Ablation Studies

Optical flow based regularization. We investigate the effect of optical flow-based regularization in model convergence, besides the results reported in Table 5. As depicted in Figure 13, our analysis

Table 6: **Modeling Inverse Dynamics with different visual encoders.** CLVOER achieves promising results on CALVIN with different visual encoders of varying sizes. *: Trained from scratch.

Encoders	Params.	Task completed in a row (%) ↑					Avg. Len. ↑
		1	2	3	4	5	
ViT-S* [41]	22M	96.0	80.2	64.4	52.6	41.4	3.35
ViT-B* [41]	86M	**96.6**	**83.8**	67.6	55.0	43.2	3.46
CLIP [23]	86M	94.0	81.2	68.0	56.2	43.4	3.43
DINOv2 [66]	307M	94.8	80.4	67.0	55.4	43.8	3.41
VC-1 [56]	86M	96.0	83.5	**70.8**	**57.5**	**45.4**	**3.53**

Figure 14: **Visualization on attention maps of the policy model.** The policy model demonstrates the ability to direct attention towards the end-effector and the object it interacts with, without the need for explicit supervision during the learning process.

illustrates the impact on model performance across different training phases. The regularization significantly boosts the model's convergence, evidenced by consistently improved metrics. Notably, the visual planner trained with 10k iterations outperforms its 100k iterations counterpart in terms of SSIM and RMSE. Furthermore, the final model, trained over 200k iterations, demonstrates superior video quality with a 0.38 decrease in FID and 0.02 in RMSE. By leveraging the pixel-space correlation to regularize the diffusion model's latent space representations, the regularization term encourages the model to focus on movement and interaction dynamics, leading to improved convergence and performance on manipulation videos.

Generalization to different visual encoders. In Table 6, we report detailed statistics regarding experiments of different visual encoders for our feedback mechanism in Figure 7(c), under the default distance threshold setting. Notably, training a 22M parameters ViT-Small model [41] from scratch already yields satisfactory results. VC-1 [56], which is trained on tons of robotics data, provides the policy with a better visual prior and brings the best performance. In general, we hypothesize that the performance bottleneck of the framework still lies in the visual planner, while IDM-based policies are easier to learn and generalize.

D Extended Visualizations

Attention maps of our policy. As discussed in Section 3.3, the state embeddings extracted by the token aggregator distribute reasonably in the latent space and exhibit decent measurement capabilities. We plot attention maps that show the tokens chosen by the token aggregator of the policy model in Figure 14 to provide additional explanation for the observation. As visualized, our model focuses on information essential for action execution, specifically the end-effector and the object it interacts with. One thing worth noting is that it is not explicitly supervised with any form of affordance.

Qualitative analysis. We also provide qualitative examples of the generated videos paired with corresponding tasks in both CALVIN and real-world environments, in Figure 15 and Figure 16, respectively. Our proposed visual planner can reason feasible sub-goals with high temporal coherence.

https://doi.org/10.52202/079017-4411

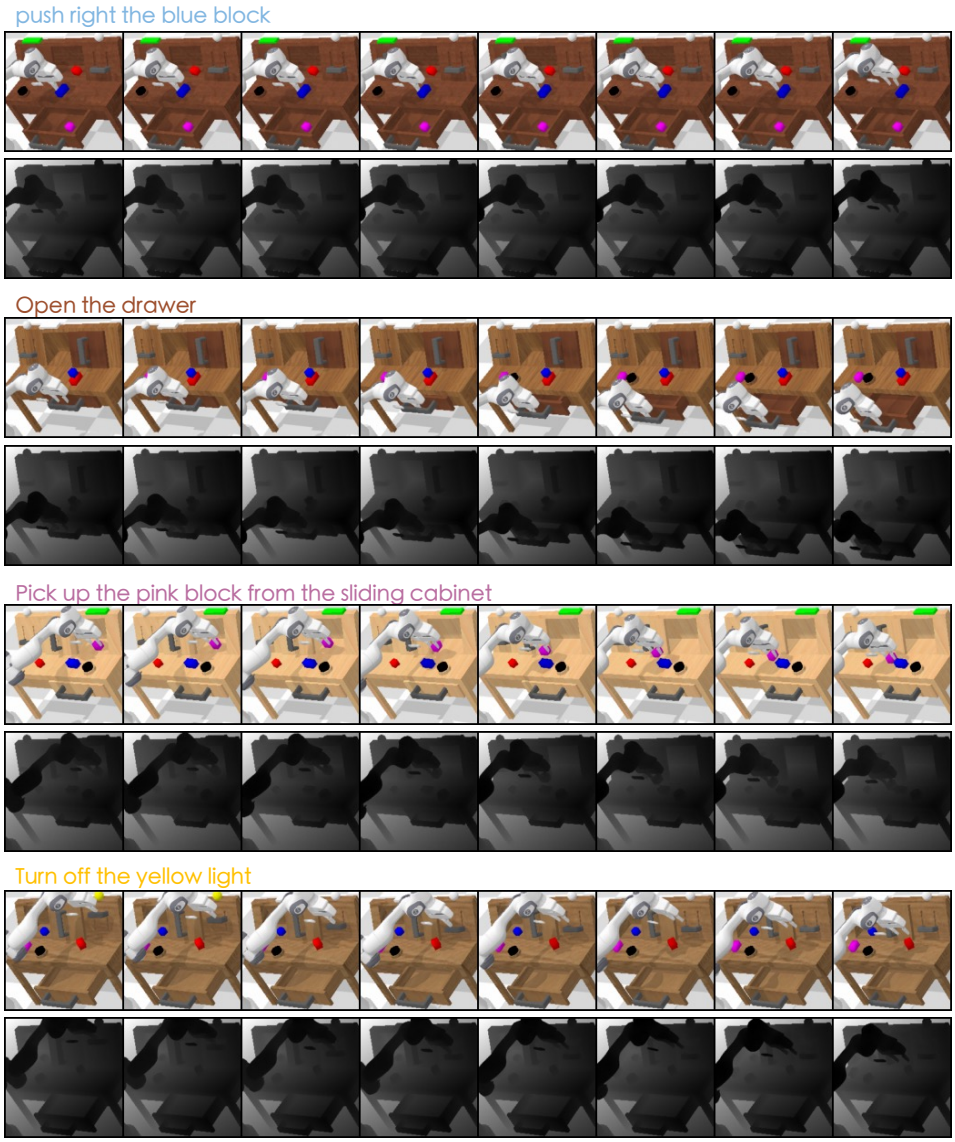


Figure 15: **Visualization on generated RGB-D visual plans.** Our model can generate reliable sub-goals with high consistency between RGB and depth.

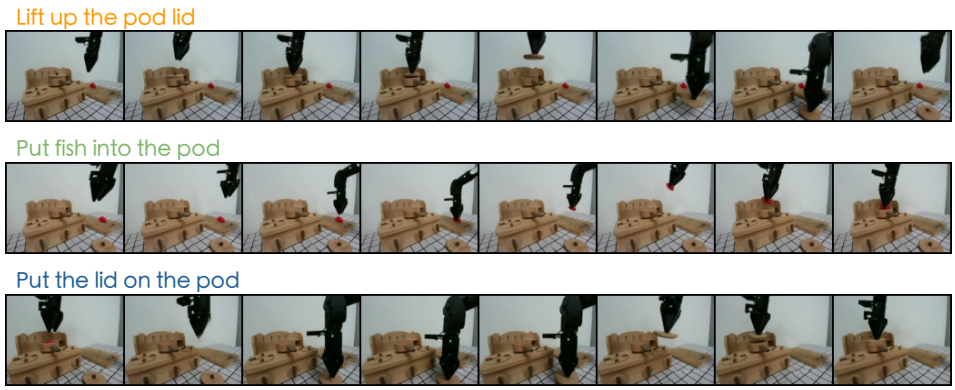


Figure 16: **Visualization on visual plans in real-world environments.**

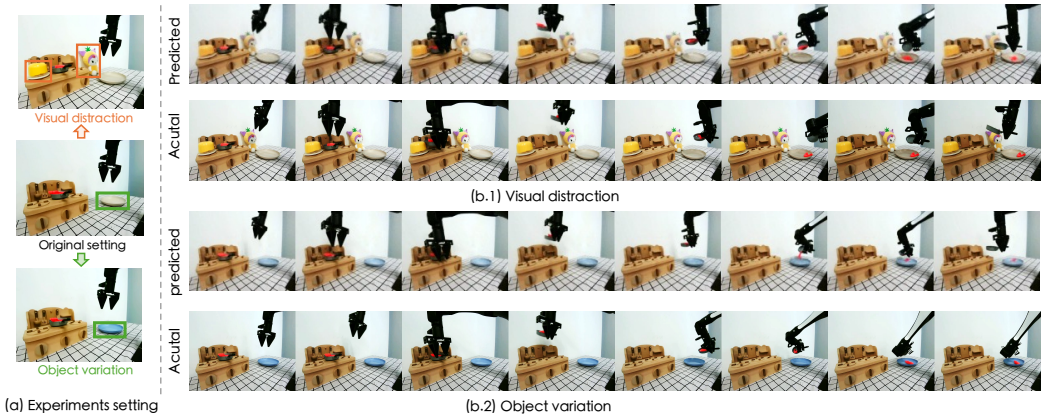


Figure 17: **The predicted visual plan and actual robot execution in real-world experiments.** Our diffusion-based visual planner (up row in each case) can still generate reasonable plans in new scenarios to facilitate successful manipulation (bottom row in each case).

E Broader Impact

CLOVER contributes to the field of robotics by addressing the limitations of open-loop systems and providing a simple yet robust baseline for closed-loop control. This advancement can inspire further research into adaptive control strategies, error modeling, and real-time feedback mechanisms, pushing the boundaries of robot learning and embodied intelligence. All our models are trained on publicly available data that is free of private and sensitive information.

F License of Assets

CALVIN [54] is an open-source simulator which is under the MIT License. The reimplemented methods for performance comparison including ACT [52], R3M [53], and AVDC [31] are all under the MIT License. The pretrained visual encoders adopted, *i.e.*, VC-1 [56], CLIP [23], and DINO [66]), are under the CC BY-NC-SA 3.0 US license, MIT license, and Apache License 2.0, respectively.

Our source code and trained models will be publicly available under Apache License 2.0.

https://doi.org/10.52202/079017-4411

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the claims to reflect the contributions and scope of the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See Section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: This paper does not include theory assumptions or proofs.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide detailed implementation parameters and architectures in Section 4.1, Appendix B.1, and Appendix B.2. We will also open-source the code and models for reproducibility.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

https://doi.org/10.52202/079017-4411

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We will release the code and models.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: See Section 4.1 and Appendix B.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The success rates of our algorithm in simulated environments and real-world robotic experiments are obtained after multiple trials as specified in Section 4.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide experimental training details in Appendix B.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The authors have thoroughly reviewed and conformed the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See discussions in Appendix E.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

https://doi.org/10.52202/079017-4411

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited the corresponding papers throughout our paper and specified their licenses in Appendix F.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We will release code and models under Apache License 2.0 (Appendix F).

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

https://doi.org/10.52202/079017-4411

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.