Bidirectional Decoding:
Improving Action Chunking via Closed-Loop Resampling

Yuejiang Liu,[*] Jubayer Ibn Hamid,[*] Annie Xie, Yoonho Lee, Maximilian Du, Chelsea Finn
Department of Computer Science, Stanford University
https://bid-robot.github.io

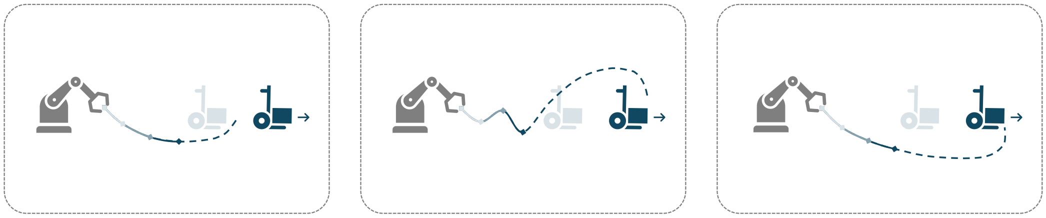
Abstract: Predicting and executing a sequence of actions without intermediate replanning, known as action chunking, is increasingly used in robot learning from human demonstrations. However, its effects on learned policies remain puzzling: some studies highlight its importance for achieving strong performance, while others observe detrimental effects. In this paper, we first dissect the role of action chunking by analyzing the divergence between the learner and the demonstrator. We find that longer action chunks enable a policy to better capture temporal dependencies by taking into account more past states and actions within the chunk. However, this advantage comes at the cost of exacerbating errors in stochastic environments due to fewer observations of recent states. To address this, we propose *Bidirectional Decoding* (BID), a test-time inference algorithm that bridges action chunking with closed-loop operations. BID samples multiple predictions at each time step and searches for the optimal one based on two criteria: (i) backward coherence, which favors samples aligned with previous decisions, (ii) forward contrast, which favors samples close to outputs of a stronger policy and distant from those of a weaker policy. By coupling decisions within and across action chunks, BID enhances temporal consistency over extended sequences while enabling adaptive replanning in stochastic environments. Experimental results show that BID substantially outperforms conventional closed-loop operations of two state-of-the-art generative policies across seven simulation benchmarks and two real-world tasks.

1 Introduction

The increasing availability of human demonstrations has spurred renewed interest in behavioral cloning [1, 2]. In particular, recent studies have highlighted the potential of learning from large-scale demonstrations to acquire a variety of complex skills [3, 4, 5, 6, 7, 8]. However, this approach still struggles with two common properties of human demonstrations: (i) strong temporal dependencies across multiple steps, such as idle pauses [4] and latent strategies [9, 10], (ii) large style variability across different demonstrations, including differences in proficiency [11] and preference [12]. Oftentimes, both properties are prevalent yet unlabeled in collected data, posing significant challenges to traditional behavioral cloning, which typically learns a discriminative model to map an input state to a target action.

In response to these challenges, recent works have pursued a generative approach characterized by two key elements: (i) predicting a sequence of actions over multiple time steps and executing all or part of the sequence, known as *action chunking* [3] or *receding horizon* [4]; (ii) modeling the distribution of action chunks and sampling from the learned model in an independent [4, 13] or weakly dependent [3, 14] manner during deployment. Some studies find these elements crucial for learning a performant policy in controlled laboratory scenarios [3, 4], while other recent work reports opposite outcomes under practical conditions [6]. The reasons behind these conflicting results remain unclear.

[*] Equal contribution.



(a) action chunking

(b) receding horizon

(c) bidirectional decoding

Figure 1: Illustration of action chunking in a stochastic environment, where a robot is challenged to catch a box on a moving trolley. (a) Vanilla action chunking [3] executes actions based on previous predictions, resulting in delayed reactions to the latest box location. (b) Receding horizon [4] allows for faster reactions, but leads to a jittery trajectory in the presence of multimodal demonstrations (*e.g.*, both left- and right-handers). (c) Our Bidirectional Decoding explicitly searches for the optimal action from multiple predictions sampled at each time step, achieving long-range consistency while maintaining closed-loop reactivity.

In this paper, we first dissect the influence of action chunking by examining the divergence between learned policies and human demonstrations. We find that when the input contains no or little history observations – short context lengths [3, 15, 16, 17, 18, 19] – increasing the length of action chunks allows a policy to implicitly condition on more past states, improving its ability to capture the temporal dependencies inherent in demonstrations. However, this advantage comes at the cost of fewer recent state observations, which can be crucial for reacting to unexpected changes in stochastic environments, such as those involving action noise or moving targets. This tradeoff raises a crucial question: How can we preserve the strengths of action chunking without suffering from its limitations in reactivity?

To address this, we introduce *Bidirectional Decoding* (BID), an inference algorithm that integrates action chunking into closed-loop control. Our main idea is to sample multiple predictions at each time step and selectively search for the most desirable action, as illustrated in Fig. 1. Specifically, BID operates on two decoding criteria: (i) backward coherence, which favors samples that are close to the sequence selected at the previous step; (ii) forward contrast, which favors samples that are close to the output of a stronger policy and distant from those of a weaker one. By coupling sequential decision-making both within and across action chunks, BID captures strong temporal dependencies over extended periods while ensuring sufficient reactivity in single-step executions.

The main contributions of this paper are twofold: (a) a thorough analysis of action chunking, and (b) a decoding algorithm to improve it. Our diagnostic simulations in a one-dimensional setting validate our theoretical analysis. Our experiments with two state-of-the-art robotic policies across seven simulations and two real-world tasks show that BID outperforms conventional closed-loop operations of action chunking by more than 26% in relative performance. BID is computationally efficient, model-agnostic, and easy to implement, serving as a plug-and-play component to enhance generative behavior cloning at test time.

2 Related Work

Behavioral Cloning. Learning from human demonstrations is becoming increasingly popular in robot learning due to recent advances in robotic teleoperation interfaces [3, 20, 21, 22]. Generative Behavior cloning, which models the distribution of demonstrations, is particularly appealing due to its algorithmic simplicity and empirical efficacy [3, 17, 22, 23, 24, 25, 26]. However, a significant limitation is compounding errors, where deviations from the training distribution accumulate over time [27, 28]. These errors can be mitigated by gathering expert correction data [27, 29, 30, 31, 32] or injecting noise during data collection [33, 34], but such strategies require additional time and effort from human operators. To address this, recent work proposes predicting a sequence of multiple actions into the future, known as action chunking, which reduces the effective control horizon [3, 35, 36, 37]. By handling sequences of actions, action chunking is also better at handling temporal dependencies in the data, such as idle pauses [4, 38] or multiple styles [11, 12, 39, 40].

However, independently drawn action sequence samples may not preserve the necessary temporal dependencies for smooth and consistent execution. Our work provides a thorough analysis of action chunking and proposes a decoding algorithm to improve it.

Decoding Algorithm. Test-time decoding algorithms have been studied in generative sequence modeling for decades, with renewed attention driven by recent advances in large language modeling (LLM). A prominent approach focuses on leveraging internal metrics, *e.g.*, likelihood scores, to improve the quality of generated sequences. Notable examples include beam search [41, 42], truncated sampling [43, 44, 45], minimum Bayes risk decoding [46, 47], and many others [48, 49]. Another line of research explores the synergy of multiple generative models, such as contrastive decoding [50] and speculative decoding [51], which jointly optimize for quality or efficiency. More recently, several studies have highlighted the potential of guiding the decoding or sampling process through the use of an external discriminative model, such as a classifier [52] or reward model [53]. In the context of robot learning, Huang et al. [54] introduced a framework to guide LLM decoding for long-horizon robotic planning. Similarly, Xu et al. [55] proposed the guidance of diffusion models for manipulator design. To the best of our knowledge, our work is the first to explore decoding algorithms for low-level robotic policy. We propose a decoding strategy centered on forward and backward temporal consistency to address the inherent tradeoffs in action chunking.

3 Analysis: Tradeoffs in Action Chunking

3.1 Preliminaries

Consider a dataset of demonstrations $\mathcal{D} = \{\tau_i\}_{i=1}^N$, where each demonstration τ_i consists of a sequence of state-action pairs $\tau_i = \{(s_1, a_1), (s_2, a_2), \cdots, (s_T, a_T)\}$ provided by a human expert. These demonstrations often exhibit strong temporal dependencies: an action a_t is not only dependent on the current state s_t, but also influenced by the previous k steps of states and actions $(s_{t-1}, a_{t-1}, \cdots, s_{t-k}, a_{t-k})$ due to unobservable latent variables. Some latent variables can globally influence an entire sequence (*e.g.*, speed, left- vs right-handed), while others may locally influence specific segments within a sequence (*e.g.*, planning strategies). They can also vary significantly between different demonstrations or between different segments of the same sequence. Fig. 2 illustrates the decision process of a human expert, highlighting the inherent temporal dependencies.

To model these temporal dependencies, recent behavior cloning methods have focused on learning the joint distribution of future actions conditioned on past states $\pi(a_t, a_{t+1}, \cdots, a_{t+l} | s_{t-c}, \cdots, s_t)$, or more succinctly $\pi(a_{t:t+l} | s_{t-c:t})$. Here, c denotes the number of past steps included as state inputs, and l represents the number of future steps for action outputs. Training such policies typically involves minimizing the divergence between the model distribution and the data distribution,

$$\pi = \arg\min_{\pi} \sum_{\tau \in \mathcal{D}} \sum_{\substack{s_{t-c:t} \\ a_{t:t+l}}} \mathcal{L}(\pi(a_{t:t+l} | s_{t-c:t}), \pi^*(a_{t:t+l} | s_{t-c:t})). \tag{1}$$

Upon training completion, the policy is deployed by sampling a sequence of actions and executing a subset or the entire sequence for $h \in [1, l]$ time steps without re-planning. This approach, commonly referred to as action chunking [3], essentially takes in c states as context and executes h actions. We thus call it a (c, h)-policy.

Interestingly, recent works have shown that the choices of context length c and action horizon h play a crucial role in the empirical success of generative behavior cloning [3, 4, 6]. Specifically, extending the context length c does not always improve performance, especially when human demonstrations are limited (refer to Appendix A.2 for more details). Instead, extending the action horizon h has become a common practice in the design of modern policies.

However, a (c, h)-policy built with a short context length and a long action horizon stands in stark contrast to human experts, which often consider a longer history while re-planning at each time step, effectively following a $(k, 1)$-policy. In the next section, we will analyze why, despite this difference, extending the action horizon can sometimes improve the learned policy in certain scenarios while limiting it in others.

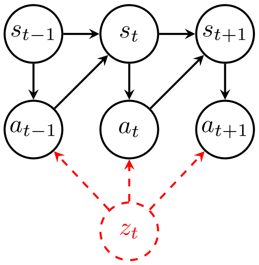


Figure 2: Illustration of the expert decision process, where each action is influenced by multiple past states due to latent variables.

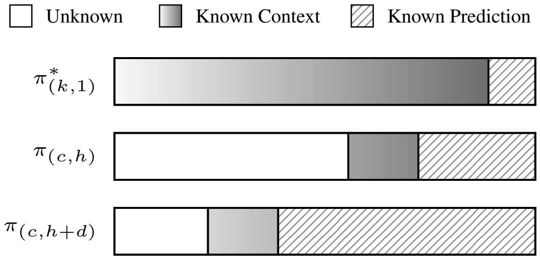


Figure 3: Illustration of $(k, 1)$-expert, (c, h)-learner and $(c, h+d)$-learner. Gray shades are observed *contexts*; darker indicates higher importance. Hatched areas denote executed *predictions*.

3.2 Analysis

To understand the influence of action chunking, we focus on the last time step of an action chunk, where the discrepancy between the expert policy and the learned policy is most pronounced. At this time step t, the expert, which is a $(k, 1)$-policy written as $\pi^* := \pi^*(a_t|s_{t-k:t}, z_{t-k:t})$, predicts a_t by conditioning on k steps of the past states and the corresponding latent variables. In contrast, a learned (c, h)-policy, written as $\pi_{(c,h)} := \pi_{(c,h)}(a_t \mid s_{t-h-c:t-h}, a_{t-h:t-1})$ is constrained to observe c steps of the past states and its predicted actions over the past $h - 1$ steps.

Considering that recent policies often use a short context length c and a moderate action horizon h, we assume the range of temporal dependency modeled by a (c, h)-policy is limited:

Assumption 1. The sum of context length and action horizon is less than the length of temporal dependency in expert demonstrations, $c + h < k$.

Additionally, since a (c, h)-policy observes only a subset of the states that the expert is conditioned on, we assume that an optimal policy must reconstruct all missing information correctly:

Assumption 2. An optimal $\pi_{c,h}$ must infer the unobserved states based on the observed states and actions by modeling the transition dynamics $P(s_{t'} \mid s_{t'-1}, a_{t'-1})$ accurately for all time step t'.

Under these assumptions, the divergence between a learned policy and an expert policy is attributed to two factors: (i) the importance of unobserved states in predicting the current action, and (ii) the difficulty of inferring them based on the available information. To more clearly see the influence of action horizon on these factors, we next compare the performance of two policies that have the same context lengths but different action horizons, $\pi_h := \pi_{(c,h)}(a_t|s_{t-h-c:t-h}, a_{t-h:t-1})$ and $\pi_{h+d} := \pi_{(c,h+d)}(a_t|s_{t-h-d-c:t-h-d}, a_{t-h-d:t-1})$, where $d > 0$ is an extended action horizon.

As illustrated in Fig. 3, each policy has access to unique information that is unavailable to the other. π_h observes some recent states, where π_{h+d} is only aware of the executed actions. On the other hand, π_{h+d} has access to some earlier states and actions, which precede all information available to π_h. We characterize the *importance* of observations as follows (formal definitions in Appendix C.1):

Definition (Expected Observation Advantage). If a policy can observe a state s_t, we say that it has an *observation advantage* α_t over another policy that cannot observe it.

Definition (Maximum Inference Disadvantage). If a policy cannot observe a state s_t, the maximum divergence arising from inferring it incorrectly is ϵ_t.

Hence, we denote the observation advantage that π_h gains from the observed recent states by α_f and the inference disadvantage it incurs from the earlier unobserved states by ϵ_b, whereas π_{h+d} conversely gains α_b but incurs ϵ_f.

The *difficulty* of inferring each unobserved state hinges on both the relevant observations as well as the environmental stochasticity. We quantify this difficulty as follows (formal definitions in Appendix C.1):

Definition (Forward Inference). Let $P_f := P(S_t = g_t|S_{t-1} = g_{t-1}, a_{t-1})$ where g_t and g_{t-1} are the ground truth states in the deterministic environment at time t and $t - 1$, respectively. In deterministic environments, $P_f = 1$, whereas in stochastic settings, P_f is smaller.

Definition (Backward Inference). Let $P_b := P(S_t = g_t | S_{t+1} = g_{t+1})$ where g_t and g_{t+1} are the ground truth states in the deterministic environment at time t and $t+1$, respectively. Since P_b is not conditioned on any action, it has higher entropy in general. In stochastic environments, P_b is small.

Given that the forward inference is generally easier than the backward inference, the performance difference between π_h and π_{h+d} is bounded by the following (proofs are deferred to Appendix C)

Proposition 1 (Consistency-Reactivity Inequalities)**.** *Let $\mathcal{L}$ be a non-linear, convex loss function. Let $\mathcal{S}^+ \subset \{s_{t-k:t}\}$ be the states both the (c, h) and the $(c, h+d)$ policies observe and let $\mathcal{S}^- := \{s_{t-k:t}\} \setminus \mathcal{S}^+$. Let $C := \{a_{t-h-d:t-1}\} \cup \mathcal{S}^+$, $G := \{a_t, z_{t-k:t}\} \cup \mathcal{S}^-$. Then, we can bound the expected loss of the $(c, h+d)$-policy and the (c, h)-policy as:*

$$\alpha_f - \epsilon_b(1 - P_b^{2d}) \leq \min_{\pi_{h+d}} \mathbb{E}_G \left[\mathcal{L}(\pi_{h+d}, \pi^*)|C\right] - \min_{\pi_h} \mathbb{E}_G \left[\mathcal{L}(\pi_h, \pi^*)|C\right] \leq -\alpha_b + \epsilon_f(1 - P_f^{2d}) \tag{2}$$

Remark 1.1. Eq. (2) provides a general comparison of the performance of the two policies. Intuitively, the advantage of each policy stems from the additional information it has access to (*i.e.* α_f for π_h and α_b for π_{h+d}) while the disadvantage is bounded by the maximum divergence arising from inferring missing information incorrectly (*i.e.* $\epsilon_b(1 - P_b^{2d})$ for π_h and $\epsilon_f(1 - P_f^{2d})$ for π_{h+d}).

We next examine two specific environmental settings: highly deterministic and highly stochastic.

In highly deterministic environments, while both policies need to infer the same number of unobserved states, π_{h+d} benefits from conditioning on additional actions, which may significantly aid in inferring the corresponding states through its action chunk. If the maximum errors ϵ_f arising from inferring these states are bounded, π_{h+d} becomes strictly advantageous:

Corollary 2 (Consistency in Deterministic Environments)**.** *In a highly deterministic environment, if a_t is temporally dependent on at least one state in $\{s_{t-h-c-d:t-h-c-1}\}$ and ϵ_f is finite,*

$$\min_{\pi_{h+d}} \mathbb{E}_G \left[\mathcal{L}(\pi_{h+d}, \pi^*)|C\right] < \min_{\pi_h} \mathbb{E}_G \left[\mathcal{L}(\pi_h, \pi^*)|C\right] \tag{3}$$

Conversely, in highly stochastic environments, inferring the unobserved states is challenging, regardless of whether the actions are known. However, the recent states are likely more important than earlier states for predicting the current action a_t. In this case, π_{h+d} becomes strictly more disadvantageous:

Corollary 3 (Reactivity in Stochastic Environments)**.** *In a highly stochastic environment, if temporal dependency decreases over time, i.e., $\alpha_f > \epsilon_b$, then*

$$\min_{\pi_{h+d}} \mathbb{E}_G \left[\mathcal{L}(\pi_{h+d}, \pi^*)|C\right] > \min_{\pi_h} \mathbb{E}_G \left[\mathcal{L}(\pi_h, \pi^*)|C\right] \tag{4}$$

In summary, there is no universally optimal action horizon across all conditions. Determining the appropriate action horizon requires careful consideration of (i) the length of temporal dependencies in the demonstrations and (ii) the level of transition stochasticity present in an environment. When both temporal dependencies and environmental stochasticities are significant, the vanilla action chunking approach leads to an inherent trade-off between these two competing factors.

4 Method: Bidirectional Decoding

As analyzed in §3, action chunking facilitates the modeling of temporal dependencies in demonstrations but sacrifices reactivity to unexpected states in stochastic environments. In this section, we address this issue by bridging long action chunks with closed-loop operations. We will first outline the general framework in §4.1 and then describe two specific criteria in §4.2.

4.1 Test-Time Search

Given a generative policy with context length c and prediction horizon l, an action chunk sampled at time t

$$a \sim \pi_\theta(a_t, a_{t+1}, \cdots, a_{t+l} | s_{t-c}, s_{t-c+1}, \cdots, s_t) \tag{5}$$

is expected to adhere to a consistent latent strategy over the next l time steps. However, naive closed-loop operation of the policy entails executing only the first action of each predicted chunk, leading to a sequence of actions $a_t^{(t)}, a_{t+1}^{(t+1)}, \cdots, a_{t+l}^{(t+l)}$ sampled at different time steps. When multiple latent strategies exist in the demonstrations and are learned by the policy (*e.g.*, left versus right, stop versus go, fast versus slow), independently sampled action chunks may oscillate between different strategies, leading to inconsistent behavior that diverges from the demonstrations.

Our main hypothesis is that while the probability of any pair of samples sharing the same latent strategy is low, the likelihood of finding a consistent pair from a large number of samples is significantly higher. This intuition motivates us to cast the problem of closed-loop action chunking as searching for the optimal action among a batch of plans sampled at each time step,

$$a^* = \arg\min_{a \in \mathcal{A}} \mathcal{L}_B(a) + \mathcal{L}_F(a), \tag{6}$$

where $\mathcal{A}$ is the set of sampled action chunks, $\mathcal{L}_B$ and $\mathcal{L}_F$ are two criteria measuring the temporal dependency with respect to the backward decision and forward plan, which we will describe next.

4.2 Bidirectional Criteria

Backward coherence. To preserve sufficient temporal dependency in closed-loop operations, a sequence of actions should (i) commit to one action chunk in the absence of unexpected changes and (ii) react smoothly to environmental changes. We use the action chunk selected at the previous time step as a reference for enforcing coherence across time. Given the previous action chunk $\hat{a} := a_{t-1}^{(t-1)}, \cdots, a_{t+l-1}^{(t-1)}$, we select action chunks that minimize the weighted sum of Euclidean distances across the $l-1$ overlapping steps:

$$\mathcal{L}_B = \sum_{\tau=0}^{l-1} \rho^\tau \left\| a_{t+\tau}^{(t)} - a_{t+\tau}^{(t-1)} \right\|_2. \tag{7}$$

Here, ρ is a decay hyperparameter to account for growing uncertainty over time. This backward loss encourages similar latent strategies between neighboring steps, while allowing for gradual adaptation to unforeseen transition dynamics.

Forward contrast. An ideal policy should predict far enough into the future to capture the planning capabilities inherent in human demonstrations. However, building such a policy can be challenging in practice due to modeling constraints and dataset limitations. Often, even the best policy available may still produce a significant number of suboptimal plans. To address this, we introduce a forward contrast objective to identify and reject these suboptimal plans. Specifically, we compare each candidate plan with two sets of reference samples: one set from a stronger policy and the other set from a weaker one. We use a well-trained model as the stronger policy, and a model from an early underfitting checkpoint or with a shorter prediction horizon as the weaker policy. Intuitively, the weaker policy cannot capture long-term planning as effectively as the stronger one. Our forward contrast loss is thus framed as minimizing the average distance between a candidate plan and a set of positive samples while maximizing its average distance from the negative ones,

$$\mathcal{L}_F = \frac{1}{N} \left(\sum_{a^+ \in \mathcal{A}^+} \sum_{\tau=0}^{l} \left\| a_{t+\tau}^{(t)} - a_{t+\tau}^{+} \right\|_2 - \sum_{a^- \in \mathcal{A}^-} \sum_{\tau=0}^{l} \left\| a_{t+\tau}^{(t)} - a_{t+\tau}^{-} \right\|_2 \right), \tag{8}$$

where $\mathcal{A}^+ = \mathcal{A} \setminus \{a\}$ is the positive set predicted by the strong policy π, $\mathcal{A}^-$ is the negative set predicted by the weaker one π', and N is the sample size.

Fig. 4 illustrates the combined effects of the backward coherence and forward contrast criteria on sample selection. Notably, not all samples in $\mathcal{A}^+$ and $\mathcal{A}^-$ are necessarily subject to the same mode. To mitigate this, we trim each set by removing samples that deviate significantly from the mode of the previous decision. This is achieved by summing over the K smallest distance values for in the positive and negative sets in Eq. (8). The full process of our decoding method is outlined in Algorithm 1. Since all steps in BID can be computed in parallel, the overall computational overhead remains modest on modern GPU devices.

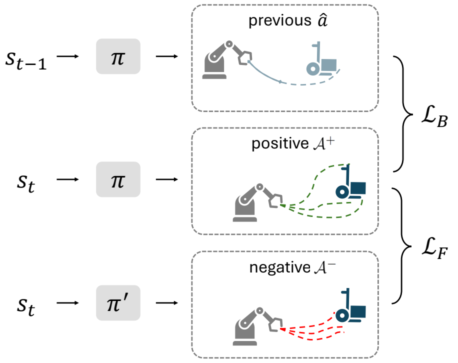


Figure 4: Illustration of bidirectional criteria.

Algorithm 1 Bidirectional Decoding

```
Require: current state $s$, batch size $N$, mode size $K$,
    previous decision $\hat{a}$, strong policy $\pi$, weak policy $\pi'$
Generate $N$ samples from each policy $a \sim \pi(s)$,
    $a' \sim \pi'(s)$ to construct the initial sets $\mathcal{A}$ and $\mathcal{A}'$
Compute the backward loss $\mathcal{L}_B$ for each sample
Select K samples with minimal $\mathcal{L}_B$ from $\mathcal{A}$ and $\mathcal{A}'$
    to construct $\mathcal{A}^+$ and $\mathcal{A}^-$, respectively
Compute the forward loss $\mathcal{L}_F$ for each sample
Select $a^* \in \mathcal{A}$ that minimizes the total loss
Update decision memory $\hat{a} \leftarrow a^*$
```

4.3 Discussions

Interpretation of our method. Our method makes no changes to the learned policy; instead, it intervenes in the model distribution through sample selection. As illustrated in Fig. 11, randomly sampled sequences may be misaligned with both the previous decisions and the target demonstrations. Given a set of candidates, the backward step first identifies the behavioral mode from the past decision stored in memory; the forward step then removes the samples with low likelihood under the target distribution using prior knowledge of positive and negative samples. By comparing samples across time steps and model horizons, our method bridges the gap between the proposal and target distributions during inference.

Relation to recent methods. Our method builds upon the receding horizon [4] and temporal ensembling [3] used in previous works, but with crucial distinctions. Receding horizon seeks a compromise between temporal dependency and dynamic uncertainty by using a moderate action horizon (*e.g.*, half of the prediction horizon), which is inevitably sup-optimal when both factors are prominent. Temporal ensembling strengthens dependency across chunks by averaging multiple decisions over time; however, weighted-averaging operations can be detrimental when consecutive decisions fall into distinct modes. Our method more effectively addresses cross-chunk dependency through dedicated behavioral search and is not mutually exclusive with the previous methods. We will demonstrate in the next section that combining our method with moving average can further improve closed-loop action chunking.

5 Experiments

In this section, we present a series of experiments to answer the following questions:

1. How does our theoretical analysis on action chunking manifest under different conditions?
2. How does the proposed method affect the closed-loop operation of a policy with action chunking?
3. How does the proposed method scale with large batch sizes and complement existing methods?

To this end, we will first validate our theoretical analysis through one-dimensional diagnostic simulations. We will then evaluate BID on seven tasks in three simulation benchmarks, including Push-T [4], RoboMimic [56], and Franka Kitchen [57]. We will subsequently examine the generality and scalability of our method under various base policies and sample sizes. We will finally demonstrate the effectiveness of BID in two challenging real-world tasks involving dynamic objects.

5.1 One-Dimensional Diagnostic Simulations

Setup. We start with a diagnostic experiment in a one-dimensional state space $\{s_0, s_1, \cdots, s_{10}\}$, where s_0 is the starting state and s_{10} is the goal state. The demonstrator plans to move forward by one step in each state, except in s_5 where it pauses unless the last five states visited were s_5. Each forward move has a success probability of $1 - \delta$, where δ denotes the level of stochastic noise in the environment (as described in §3.1). Given these demonstrations, we train a collection of policies

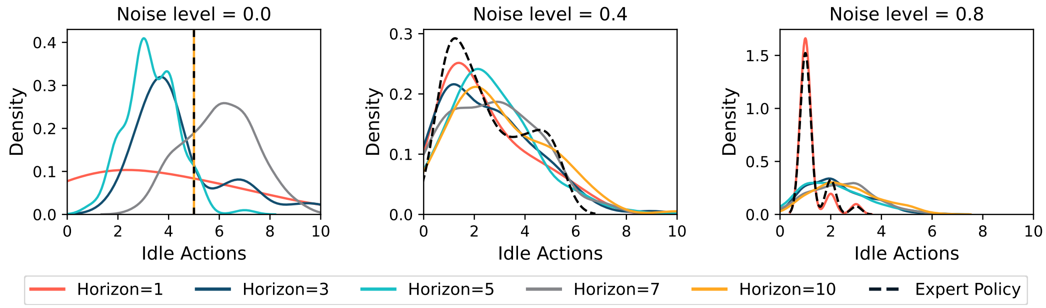


Figure 5: Probability distributions of idle actions taken by learners with varying action horizons in environments with varying stochasticity. The noise level in the environment grows from left to right.

with different action horizons $h \in \{1, 2, 3, 5, 7, 10\}$. We investigate under what action horizon our learner can better imitate the distribution of idle actions taken by the expert over multiple rollouts.

Result. As shown in Fig. 5, when the environment is deterministic ($\delta = 0$), larger action horizons capture the expert distribution better, consistent with Corollary 2. With an action horizon of 10, the learner achieves zero total variation distance with the expert distribution. Conversely, when the environment is highly stochastic $\delta = 0.8$, an action horizon of 1 outperforms all other learners. With moderate noise $\delta = 0.4$, there is no discernible monotonic pattern due to the tradeoff revealed in Proposition 1. Refer to Appendix A for more detailed results.

5.2 Effects of BID on Closed-Loop Action Chunking

Setup. We next examine the effect of our decoding algorithm on closed-loop action chunking in seven simulation tasks. Throughout our experiments, we use Diffusion Policy [4], a state-of-the-art algorithm for generative behavioral cloning, trained on human demonstrations as the base policy. We use the official configurations and checkpoints of the Diffusion Policy in our experiments and consider several competitive inference methods as points of comparison:

- *Vanilla [4]*: Execute the first action of a randomly sampled sequence in a closed-loop manner.
- *Lowvar [13]*: Similar to *Vanilla*, but reduce variance in the initial noise for the diffusion process.
- *Warmstart [14]*: Similar to *Lowvar*, but warm-start the initial noise for the diffusion process from the previous decision.
- *Exponential Moving Average (EMA) [3]*: Smooth action chunking by averaging a new prediction a with the previous one $\hat{a}$ for each overlapping step $a_t = \lambda a_t + (1 - \lambda)\hat{a}_t$. This method is also known as temporal ensembling. Here, $\lambda \in (0, 1)$ is the decay rate of the previous prediction. By default, we set $\lambda = 0.75$.

For BID, we use batch size $N = 30$ and mode size $K = 10$. For each method-environment pair, we report an average score over 100 episodes. Please refer to Appendix B for implementation details.

Result. Our main empirical finding is that while existing inference methods offer some benefits for closed-loop operations, they lack robustness. As shown in Table 1, *Lowvar* and *Warmstart* yield clear improvements over the vanilla closed-loop operation in specific tasks, such as Transport and Franka Kitchen. However, their average performance gains are relatively mild, likely due to the difficulty in controlling the prediction variance caused by stochastic noise at each step of the diffusion process. EMA generally produces better results, yet the improvements vary significantly across different tasks and even degrade the performance in Tool Hang. The challenges of tuning EMA are further discussed in Appendix A. In comparison, BID consistently achieves substantial gains across all tasks, surpassing the vanilla baseline by over 26% in relative improvements.

5.3 Scalability and Compatibility of BID

Setup. We further assess two key properties of BID: scalability with large batch sizes and compatibility with existing inference methods. For scalability, we experiment with batch sizes of $\{1, 5, 15, 30\}$. For compatibility, we apply BID with a batch size of 15 to two competitive baselines, Warmstart and EMA. These experiments are conducted in the Push-T task.

Method	Push T	Lift	Can	Square	Transport	ToolHang	Kitchen	Average
Vanilla [4]	0.78	0.62	0.89	0.74	0.43	0.50	0.22	0.60
Lowvar [13]	0.79	0.54	0.91	0.75	0.52	0.52	0.25	0.61
Warmstart [14]	0.79	0.53	0.92	0.78	0.47	0.51	0.27	0.61
EMA [3]	0.83	0.77	0.92	0.75	0.59	0.46	0.51	0.69
BID (ours)	**0.85**	**0.91**	**0.96**	**0.79**	**0.62**	**0.56**	**0.60**	**0.76**

Table 1: Comparison of different methods for the closed-loop operation of diffusion policies. Evaluations are based on the mean score over 100 episodes in the Push-T, RoboMimic, and 4-Object Franka Kitchen tasks.

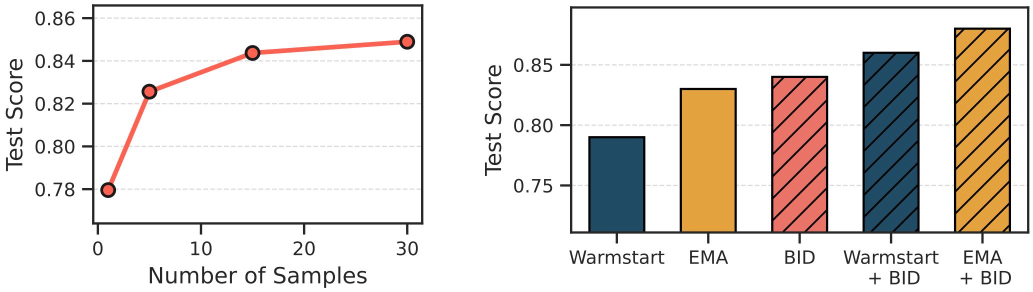


Figure 6: BID benefits from more samples (left) and complements existing inference methods (right). All methods are evaluated for Diffusion Policy on the Pust-T task.

Result. As shown in Fig. 6, our method clearly benefits from the large batch size and is not yet saturated with the default batch size used in our experiment in §5.2. Moreover, the benefits from sampling with BID are complementary to that of existing inference methods. These two properties highlight the strong potential of the proposed method in practice.

5.4 Generality and Overhead of BID

Setup. To examine the generality and overhead of our method, we next extend our experiment to VQ-BET [6], a state-of-the-art transformer-based policy. Specifically, we use the public checkpoint on the Push-T task provided by LeRobot [58] as the base policy. We use a checkpoint early-terminated at 100 epochs as the weak policy in forward contrast. The computational time was measured on a desktop equipped with an NVIDIA A5000 GPU.

Result. Table 2 summarizes the results of the baseline and our method with a batch size of 16 samples. We observe that the vanilla random sampling performs significantly worse than BID in both closed and open-loop operations. Notably, the vanilla open-loop approach exhibits a rapid performance decline as the environment becomes increasingly stochastic. Even in closed-loop operations, the vanilla baseline still experiences a significant performance drop. In comparison, the closed-loop BID demonstrates much higher robustness to stochastic noise.

The experiments on VQ-BET also confirm the absence of a universally optimal action chunk size. Shorter action horizons tend to be more effective in noisy environments, while longer horizons excel in cleaner settings. This variability aligns with our theoretical analysis in §3.1.

Table 3 details the computational overhead associated with BID at varying batch sizes. The result shows that the performance gains of our method come with a 2-3x increase in computational overhead. We expect that this overhead will be less of a constraint with higher-end GPUs.

5.5 Real-world Experiments

Beyond the simulation experiments described above, we further evaluate the proposed BID through two real-world experiments.

5.5.1 Dynamic Placing

Task. We consider a task where the robot is to deliver an object held in its gripper into a cup held by a human. As shown in Fig. 7, this task comprises four main stages and presents two core

Stochastic Noise	0.0	1.0	1.5
Vanilla Open-Loop	61.0	39.0	19.4
BID Open-Loop	**65.2**	39.8	21.4
Vanilla Closed-Loop	52.0	50.4	44.2
BID Closed-Loop	56.6	**54.8**	**54.4**

Table 2: Success rates of VQ-BeT on the Push-T task under various conditions. BID consistently outperforms the vanilla counterpart. Closed-loop BID is particularly advantageous in stochastic settings.

Sample Size	Success (%)	Time (ms)
1 (vanilla)	52.0	12.6
8 (ours)	53.8	25.6
16 (ours)	56.6	26.4
32 (ours)	56.6	27.3

Table 3: Success rates and inference times of VQ-BeT across varying sample sizes. BID benefits from a larger sample size at the cost of a doubled computational overhead, measured on an A5000 GPU.

challenges. First, due to the similar size of the object and the cup, the robot must achieve high precision to place the object accurately into the cup. Second, the position of the cup is not fixed, requiring the robot to adjust its plans based on the latest position continuously. This task mirrors real-world scenarios where robots interact with a dynamic environment, accommodating moving objects and agents.

Demonstration. In light of temporal dependencies and style variations in human behaviors, we intentionally collect a diverse set of demonstration data, differing in factors such as average speed, idling pause, and overall trajectory. We gather a total of 150 demonstration episodes: 50 clean and consistent demonstrations, and 100 noisy and diverse demonstrations. All demonstrations successfully accomplish the task. Additional, the location of the cup is fixed and static within each episode.

Robot. Following previous works [4, 13], we use a Franka Panda as the robot hardware and the vision-based diffusion policy for its operation. The robot is equipped with two cameras: one egocentric camera mounted at the wrist of the robot, one third-person camera mounted at a static bracket. Both cameras provide visual observations at a resolution of 256×256 pixels. The robot operates at a frequency of 10 Hz, with a prediction horizon of 16 time steps.

Evaluation. We evaluate our method in comparison to vanilla random sampling under two conditions: *static target*, where the target cup remains fixed throughout the evaluation, and *dynamic target*, where the target cup is gradually moved. In the dynamic setting, the location of the cup stays within the range of training locations, but the movement is not encountered during training. This evaluation protocol is designed to explicitly assess the ability of the policy to react to unexpected dynamics in the environment. Each method-setting pair is tested over 20 episodes, with both the initial and target locations randomized across different episodes.

Result. We summarize the result of the real-world experiments in Fig. 9. The success rate of vanilla random sampling is generally limited due to oscillations between different latent strategies, which quickly diverge from the distribution of demonstrations. This issue is particularly pronounced in the dynamic setting, where the vanilla baseline struggles to account for the target movements within an action chunk lasting for 1.6 seconds. In contrast, the proposed BID method significantly improves performance in both static and dynamic settings. Notably, BID maintains a similar success rate in the dynamic setting as in the static setting, suggesting its potential to extend action chunking into uncertain environments.

5.5.2 Dynamic Picking

Task. Next, we consider a task where the robot is required to pick up a cup and place it onto a nearby saucer. The cup was pulled with a string until the robot's gripper successfully grasped it. The task consists of five main stages, which are illustrated in Fig. 8. This setup also tests the robot's capability to interact with a dynamic environment, a critical challenge in real-world applications.

Policy and Robot. We utilized the publicly available diffusion policy checkpoint from UMI [22] without any additional fine-tuning. Notably, the policy was originally trained using demonstrations in a static setting, where the cup's position remained constant throughout the task. Our experimental setup mirrored the one described by UMI, using the same UR5 robot hardware. This allowed us to directly evaluate the policy's transferability to a dynamic environment, where the cup's position changes during the task.

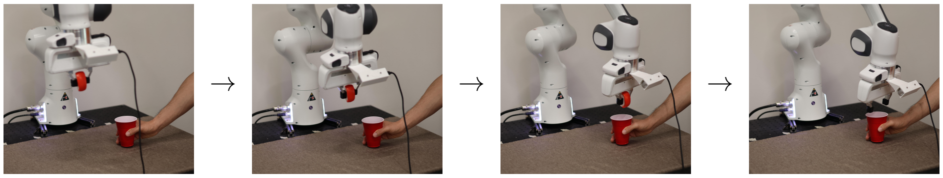

(a) initialize (b) approach (c) decelerate (d) place

Figure 7: Human demonstrations on a Franka Panda robot for a real-world object delivery task. The robot is tasked with delivering an object held in its gripper into a cup held by a human. Each demonstration consists of four main stages: (a) initialize the robot position randomly, (b) approach the target cup, (c) slow down near the target cup, and (d) release the object. The position of the target cup may change during an episode.

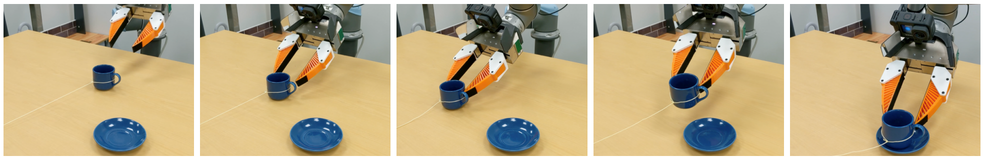

(a) initialize (b) approach (c) grasp (d) pick (e) place

Figure 8: The robot is tasked with picking up a cup and placing it on a saucer nearby. The four main stages are (a) initializing the robot, (b) approaching the target cup, (c) grasping the target cup, (d) picking up the cup, and (e) placing the cup on the target saucer. The position of the target cup may change during an episode.

Evaluation. We evaluated BID against three baselines: vanilla random sampling in both open-loop and closed-loop configurations, and EMA (closed-loop). These methods were tested under two conditions: *static target*, where the cup remained in a fixed position, and *dynamic target*, where the cup was moved using the string. Each method-setting combination was tested across 20 episodes, with the initial positions of the cup and saucer kept consistent to ensure controlled comparisons.

Results. The results, summarized in Fig. 10, highlight the challenges of the dynamic setting. Open-loop vanilla sampling performed poorly due to its inability to adapt to the cup's movements, often failing to approach the cup as it was pulled. While closed-loop vanilla sampling showed improved reactivity, it suffered from inconsistent trajectories, resulting in jittery behavior when attempting to grasp and place the cup. Similarly, closed-loop EMA sampling demonstrated higher adaptability to environmental changes but often failed to firmly grasp the cup, likely due to the limitations of naive averaging, which compromises commitment to a specific strategy. In contrast, BID achieved at least a 2x improvement in success rate compared to all other methods in the dynamic setting, while maintaining its performance in the static setting, demonstrating both adaptability and precision in dynamic environments.

Other experiments. Please refer to Appendix A for additional analyses and ablations.

6 Conclusion

Summary. We have analyzed the strengths and limitations of action chunking for robot learning from human demonstrations. Based on our analysis, we proposed Bidirectional Decoding (BID), an inference algorithm that takes into account both past decisions and future plans for sample selection. Our experimental results show that BID can consistently improve closed-loop operations, scale well with computational resources, and complement existing methods. We hope these findings provide a new perspective on addressing the challenges of generative behavioral cloning at test time.

Limitations. One major limitation of BID lies in its computational complexity. While the decoding process can be parallelized on modern GPUs, it may remain prohibitive for high-frequency operations on low-cost robots. Designing algorithms that can generate quality yet diverse action chunks under batch size constraints can be an interesting avenue for future research. Additionally, our analysis and method have been limited to policies with short context lengths, driven by their empirical

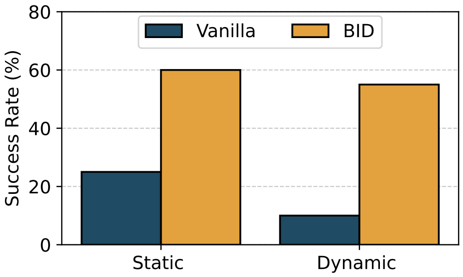


Figure 9: Success rate of object delivery. Each method-setting is evaluated across 20 episodes. BID achieves much higher success rate than the vanilla baseline, effectively handling the diverse demonstrations and dynamic target.

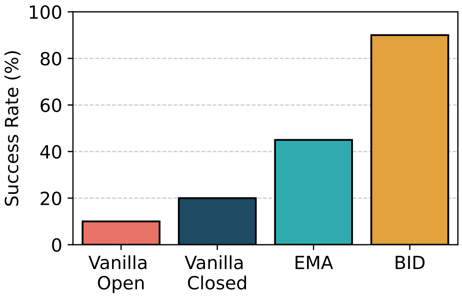


Figure 10: Success rate of cup replacement in the dynamic setting. Each method is evaluated across 20 episodes. Existing methods degrade substantially under slow cup movements, whereas BID retains a strong performance.

effectiveness with limited human demonstrations. Developing techniques capable of learning robust long-context policies can be another compelling direction for future research.

Acknowledgments

We would like to thank Anikait Singh, Eric Mitchell, Kyle Hsu, Moo Jin Kim, Annie Chen, Sasha Khazatsky, Rhea Malhotra and other members of the IRIS lab for valuable comments on this project. We would like to thank Yifan Hou, Huy Ha, Chuer Pan, Austin Patel, and the REAL lab for helpful support in setting up real-world experiments. We would like to thank Asher Spector and Danny Tse for their insightful feedback on the theoretical analysis. This work is supported by the AI Institute, ONR grant N00014-21-1-2685, an NSF CAREER award, and an SNSF Postdoc fellowship.

References

[1] C. G. Atkeson and S. Schaal. Robot Learning From Demonstration. In *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, pages 12–20, July 1997.

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

[3] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. In *Robotics: Science and Systems (RSS) 2023*, Apr. 2023.

[4] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. In *Robotics: Science and Systems XIX*. Robotics: Science and Systems Foundation, July 2023. ISBN 978-0-9923747-9-2.

[5] Z. Fu, T. Z. Zhao, and C. Finn. Mobile ALOHA: Learning Bimanual Mobile Manipulation with Low-Cost Whole-Body Teleoperation, Jan. 2024.

[6] S. Lee, Y. Wang, H. Etukuru, H. J. Kim, N. M. M. Shafiullah, and L. Pinto. Behavior Generation with Latent Actions, Mar. 2024.

[7] O. X.-E. Collaboration, A. Padalkar, A. Pooley, A. Mandlekar, A. Jain, A. Tung, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Singh, A. Garg, A. Brohan, A. Raffin, A. Wahid, B. Burgess-Limerick, B. Kim, B. Schölkopf, B. Ichter, C. Lu, C. Xu, C. Finn, C. Xu, C. Chi, C. Huang, C. Chan, C. Pan, C. Fu, C. Devin, D. Driess, D. Pathak, D. Shah, D. Büchler, D. Kalashnikov, D. Sadigh, E. Johns, F. Ceola, F. Xia, F. Stulp, G. Zhou, G. S. Sukhatme, G. Salhotra, G. Yan, G. Schiavi, G. Kahn, H. Su, H.-S. Fang, H. Shi, H. B. Amor, H. I. Christensen, H. Furuta, H. Walke, H. Fang, I. Mordatch, I. Radosavovic, I. Leal, J. Liang,

J. Abou-Chakra, J. Kim, J. Peters, J. Schneider, J. Hsu, J. Bohg, J. Bingham, J. Wu, J. Wu, J. Luo, J. Gu, J. Tan, J. Oh, J. Malik, J. Booher, J. Tompson, J. Yang, J. J. Lim, J. Silvério, J. Han, K. Rao, K. Pertsch, K. Hausman, K. Go, K. Gopalakrishnan, K. Goldberg, K. Byrne, K. Oslund, K. Kawaharazuka, K. Zhang, K. Rana, K. Srinivasan, L. Y. Chen, L. Pinto, L. Fei-Fei, L. Tan, L. Ott, L. Lee, M. Tomizuka, M. Spero, M. Du, M. Ahn, M. Zhang, M. Ding, M. K. Srirama, M. Sharma, M. J. Kim, N. Kanazawa, N. Hansen, N. Heess, N. J. Joshi, N. Suenderhauf, N. Di Palo, N. M. M. Shafiullah, O. Mees, O. Kroemer, P. R. Sanketi, P. Wohlhart, P. Xu, P. Sermanet, P. Sundaresan, Q. Vuong, R. Rafailov, R. Tian, R. Doshi, R. Martín-Martín, R. Mendonca, R. Shah, R. Hoque, R. Julian, S. Bustamante, S. Kirmani, S. Levine, S. Moore, S. Bahl, S. Dass, S. Sonawani, S. Song, S. Xu, S. Haldar, S. Adebola, S. Guist, S. Nasiriany, S. Schaal, S. Welker, S. Tian, S. Dasari, S. Belkhale, T. Osa, T. Harada, T. Matsushima, T. Xiao, T. Yu, T. Ding, T. Davchev, T. Z. Zhao, T. Armstrong, T. Darrell, V. Jain, V. Vanhoucke, W. Zhan, W. Zhou, W. Burgard, X. Chen, X. Wang, X. Zhu, X. Li, Y. Lu, Y. Chebotar, Y. Zhou, Y. Zhu, Y. Xu, Y. Wang, Y. Bisk, Y. Cho, Y. Lee, Y. Cui, Y.-H. Wu, Y. Tang, Y. Zhu, Y. Li, Y. Iwasawa, Y. Matsuo, Z. Xu, and Z. J. Cui. Open X-Embodiment: Robotic Learning Datasets and RT-X Models, Dec. 2023.

[8] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, P. D. Fagan, J. Hejna, M. Itkina, M. Lepert, Y. J. Ma, P. T. Miller, J. Wu, S. Belkhale, S. Dass, H. Ha, A. Jain, A. Lee, Y. Lee, M. Memmel, S. Park, I. Radosavovic, K. Wang, A. Zhan, K. Black, C. Chi, K. B. Hatch, S. Lin, J. Lu, J. Mercat, A. Rehman, P. R. Sanketi, A. Sharma, C. Simpson, Q. Vuong, H. R. Walke, B. Wulfe, T. Xiao, J. H. Yang, A. Yavary, T. Z. Zhao, C. Agia, R. Baijal, M. G. Castro, D. Chen, Q. Chen, T. Chung, J. Drake, E. P. Foster, J. Gao, D. A. Herrera, M. Heo, K. Hsu, J. Hu, D. Jackson, C. Le, Y. Li, K. Lin, R. Lin, Z. Ma, A. Maddukuri, S. Mirchandani, D. Morton, T. Nguyen, A. O'Neill, R. Scalise, D. Seale, V. Son, S. Tian, E. Tran, A. E. Wang, Y. Wu, A. Xie, J. Yang, P. Yin, Y. Zhang, O. Bastani, G. Berseth, J. Bohg, K. Goldberg, A. Gupta, A. Gupta, D. Jayaraman, J. J. Lim, J. Malik, R. Martín-Martín, S. Ramamoorthy, D. Sadigh, S. Song, J. Wu, M. C. Yip, Y. Zhu, T. Kollar, S. Levine, and C. Finn. DROID: A Large-Scale In-The-Wild Robot Manipulation Dataset, Mar. 2024.

[9] A. Xie, D. Losey, R. Tolsma, C. Finn, and D. Sadigh. Learning Latent Representations to Influence Multi-Agent Interaction. In *Proceedings of the 2020 Conference on Robot Learning*, pages 575–588. PMLR, Oct. 2021.

[10] X. Ma, S. Patidar, I. Haughton, and S. James. Hierarchical Diffusion Policy for Kinematics-Aware Multi-Task Robotic Manipulation, Mar. 2024.

[11] S. Belkhale, Y. Cui, and D. Sadigh. Data quality in imitation learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[12] A. Kuefler and M. J. Kochenderfer. Burn-in demonstrations for multi-modal imitation learning. *arXiv preprint arXiv:1710.05090*, 2017.

[13] A. Prasad, K. Lin, J. Wu, L. Zhou, and J. Bohg. Consistency Policy: Accelerated Visuomotor Policies via Consistency Distillation, May 2024.

[14] M. Janner, Y. Du, J. B. Tenenbaum, and S. Levine. Planning with Diffusion for Flexible Behavior Synthesis. In *International Conference on Machine Learning (ICML)*, May 2022.

[15] A. Mandlekar, F. Ramos, B. Boots, S. Savarese, L. Fei-Fei, A. Garg, and D. Fox. IRIS: Implicit Reinforcement without Interaction at Scale for Learning Control from Offline Robot Manipulation Data. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4414–4420, May 2020.

[16] H. Bharadhwaj, J. Vakil, M. Sharma, A. Gupta, S. Tulsiani, and V. Kumar. RoboAgent: Generalization and Efficiency in Robot Manipulation via Semantic Augmentations and Action Chunking, Sept. 2023.

[17] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, P. Florence, C. Fu, M. G. Arenas, K. Gopalakrishnan, K. Han, K. Hausman, A. Herzog, J. Hsu, B. Ichter, A. Irpan, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, L. Lee, T.-W. E. Lee, S. Levine, Y. Lu, H. Michalewski, I. Mordatch, K. Pertsch, K. Rao, K. Reymann, M. Ryoo, G. Salazar, P. Sanketi, P. Sermanet, J. Singh, A. Singh, R. Soricut, H. Tran, V. Vanhoucke, Q. Vuong, A. Wahid, S. Welker, P. Wohlhart, J. Wu, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control, July 2023.

[18] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, K.-H. Lee, S. Levine, Y. Lu, U. Malla, D. Manjunath, I. Mordatch, O. Nachum, C. Parada, J. Peralta, E. Perez, K. Pertsch, J. Quiambao, K. Rao, M. Ryoo, G. Salazar, P. Sanketi, K. Sayed, J. Singh, S. Sontakke, A. Stone, C. Tan, H. Tran, V. Vanhoucke, S. Vega, Q. Vuong, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich. RT-1: Robotics Transformer for Real-World Control at Scale. In *Robotics: Science and Systems XIX*. Robotics: Science and Systems Foundation, July 2023. ISBN 978-0-9923747-9-2.

[19] L. X. Shi, A. Sharma, T. Z. Zhao, and C. Finn. Waypoint-Based Imitation Learning for Robotic Manipulation. In *Proceedings of The 7th Conference on Robot Learning*, pages 2195–2209. PMLR, Dec. 2023.

[20] A. Sivakumar, K. Shaw, and D. Pathak. Robotic telekinesis: Learning a robotic hand imitator by watching humans on youtube. *Robotics: Science and Systems (RSS)*, 2022.

[21] P. Wu, Y. Shentu, Z. Yi, X. Lin, and P. Abbeel. Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators. *arXiv preprint arXiv:2309.13037*, 2023.

[22] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. *arXiv preprint arXiv:2402.10329*, 2024.

[23] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022.

[24] P. Florence, C. Lynch, A. Zeng, O. A. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson. Implicit behavioral cloning. In *Conference on Robot Learning*, pages 158–168. PMLR, 2022.

[25] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[26] N. M. Shafiullah, Z. Cui, A. A. Altanzaya, and L. Pinto. Behavior transformers: Cloning k modes with one stone. *Advances in neural information processing systems*, 35:22955–22968, 2022.

[27] S. Ross, G. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[28] L. Ke, J. Wang, T. Bhattacharjee, B. Boots, and S. Srinivasa. Grasping with chopsticks: Combating covariate shift in model-free imitation learning for fine manipulation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6185–6191. IEEE, 2021.

[29] M. Kelly, C. Sidrane, K. Driggs-Campbell, and M. J. Kochenderfer. Hg-dagger: Interactive imitation learning with human experts. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8077–8083. IEEE, 2019.

[30] K. Menda, K. Driggs-Campbell, and M. J. Kochenderfer. Ensembledagger: A bayesian approach to safe imitation learning. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5041–5048. IEEE, 2019.

[31] R. Hoque, A. Balakrishna, E. Novoseller, A. Wilcox, D. S. Brown, and K. Goldberg. Thriftydagger: Budget-aware novelty and risk gating for interactive imitation learning. *arXiv preprint arXiv:2109.08273*, 2021.

[32] R. Hoque, A. Balakrishna, C. Putterman, M. Luo, D. S. Brown, D. Seita, B. Thananjeyan, E. Novoseller, and K. Goldberg. Lazydagger: Reducing context switching in interactive imitation learning. In *2021 IEEE 17th international conference on automation science and engineering (case)*, pages 502–509. IEEE, 2021.

[33] M. Laskey, J. Lee, R. Fox, A. Dragan, and K. Goldberg. Dart: Noise injection for robust imitation learning. In *Conference on robot learning*, pages 143–156. PMLR, 2017.

[34] D. Brandfonbrener, S. Tu, A. Singh, S. Welker, C. Boodoo, N. Matni, and J. Varley. Visual backtracking teleoperation: A data collection protocol for offline image-based reinforcement learning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11336–11342. IEEE, 2023.

[35] L. Lai, A. Z. Huang, and S. J. Gershman. Action chunking as policy compression. 2022.

[36] A. George and A. B. Farimani. One act play: Single demonstration behavior cloning with action chunking transformers. *arXiv preprint arXiv:2309.10175*, 2023.

[37] H. Bharadhwaj, J. Vakil, M. Sharma, A. Gupta, S. Tulsiani, and V. Kumar. Roboagent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking. *2024 IEEE International Conference on Robotics and Automation*, 2023.

[38] G. Swamy, S. Choudhury, D. Bagnell, and S. Wu. Causal imitation learning under temporally correlated noise. In *International Conference on Machine Learning*, pages 20877–20890. PMLR, 2022.

[39] Y. Li, J. Song, and S. Ermon. Infogail: Interpretable imitation learning from visual demonstrations. *Advances in neural information processing systems*, 30, 2017.

[40] K. Gandhi, S. Karamcheti, M. Liao, and D. Sadigh. Eliciting compatible demonstrations for multi-human imitation learning. In *Conference on Robot Learning*, pages 1981–1991. PMLR, 2023.

[41] M. Freitag and Y. Al-Onaizan. Beam search strategies for neural machine translation. *First Workshop on Neural Machine Translation*, 2017.

[42] A. Vijayakumar, M. Cogswell, R. Selvaraju, Q. Sun, S. Lee, D. Crandall, and D. Batra. Diverse beam search for improved description of complex scenes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[43] A. Fan, M. Lewis, and Y. Dauphin. Hierarchical neural story generation. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018.

[44] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi. The curious case of neural text degeneration. *International Conference on Learning Representations*, 2020.

[45] J. Hewitt, C. Manning, and P. Liang. Truncation Sampling as Language Model Desmoothing. In Y. Goldberg, Z. Kozareva, and Y. Zhang, editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3414–3427, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics.

[46] S. Kumar and W. Byrne. Minimum Bayes-Risk Decoding for Statistical Machine Translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 169–176, Boston, Massachusetts, USA, May 2004. Association for Computational Linguistics.

[47] M. Müller and R. Sennrich. Understanding the Properties of Minimum Bayes Risk Decoding in Neural Machine Translation. In C. Zong, F. Xia, W. Li, and R. Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 259–272, Online, Aug. 2021. Association for Computational Linguistics.

[48] S. Welleck, I. Kulikov, S. Roller, E. Dinan, K. Cho, and J. Weston. Neural text generation with unlikelihood training. *International Conference on Learning Representations*, 2019.

[49] S. Basu, G. S. Ramachandran, N. S. Keskar, and L. R. Varshney. Mirostat: A neural text decoding algorithm that directly controls perplexity. *International Conference on Learning Representations*, 2021.

[50] X. L. Li, A. Holtzman, D. Fried, P. Liang, J. Eisner, T. Hashimoto, L. Zettlemoyer, and M. Lewis. Contrastive Decoding: Open-ended Text Generation as Optimization. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12286–12312, Toronto, Canada, July 2023. Association for Computational Linguistics.

[51] Y. Leviathan, M. Kalman, and Y. Matias. Fast Inference from Transformers via Speculative Decoding. In *Proceedings of the 40th International Conference on Machine Learning*, pages 19274–19286. PMLR, July 2023.

[52] P. Dhariwal and A. Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[53] M. Khanov, J. Burapacheep, and Y. Li. ARGS: Alignment as Reward-Guided Search. In *The Twelfth International Conference on Learning Representations*, Oct. 2023.

[54] W. Huang, F. Xia, D. Shah, D. Driess, A. Zeng, Y. Lu, P. Florence, I. Mordatch, S. Levine, K. Hausman, and B. Ichter. Grounded Decoding: Guiding Text Generation with Grounded Models for Embodied Agents. In *Thirty-Seventh Conference on Neural Information Processing Systems*, Nov. 2023.

[55] X. Xu, H. Ha, and S. Song. Dynamics-Guided Diffusion Model for Robot Manipulator Design, Feb. 2024.

[56] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín. What Matters in Learning from Offline Human Demonstrations for Robot Manipulation. In *Proceedings of the 5th Conference on Robot Learning*, pages 1678–1690. PMLR, Jan. 2022.

[57] A. Gupta, V. Kumar, C. Lynch, S. Levine, and K. Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. In *Conference on Robot Learning*, pages 1025–1037. PMLR, 2020.

[58] R. Cadene, S. Alibert, A. Soare, Q. Gallouedec, A. Zouitine, and T. Wolf. Lerobot: State-of-the-art machine learning for real-world robotics in pytorch. https://github.com/huggingface/lerobot, 2024.

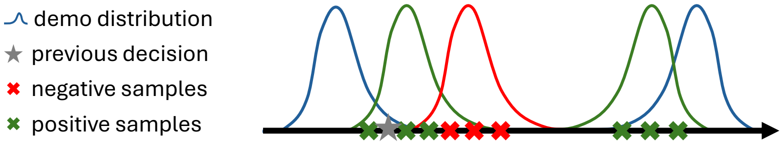


Figure 11: Distributional interpretation of BID. The backward criterion (Equation 7) favors samples close to the past decision; the forward criterion (Equation 8) promotes samples with a high likelihood under the target distribution.

	Action Horizon				
Noise Level	1	3	5	7	10
0.0	4.21	1.75	1.55	1.28	**0.00**
0.4	0.55	**0.30**	0.95	0.53	0.93
0.8	**0.04**	0.98	1.23	1.26	1.44

Table 4: Total variation distance between the action distributions of each model and the expert in environments with varying noise levels. Lower values indicate better performance.

A Additional Experiments

A.1 One-dimensional Simulations

In addition to Fig. 5, we summarize the total variation distance between each learned policy and the demonstration in the one-dimensional simulation. Our results indicate that a shorter action horizon is more effective in noisier environments, whereas a longer action horizon yields better performance in static environments.

A.2 Other Horizon Choices

Setup. Our work builds on the premise that the prediction horizon is longer than the context length, as commonly designed for recent policies. While BID mitigates the inherent limitations of this design choice through test-time decoding, an important question remains: could extending the history context itself yield stronger policies? To understand this, we trained diffusion policies with varying combinations of prediction horizons and context lengths on the Push-T task. Specifically, we use a short context length ($c = 2$) and a short prediction horizon ($h = 2$) as our baseline, and incrementally increase these parameters to larger values $6, 10, 14$ to assess their impact.

Result. Fig. 12 compares the performance of the policy learned with different $\Delta h = h - c$. As expected, the policy with both a short prediction horizon and a short context length struggles to capture long-range temporal dependencies, leading to suboptimal performance. Interestingly, extending the context length initially boosts performance ($\Delta h = -4$), but this trend reverses as the context length becomes too long ($\Delta h \leq -8$), likely due to overfitting to an increased number of spurious features. In contrast, expanding the prediction horizon results in more robust performance improvements, validating its pivotal role in policy design given limited demonstrations.

A.3 Ablation Study of Forward Contrast

Setup. To understand the effect of forward contrast (Equation 8), we evaluate the full version of our method against three reduced variants: without forward contrast, without positive samples (negative samples only), and without negative samples (positive samples only). Similar to §5.3, our ablation study is conducted in the representative Push-T task.

Result. Fig. 13 summarizes the result of this ablation study. Notably, both positive and negative samples are essential for effective sample selection, and omitting either leads to significant performance declines. We conjecture that, without negative samples, our decoding method reduces to an approximate maximum a posteriori estimation, which can result in suboptimal decisions due to

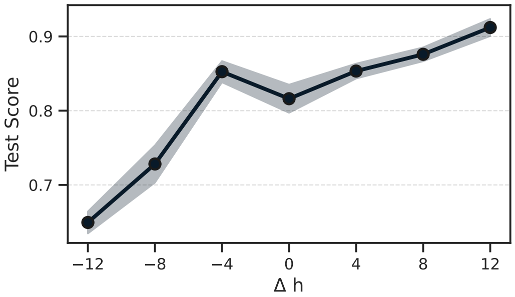
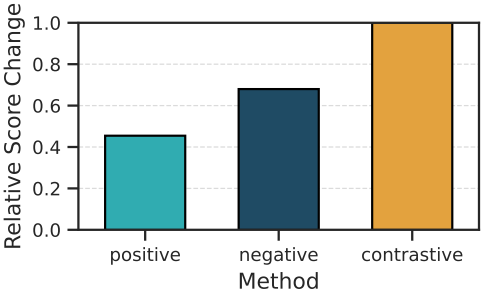


Figure 12: Effect of prediction horizon (h) and context length (c) on diffusion policies in the Push-T task. The baseline is set at $h = 2$ and $c = 2$, with $\Delta h = h - c = 0$. Extending the prediction horizon ($h > 2$) consistently improves performance, whereas extending the context length ($c > 2$) can cause substantial performance declines. Each model is trained for $5k$ epochs. Results are averaged over the last five checkpoints.



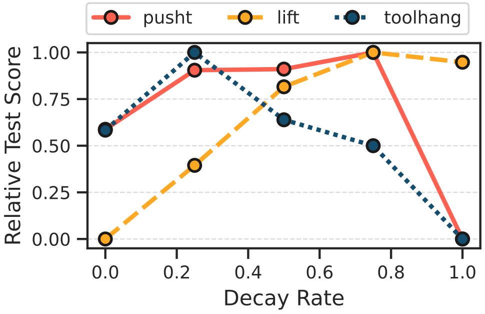


Figure 13: Effect of positive and negative samples on forward contrast. We measure the improvements over the vanilla baseline in the Push-T task, relative to the full version of BID. Using only positive or only negative samples does not achieve the full performance gains seen with the full contrastive.

Figure 14: Effect of the decay rate for the exponential moving average. In each task, we measure the relative performance among different decay rates. The optimal decay rate varies by task, leading to a practical challenge of identifying a universal temporal ensembling strategy [3].

modeling errors. Conversely, without positive samples, the selected samples may be biased towards rare instances. This result highlights the importance of both components and suggests the potential for extending this paradigm in future work.

A.4 Challenges for Temporal Ensebmling

EMA exhibits competitive performance in Table 1. However, tuning its decay rate can be difficult in practice. Fig. 14 shows the sensitivity of EMA to the decay rate across three different tasks, where the optimal choices differ significantly. We conjecture that this high sensitivity stems from the variability in the latent strategies between consecutive predictions. When consecutive predictions follow similar strategies, a lower decay rate (*i.e.*, stronger moving average) can enhance smoothness and improve performance. Conversely, when consecutive predictions diverge in their underlying strategies, averaging them can introduce adverse effects. Our method promotes coherence in latent strategies and thus effectively complements temporal ensembling, as evidenced in Fig. 6.

B Additional Details

Simulation Details. Our simulation experiments are conducted on three robot manipulation benchmarks. We use the training data collected from human demonstrations in each benchmark.

Push-T: We adopt the Push-T environment introduced in [4], where the goal is to push a T-shaped block on a table to a target position. The action space is two-dimensional end-effector velocity control. The training dataset contains 206 demonstrations collected by humans.

name	value
batch size N	30
mode size K	10
prediction horizon l	16
temporal coherence decay ρ	0.9
moving average decay λ	0.75

Table 5: Default hyper-parameters in our experiments.

Robomimic: We use five tasks in the Robomimic suite [56], namely Lift, Can, Square, Transport, and Tool Hang. The training dataset for each task contains 300 episodes collected from multi-human (MH) demonstrations.

Franka Kitchen: We use the Franka Kitchen environment from [57], featuring a Franka Panda arm with a seven-dimensional action space and 566 human-collected demonstrations. The learned policy is evaluated on test cases involving four or more objects (p4), a challenging yet practical task for robotic manipulation in household contexts.

Implementation Details. Our implementations are built upon the official code of Diffusion Policy [4], with modifications made solely to the inference process. Our simulation experiments use state inputs, while our real-world experiments utilize visual inputs. For each simulation task, we use the best checkpoint available from `https://diffusion-policy.cs.columbia.edu/data/experiments` and evaluate it in the closed-loop operation, *i.e.*, action horizon is set to 1. For forward contrast, we train the weak policy for 50-100 epochs, resulting in a diverse set of suboptimal plans. The core hyperparameters are summarized in Table 5.

For VQ-BeT [6], we use the best public checkpoint from the LeRobot Repository [58] as the strong policy whereas we used a checkpoint trained for 100000 iterations as the weak policy. Backward coherence, forward contrast and BID require sample diversity, so we select temperature=0.5 for these methods whereas for the vanilla sampling, we used the default value temperature=0.1. Our code will be released publicly.

C Proofs

First, we establish the following lemma which will help us compare different function classes:

Lemma 4. *Let $\mathcal{L}$ be a convex function and let X and Y be two random variables. Let G be the class of functions $g(X)$ that accept X as an input. Then*

$$\min_{g(X)\in G} \mathbb{E}_{X,Y}\left[\mathcal{L}(f(X,Y),g(X))\right] = \mathbb{E}_X\left[\min_{c\in\mathbb{R}} \mathbb{E}_Y\left[\mathcal{L}(f(X,Y),c)|X\right]\right].$$

Proof. The left hand side is less than or equal to the right hand side by the following logic:

$$\begin{aligned}\mathbb{E}_X\left[\min_{c\in\mathbb{R}} \mathbb{E}_Y\left[\mathcal{L}(f(X,Y),c)|X\right]\right] &= \mathbb{E}_X\left[\mathbb{E}_Y\left[\mathcal{L}(f(X,Y),c^*(X))|X\right]\right] \\ &\geq \min_{g(X)\in G} \mathbb{E}_{X,Y}\left[\mathcal{L}(f(X,Y),g(X))\right]\end{aligned}$$

where we used $c^*(X) := \arg\min_c \mathbb{E}_X[\mathcal{L}(f(X,Y),c)|X]$. We get the inequality by recognizing that $\mathbb{R} \subsetneq G$. On the other hand, the left hand side is greater than or equal to the right hand side. For any $g(X)$, we have:

$$\begin{aligned}\mathbb{E}[\mathcal{L}(f(X,Y),g(X))] &= \mathbb{E}_X\left[\mathbb{E}_Y\left[\mathcal{L}(f(X,Y),g(X))|X\right]\right] \\ &\geq \mathbb{E}_X\left[\min_g \mathbb{E}_Y\left[\mathcal{L}(f(X,Y),g(X))|X\right]\right] \\ &= \mathbb{E}_X\left[\min_c \mathbb{E}_Y\left[\mathcal{L}(f(X,Y),c)|X\right]\right].\end{aligned}$$

With these two inequalities, we conclude. □

Next, we prove the following lemma. This straightforward, and almost trivial, result is provided as a separate lemma because we simplify terms in this manner quite often throughout our proofs.

Lemma 5. *Let $\mathcal{L}$ be a convex function and let X, Y be two random variables. Then,*

$$\begin{aligned}&\min_f \mathbb{E}_{X,Y}\left[P(X'=X)\mathcal{L}(f(X'), S(X,Y))\right] + \mathbb{E}_{X,Y}\left[\sum_{X'\neq X} P(X')\mathcal{L}\left(f(X'), S(X,Y)\right)\right] \\ &\leq \min_f\{\mathbb{E}_{X,Y}\left[\mathcal{L}(f(X), S(X,Y))\right]\} + \epsilon\end{aligned}$$

where $\epsilon = \max_{X'\neq X, X, Y}\{\mathcal{L}(f^*(X'), S(X,Y)\}$ *and* $f^* = \arg\min_f\{\mathbb{E}_{X,Y}\left[\mathcal{L}(f(X), S(X,Y)\right]\}$

Proof.

$$\begin{aligned}&\min_f \mathbb{E}_{X,Y}\left[P(X'=X)\mathcal{L}(f(X'), S(X,Y))\right] + \mathbb{E}_{X,Y}\left[\sum_{X'\neq X} P(X')\mathcal{L}\left(f(X'), S(X,Y)\right)\right] \\ &\leq \min_f \mathbb{E}_{X,Y}\left[\mathcal{L}(f(X), S(X,Y))\right] + \mathbb{E}_{X,Y}\left[\sum_{X'\neq X} P(X')\mathcal{L}\left(f(X'), S(X,Y)\right)\right] \\ &\leq \min_f\{\mathbb{E}_{X,Y}\left[\mathcal{L}(f(X), S(X,Y))\right]\} + \mathbb{E}_{X,Y}\left[\sum_{X'\neq X} P(X')\mathcal{L}\left(f^*(X'), S(X,Y)\right)\right] \\ &\leq \min_f\{\mathbb{E}_{X,Y}\left[\mathcal{L}(f(X), S(X,Y))\right]\} + \mathbb{E}_{X,Y}\left[\sum_{X'\neq X} P(X')\epsilon\right] \\ &\leq \min_f\{\mathbb{E}_{X,Y}\left[\mathcal{L}(f(X), S(X,Y))\right]\} + \epsilon\end{aligned}$$

□

C.1 Definitions

Note that the informal definitions provided in §3.1 for expected observation advantage and maximum inference disadvantage have some deviations from the mathematical definitions. The informal definitions only attempt to provide intuition for what these terms might mean but they are not sufficient to describe the full construction.

To define the terms formally, we, first, analyze the effect of reducing context horizon. We show that, provided action horizon is constant, decreasing context horizon causes performance of the optimal policy to decrease.

Consider a (c, h)-policy whose probability of taking action a_t at time t in a chunk generated at t is referred to as

$$\pi_{(c,h)} := \pi_{(c,h)}(a_t | s_{t-c:t}).$$

On the other hand, consider a $(c+1, h)$-policy whose probability of taking action a_t in a chunk generated at time t is referred to as

$$\pi_{(c+1,h)} := \pi_{(c+1,h)}(a_t | s_{t-c-1:t}).$$

Lastly, consider a $(k, 1)$-expert whose probability of taking action a_t at time t is π^*.

Proposition 6 (Backward Context is valuable)**.** *Let $\mathcal{L}$ be a non-linear, convex function. Let $c < k$. Let $G := \{a_t, s_{t-k:t-c-1}, z_{t-k:t}\}$ and let $C := \{s_{t-c:t}\}$. Then,*

$$\min_{\pi_{(c+1,h)}} \mathbb{E}_G\left[\mathcal{L}(\pi_{(c+1,h)}, \pi^*)\Big| C\right] \leq \min_{\pi_{(c,h)}} \mathbb{E}_G\left[\mathcal{L}(\pi_{(c,h)}, \pi^*)\Big| C\right]$$

Proof. We refer to the class of functions that accept a_t and $s_{t-c-1:t}$ as inputs as $X_{+,+}$. Similarly, the class of functions that do not accept a_t as inputs but accept $s_{t-c-1:t}$ as inputs is X_+. The function class that accepts only $s_{t-c:t}$ and not s_{t-c-1} or a_t as inputs are elements of X_-. Lastly, the function class that accepts $s_{t-c:t}$ and a_t as inputs, but not s_{t-c-1}, are elements of $X_{-,-}$.

$$
\begin{aligned}
&\min_{\pi_{(c+1,h)} \in X_{+,+}} \mathbb{E}_G \left[\mathcal{L}(\pi_{(c+1,h)}, \pi^* \middle| C \right] \\
&= \mathbb{E}_{a_t} \left[\min_{\pi'_{(c+1,h)} \in X_+} \mathbb{E}_{s_{t-c-1}} \left[\mathbb{E}_{s_{t-k:t-c-2}, z_{t-k:t}} \left[\mathcal{L}(\pi'_{(c+1,h)}, \pi^*) \middle| a_t, s_{t-c-1}, C \right] \middle| a_t, C \right] \middle| C \right] \\
&\hfill \text{(Lemma 4)} \\
&= \mathbb{E}_{a_t} \left[\mathbb{E}_{s_{t-c-1}} \left[\min_{\pi'_{(c,h)} \in X_-} \mathbb{E}_{s_{t-k:t-c-2}, z_{t-k:t}} \left[\mathcal{L}(\pi'_{(c,h)}, \pi^*) \middle| a_t, s_{t-c-1}, C \right] \middle| a_t, C \right] \middle| C \right] \quad \text{(Lemma 4)} \\
&\le \mathbb{E}_{a_t} \left[\min_{\pi'_{(c,h)} \in X_-} \mathbb{E}_{s_{t-c-1}} \left[\mathbb{E}_{s_{t-k:t-c-2}, z_{t-k:t}} \left[\mathcal{L}(\pi'_{(c,h)}, \pi^*) \middle| a_t, s_{t-c-1}, C \right] \middle| a_t, C \right] \middle| C \right] \\
&\hfill \text{(Jensen's inequality)} \\
&= \min_{\pi_{(c,h)} \in X_{-,-}} \mathbb{E}_{a_t} \left[\mathbb{E}_{s_{t-c-1}} \left[\mathbb{E}_{s_{t-k:t-c-2}, z_{t-k:t}} \left[\mathcal{L}(\pi_{(c,h)}, \pi^*) \middle| a_t, s_{t-c-1}, C \right] \middle| a_t, C \right] \middle| C \right] \quad \text{(Lemma 4).}
\end{aligned}
$$

Use the law of total expectation to conclude. □

Now, we formalize the definitions of *Expected Observation Advantage* and *Maximum Inference Disadvantage*.

Recall that, in §3.1, we have two policies: $\pi_{(c,h)}$ and $\pi_{(c,h+d)}$; the former sees more recent states while the latter remembers more past states. First, we define an agent that gets access to all the information that both learners, combined, have: a $(c+d,h)$-policy whose probability of taking action a_t in a chunk generated at time $t-h$ is

$$\pi_{(c+d,h)} \coloneqq \pi_{(c+d,h)}(a_t | s_{t-h-d-c:t-h}, a_{t-h:t-1}).$$

Observe that $\pi_{(c+d,h)}$ has access to more context than $\pi_{(c,h)}$, particularly the knowledge of states $s_{t-h-c-d:t-h-c-1}$.

Definition (Expected Observation Advantage (α_b)). We know, using Proposition 6, $\pi_{(c+d,h)}$ has lower divergence with respect to π^* than $\pi_{(c,h)}$. We say that the advantage $\pi_{(c+d,h)}$ gets from the extra information is α_b. More formally, we say that

$$0 \le \alpha_b = \min_{\pi_{(c,h)}} \mathbb{E}_G \left[\mathcal{L}(\pi_{(c,h)}, \pi^*)) \middle| C \right] - \min_{\pi_{(c+d,h)}} \mathbb{E}_G \left[\mathcal{L}(\pi_{(c+d,h)}, \pi^*) \middle| C \right] \tag{9}$$

where $C = \{s_{t-h-c:t-h}, a_{t-h:t-1}\}$ and $G = \{a_t, s_{t-k:t-h-c-1}, s_{t-h+1:t}, z_{t-k:t}\}$. In particular, $\alpha_b = 0$ when $s_{t-h-d-c:t-h-c-1}$ can be deterministically inferred by $\pi_{(c,h)}$ or when the expert policy is independent of them. However, this is extremely unlikely since $\pi_{(c,h)}$ does not know the actions taken in those time steps (even more unlikely in a stochastic environment) and the expert's action depends on the last k time steps.

Definition (Maximum Inference Disadvantage (ϵ_f)). Consider the maximum divergence that can be accumulated by the $(c,h+d)$-policy from not knowing the recent states at time steps $s_{t-h-d+1:t-h}$ and let that be ϵ_f. More formally, we say that, for fixed C from Proposition 1, any state in $\mathcal{S}^-$ and any $z_{t-k:t}$ and any $\hat{s}_{t-h-d+1:t-h} \neq s_{t-h-d+1:t-h}$:

$$\mathcal{L}(\pi_{(c+d,h)}(a_t | s_{t-h-d-c:t-h-d}, \hat{s}_{t-h-d+1:t-h} \neq s_{t-h-d+1:t-h}, a_{t-h:t-1}), \pi^*) \le \epsilon_f. \tag{10}$$

Here, $\pi_{(c+d,h)} \coloneqq \arg\min_{\pi_{(c+d,h)}} \mathbb{E}_G[\mathcal{L}(\pi_{(c+d,h)}, \pi^*)|C]$ is the optimal $(c+d,h)$-policy.

Intuitively, maximum inference disadvantage captures the maximum dependency on relative *time steps* whereas expected observation advantage captures the expected advantage from observing some given *states*.

To define α_f and ϵ_b, we prove a second version of Proposition 6. Consider a (c, h)-policy whose probability of taking action a_t at time t in a chunk generated at t is referred to as

$$\pi_{(c,h)} := \pi_{L(c,h)}(a_t|s_{t-c:t}).$$

On the other hand, consider a $(c-1, h+1)$-policy whose probability of taking action a_t in a chunk generated at time $t-1$ is referred to as

$$\pi_{(c-1,h+1)} := \pi_{(c-1,h+1)}(a_t|s_{t-c:t-1}).$$

Lastly, consider a $(k, 1)$-expert whose probability of taking action a_t at time t is π^*.

Proposition 7 (Forward Context is valuable)**.** *Let $\mathcal{L}$ be a non-linear, convex function. Let $c < k$. Let $G := \{a_t, s_{t-k:t-c-1}, s_t, z_{t-k:t}\}$ and let $C := \{s_{t-c:t-1}, a_{t-1}\}$. Then,*

$$\min_{\pi_{(c,h)}} \mathbb{E}_G \left[\mathcal{L}(\pi_{(c,h)}, \pi^*)\middle|C\right] \le \min_{\pi_{(c-1,h+1)}} \mathbb{E}_G \left[\mathcal{L}(\pi_{(c-1,h+1)}, \pi^*)\middle|C\right]$$

Proof. The proof is similar to that of Proposition 6. We refer to the class of functions that accept a_t and $s_{t-c:t}$ as inputs as $X_{+,+}$. Similarly, the class of functions that do not accept a_t as inputs but accept $s_{t-c:t}$ as inputs is X_+. The function class that accepts only $s_{t-c:t-1}$ and not s_t or a_t as inputs are elements of X_-. Lastly, the function class that accepts $s_{t-c:t-1}$ and a_t as inputs, but not s_t, are elements of $X_{-,-}$.

$$\begin{aligned}
&\min_{\pi_{(c,h)}\in X_{+,+}} \mathbb{E}_G \left[\mathcal{L}(\pi_{(c,h)}, \pi^*\middle|C\right] \\
&= \mathbb{E}_{a_t}\left[\min_{\pi'_{(c,h)}\in X_+} \mathbb{E}_{s_t}\left[\mathbb{E}_{s_{t-k:t-c-1},z_{t-k:t}}\left[\mathcal{L}(\pi'_{(c,h)}, \pi^*)\middle|a_t, s_t, C\right]\middle|a_t, C\right]\middle|C\right] && \text{(Lemma 4)} \\
&= \mathbb{E}_{a_t}\left[\mathbb{E}_{s_t}\left[\min_{\pi'_{(c-1,h+1)}\in X_-} \mathbb{E}_{s_{t-k:t-c-2},z_{t-k:t}}\left[\mathcal{L}(\pi'_{(c-1,h+1)}, \pi^*)\middle|a_t, s_t, C\right]\middle|a_t, C\right]\middle|C\right] && \text{(Lemma 4)} \\
&\le \mathbb{E}_{a_t}\left[\min_{\pi'_{(c-1,h+1)}\in X_-} \mathbb{E}_{s_t}\left[\mathbb{E}_{s_{t-k:t-c-1},z_{t-k:t}}\left[\mathcal{L}(\pi'_{(c-1,h+1)}, \pi^*)\middle|a_t, s_t, C\right]\middle|a_t, C\right]\middle|C\right] \\
& && \text{(Jensen's inequality)} \\
&= \min_{\pi_{(c-1,h+1)}\in X_{-,-}} \mathbb{E}_{a_t}\left[\mathbb{E}_{s_t}\left[\mathbb{E}_{s_{t-k:t-c-1},z_{t-k:t}}\left[\mathcal{L}(\pi_{(c-1,h+1)}, \pi^*)\middle|a_t, s_t, C\right]\middle|a_t, C\right]\middle|C\right] && \text{(Lemma 4).}
\end{aligned}$$

Use the law of total expectation to conclude. □

Using this, we can define ϵ_b and α_f in a similar manner:

Definition (Expected Observation Advantage (α_f)). Recall that we have two models: $\pi_{(c,h)}$ and $\pi_{(c,h+d)}$ and a hypothetical $(c, h+d)$-policy that has access to all the information both our learners have (as in Eq. (9) and Eq. (10)). Observe that $\pi_{(c+d,h)}$ has access to more context than $\pi_{(c,h+d)}$, particularly the knowledge of states $s_{t-h-d+1:t-h}$. Therefore, we know, using Proposition 7, $\pi_{(c+d,h)}$ has lower divergence with respect to π^* than $\pi_{(c,h+d)}$. We say that the advantage $\pi_{(c+d,h)}$ gets from the extra information is α_f. More formally, we say that

$$0 \le \alpha_f = \min_{\pi_{(c,h+d)}} \mathbb{E}_G \left[\mathcal{L}(\pi_{(c,h+d)}, \pi^*))\middle|C\right] - \min_{\pi_{(c+d,h)}} \mathbb{E}_G \left[\mathcal{L}(\pi_{(c+d,h)}, \pi^*)\middle|C\right] \qquad (11)$$

where $C = \{s_{t-h-d-c:t-h-d}, a_{t-h-d:t-1}\}$ and $G = \{a_t, s_{t-k:t-h-d-c-1}, s_{t-h-d+1:t}, z_{t-k:t}\}$. In particular, $\alpha_f = 0$ when $\pi_{(c,h+d)}$ can infer $s_{t-h-d+1:t-h}$ perfectly *i.e.* when the environment is completely static with $P_f = 1$. This makes sense–in the static environment, observing these states does not provide any advantage since the optimal $\pi_{(c,h+d)}$ can infer these states anyway using the actions taken at those time steps.

Note how this formal definition has some difference from the informal one. In particular, π_h only observes $s_{t-h-c:t-h}$, not necessarily all of $s_{t-h-d+1:t-h}$. So, some of the value of α_f can be informally called the "advantage" π_h gets over π_{h+d}, but not necessarily all of it. Nevertheless, our proofs will be using this formal definition.

Definition (Maximum Inference Disadvantage (ϵ_b)). Consider the maximum divergence that can be accumulated by the (c, h)-model from not knowing the past states $s_{t-h-d-c:t-h-c-1}$ and let that be ϵ_b. More formally, we say that, for fixed C from Proposition 1, any state in $\mathcal{S}^-$ and any $z_{t-k:t}$ and any $\hat{s}_{t-h-d-c:t-h-c-1} \neq s_{t-h-d-c:t-h-c-1}$:

$$\mathcal{L}(\pi_{(c+d,h)}(a_t|\hat{s}_{t-h-d-c:t-h-c-1} \neq s_{t-h-d-c:t-h-c-1}, s_{t-h-c:t-h}, a_{t-h:t-1}), \pi^*) \leq \epsilon_b. \tag{12}$$

Here, $\pi_{(c+d,h)} \coloneqq \arg\min_{\pi_{(c+d,h)}} \mathbb{E}_G[\mathcal{L}(\pi_{(c+d,h)}, \pi^*)|C]$ is the optimal $(c+d, h)$-policy.

The intuitive relationship between α_f and ϵ_f (and the same for α_b and ϵ_b) holds:

Proposition 8. *$\alpha_f \leq \epsilon_f$ and $\alpha_b \leq \epsilon_b$.*

Proof. We prove the first inequality; the second can be proven in the same manner. We use Assumption 2 to write $\pi_{(c,h+d)} = \mathbb{E}_{s_{t-h-d+1:t-h} \sim P}\left[\pi_{(c+d,h)} \mid s_{t-h-d-c:t-h-d}, a_{t-h-d:t-1}\right]$ where P is the environment's transition dynamics. Let

$$P_{\text{correct inference}} = P(\hat{s}_{t-h-d+1:t-h} = s_{t-h-d+1:t-h}|s_{t-h-d-c:t-h-d}, a_{t-h-d:t-1})$$

and

$$P_{\text{incorrect inference}} = P(\hat{s}_{t-h-d+1:t-h} \neq s_{t-h-d+1:t-h}|s_{t-h-d-c:t-h-d}, a_{t-h-d:t-1}).$$

Then,

$$\begin{aligned}
\alpha_f &= \min_{\pi_{(c,h+d)}} \mathbb{E}_G\left[\mathcal{L}(\pi_{(c,h+d)}, \pi^*)\middle|C\right] - \min_{\pi_{(c+d,h)}} \mathbb{E}_G\left[\mathcal{L}(\pi_{(c+d,h)}, \pi^*)\middle|C\right] \\
&= \min_{\pi_{(c+d,h)}} \mathbb{E}_G\left[\mathcal{L}(P_{\text{correct inference}}\pi_{(c+d,h)} + P_{\text{incorrect inference}}\pi_{(c+d,h)}, \pi^*)\middle|C\right] \\
&- \min_{\pi_{(c+d,h)}} \mathbb{E}_G\left[\mathcal{L}(\pi_{(c+d,h)}, \pi^*)\middle|C\right] \\
&\leq \min_{\pi_{(c+d,h)}} \{\mathbb{E}_G\left[P_{\text{incorrect inference}}\mathcal{L}(\pi_{(c+d,h)}(\text{conditioning on incorrect inference}), \pi^*)\middle|C\right] \\
&+ \mathbb{E}_G\left[P_{\text{correct inference}}\mathcal{L}(\pi_{(c+d,h)}(\text{conditioning on correct inference}), \pi^*)\middle|C\right]\} \\
&- \min_{\pi_{(c+d,h)}} \mathbb{E}_G\left[\mathcal{L}(\pi_{(c+d,h)}, \pi^*)\middle|C\right] \qquad \text{(Convexity)} \\
&\leq \mathbb{E}_G\left[P_{\text{incorrect inference}}\mathcal{L}(\hat{\pi}^*_{(c+d,h}(\text{conditioning on incorrect inference}), \pi^*)\middle|C\right] \\
&+ \mathbb{E}_G\left[\mathcal{L}(\pi^*_{(c+d,h)}, \pi^*)\middle|C\right] - \mathbb{E}_G\left[\mathcal{L}(\pi^*_{(c+d,h)}, \pi^*)\middle|C\right] \\
&\qquad \text{(Bounding probabilities by 1 and Lemma 5)} \\
&\leq \mathbb{E}_G\left[P_{\text{incorrect inference}}\epsilon_f \mid C\right] \\
&\leq \epsilon_f
\end{aligned}$$

Here, $\pi^*_{(c+d,h)} \coloneqq \arg\min_{\pi_{(c+d,h)}} \mathbb{E}_G\left[\mathcal{L}(\pi_{(c+d,h)}, \pi^*)\middle|C\right]$. □

Definition (Forward and Backward Inference). For a fixed time step t and C (as in §3.1), consider the time steps $\{t-h-d+1 : t-h\}$. Define

$$P_f(t') \coloneqq P(S_{t'} = g_{t'}|S_{t'-1} = g_{t'-1}, A_{t'-1} = a_{t'-1})$$

for all $t' \in [t-h-d+1 : t-h]$ with $g_{t'}, g_{t'-1}, a_{t'-1}$ being the ground truth states and action in the deterministic environment. We assume that $P_f(t') = 1$ in a deterministic environment. In a stochastic environment, $P_f(t') < 1$ for all t' and as the stochasticity increases, these values decrease and approach 0. Then, we define

$$P_f \coloneqq \sup\{P_f(t')|t' \in [t-h-d+1 : t-h]\}.$$

Now, consider the time steps $\{t-h-d-c : t-h-c-1\}$. Define

$$P_b(t') \coloneqq P(S_{t'} = g_{t'} | S_{t'+1} = g_{t'+1})$$

for all $t' \in [t-h-d-c : t-h-c-1]$. Since this is not conditioned on any action and as conditioning on $a_{t'}$ reduces entropy, we assume that $P_b(t') < 1$ for all $t' \in [t-h-d-c : t-h-c-1]$ in all environments. As stochasticity increases, $P_b(t')$ decreases and approaches 0. Then, define

$$P_b \coloneqq \sup\{P_b(t') | t' \in [t-h-d-c : t-h-c-1]\}.$$

C.2 Discussion on Assumption 2

Before we prove the next result, we briefly discuss Assumption 2. Note that, using law of total probability, we can already write:

$$\begin{aligned}
&\pi_{(c,h)}(a_t | s_{t-h-c:t-h}, a_{t-h:t-1}) \\
&= \sum_{\substack{s_{t-k:t-h-c-1}, \\ s_{t-h+1:t}}} \hat{P}(s_{t-k:t-h-c-1}, s_{t-h+1:t} | s_{t-h-c:t-h}, a_{t-h:t-1}) \pi_{(k,1)}(a_t | s_{t-k:t}, a_{t-h:t-1}) \\
&\hspace{25em} \text{(Law of Total Probability)} \\
&= \mathbb{E}_{s_{t-k:t-h-c-1}, s_{t-h+1:t} \sim \hat{P}} \left[\pi_{(k,1)}(a_t | s_{t-k:t}) | s_{t-h-c:t-h}, a_{t-h:t-1} \right].
\end{aligned}$$

Here $\hat{P}$ is the policy's *learned* environment dynamics. Assumption 2 allows us to replace $\hat{P}$ with the true environment dynamics P. In other words, we assume that an optimal policy has already learned these distributions as optimally as possible. Given we are talking about the optimal policy trained with infinite data, this is a reasonable assumption. Note that this does not make inference trivial - the policy learns the distribution but, given the distribution has non-zero entropy, the policy can still infer the wrong state.

Thus, using assumption 2, we can write the optimal policy as

$$\pi_{(c,h)} = \mathbb{E}_{\hat{s}_{t-k:t-h-c-1}, \hat{s}_{t-h+1:t} \sim P} \left[\pi_{(k,1)} | s_{t-h-c:t-h}, a_{t-h:t-1} \right]$$

where $P(S_{t+1} = s_{t+1} | S_t = s_t, a_t)$ is the environment's transition dynamics and $\pi_{(k,1)}$ is the distribution of a_t under a $(k,1)$-model.

C.3 Consistency-Reactivity Inequalities

Now we prove the Consistency-Reactivity Inequalities. We prove the upper and lower bound separately:

Proposition 1 (Consistency-Reactivity Inequalities - Upper Bound). Let $\mathcal{L}$ be a non-linear, convex loss function. Let $\mathcal{S}^+ \subset \{s_{t-k:t}\}$ be the states both models observe and let $\mathcal{S}^- \coloneqq \{s_{t-k:t}\} \setminus \mathcal{S}^+$. Let $C \coloneqq \{a_{t-h-d:t-1}\} \cup \mathcal{S}^+$, $G \coloneqq \{a_t, z_{t-k:t}\} \cup \mathcal{S}^-$. Then, we can bound the expected loss of the $(c, h+d)$-policy and the (c, h)-policy as:

$$\min_{\pi_{h+d}} \mathbb{E}_G \left[\mathcal{L}(\pi_{h+d}, \pi^*) | C \right] \leq \min_{\pi_h} \mathbb{E}_G \left[\mathcal{L}(\pi_h, \pi^*) | C \right] - \alpha_b + \epsilon_f (1 - P_f^{2d}).$$

Proof. For ease of notation, we will write $x_{a:}^{b}$ to mean $x_{a:b}$. Additionally, for greater clarity, we will explicitly include the context length of each model, so $\pi_{(c,h)} = \pi_h$ and $\pi_{(c,h+d)} = \pi_{h+d}$. We start by writing, using Assumption 1,

$$\begin{aligned}
&\pi_{(c,h+d)}(a_t | s_{t-h-d-c:t-h-d}, a_{t-h-d:t-1}) \\
&= \pi_{(c,h+d)}(a_t | s_{t-h-d-c:}^{t-h-d}, a_{t-h-d:}^{t-1}) \\
&= \mathbb{E}_{\hat{s}_{t-h-d+1:t-h}} \left[\pi_{(c+d,h)}(a_t | s_{t-h-d-c:}^{t-h-d}, \hat{s}_{t-h-d+1:}^{t-h}, a_{t-h-d:}^{t-1}) \middle| s_{t-h-d-c:}^{t-h-d}, a_{t-h-d:}^{t-1} \right].
\end{aligned}$$

Using this, we expand the left hand side of our inequality:

$$\begin{aligned}
&\min_{\pi_{(c,h+d)}} \mathbb{E}_G \left[\mathcal{L}(\pi_{(c,h+d)}, \pi^*)|C\right] \\
&= \min_{\pi_{(c+d,h)}} \mathbb{E}_G \Big[\mathcal{L}(\mathbb{E}_{\hat{s}_{t-h-d+1:t-h}} \Big[\pi_{(c+d,h)}\Big|C\Big], \pi^*)|C\Big] \\
&= \min_{\pi_{(c+d,h)}} \mathbb{E}_G \Big[\mathcal{L}(P(\hat{s}^{t-h}_{t-h-d+1:}|s^{t-h-d}_{t-h-d-c:}, a^{t-h-1}_{t-h-d:})\pi_{(c+d,h)}(a_t|\cdots, g^{t-h}_{t-h-d+1:}) + \\
&\sum_{\substack{\hat{s}_{t-h-d+1:t-h} \\ \text{not all } g_{t'}}} P(\hat{s}^{t-h}_{t-h-d+1:}|s_{t-h-d}, a^{t-h-1}_{t-h-d:})\ \pi_{(c+d,h)}(a_t|\cdots, \hat{s}^{t-h}_{t-h-d+1:}), \pi^*)|C\Big] \\
&\leq \min_{\pi_{(c+d,h)}} \mathbb{E}_G \Big[\mathcal{L}(P^d_f \pi_{(c+d,h)}(a_t|\cdots, g^{t-h}_{t-h-d+1:}) + \\
&\sum_{\substack{\hat{s}_{t-h-d+1:t-h} \\ \text{not all } g_{t'}}} P(\hat{s}^{t-h}_{t-h-d+1:}|s_{t-h-d}, a^{t-h-1}_{t-h-d:})\ \pi_{(c+d,h)}(a_t|\cdots, \hat{s}^{t-h}_{t-h-d+1:}), \pi^*)|C\Big]
\end{aligned}$$

where we computed the expectation $\mathbb{E}_{\hat{s}_{t-h-d+1:t-h}} \Big[\pi_{(c+d,h)}\Big|C\Big]$ by grouping into two terms : one where every $\hat{s}_{t-h-d+1:t-h} = g_{t-h-d+1:t-h}$ and one where there is at least one term $\hat{s}_i$ that is not g_i. This grouping was done using the definition of noise in our environment. We introduce the following notation here

$$\hat{P}_{\neq g_{t'}} := \sum_{\substack{\hat{s}_{t-h-d+1:t-h} \\ \text{not all } g_{t'}}} P(\hat{s}^{t-h}_{t-h-d+1:}|s_{t-h-d}, a^{t-h-1}_{t-h-d:}).$$

Similarly,

$$P_{\neq g_{t'}} := \sum_{\substack{s_{t-h-d+1:t-h} \\ \text{not all } g_{t'}}} P(s^{t-h}_{t-h-d+1:}|s_{t-h-d}, a^{t-h-1}_{t-h-d:}).$$

With this notation, we continue our expansion:

$$\begin{aligned}
&\min_{\pi_{(c,h+d)}} \mathbb{E}_G \left[\mathcal{L}(\pi_{(c,h+d)}, \pi^*)|C\right] \\
&\leq \min_{\pi_{(c+d,h)}} \mathbb{E}_G \Big[\mathcal{L}(P^d_f \pi_{(c+d,h)}(a_t|\cdots, g^{t-h}_{t-h-d+1:}) + \\
&\sum_{\substack{\hat{s}_{t-h-d+1:t-h} \\ \text{not all } g_{t'}}} P(\hat{s}^{t-h}_{t-h-d+1:}|s_{t-h-d}, a^{t-h-1}_{t-h-d:})\ \pi_{(c+d,h)}(a_t|\cdots, \hat{s}^{t-h}_{t-h-d+1:}), \pi^*)|C\Big] \\
&\leq \min_{\pi_{(c+d,h)}} \mathbb{E}_G \Big[\mathcal{L}(P^d_f \pi_{(c+d,h)}(a_t|\cdots, g^{t-h}_{t-h-d+1:}) + \hat{P}_{\neq g_{t'}}\ \pi_{(c+d,h)}(a_t|\cdots, \hat{s}^{t-h}_{t-h-d+1:}), \pi^*)|C\Big] \\
&\leq \min_{\pi_{(c+d,h)}} \mathbb{E}_G \Big[P^d_f \mathcal{L}(\pi_{(c+d,h)}(a_t|\cdots, g^{t-h}_{t-h-d+1:}), \pi^*)|C\Big] \\
&+ \mathbb{E}_G \Big[\hat{P}_{\neq g_{t'}} \mathcal{L}(\pi_{(c+d,h)}(a_t|\cdots, \hat{s}^{t-h}_{t-h-d+1:}), \pi^*)|C\Big]
\end{aligned}$$

where we got the inequality using the fact that $\mathcal{L}$ is a convex function and, thus, convex in each argument. Next, we take the expectation over $s_{t-h-d+1:t-h}$ by grouping the terms into two: one where every $s_{t-h-d+1:t-h} = g_{t-h-d+1:t-h}$ and one where there is at least one term $s_i \neq g_i$. Then, with some suppression of notation in the expression of $\pi_{(c+d,h)}$ and $G' := G \setminus \{a_t, s_{t-h-d+1:t-h}\}$:

$$\begin{aligned}
&\min_{\pi_{(c,h+d)}} \mathbb{E}_G \left[\mathcal{L}(\pi_{(c,h+d)}, \pi^*)|C\right] \\
&\leq \min_{\pi_{(c+d,h)}} \mathbb{E}_{a_t} \\
&\Big[P^d_f\ P^d_f \mathbb{E}_{G'} \Big[\mathcal{L}(\pi_{(c+d,h)}(...\hat{s}^{t-h}_{t-h-d+1:} = g^{t-h}_{t-h-d+1:}), \pi^*)\Big|..., s^{t-h}_{t-h-d+1:} = g^{t-h}_{t-h-d+1:}\Big]
\end{aligned}$$

$$
\begin{aligned}
&+ P_{\neq g_{t'}}\ P_f^d \mathbb{E}_{G'}\left[\mathcal{L}(\pi_{(c+d,h)}(...\hat{s}_{t-h-d+1:}^{t-h} = g_{t-h-d+1:}^{t-h}), \pi^*)\Big|..., s_{t-h-d+1:}^{t-h} \neq g_{t-h-d+1:}^{t-h}\right] \\
&+ P_f^d \hat{P}_{\neq g_{t'}} \mathbb{E}_{G'}\left[\mathcal{L}(\pi_{(c+d,h)}(...\hat{s}_{t-h-d+1:}^{t-h} \neq g_{t-h-d+1:}^{t-h}), \pi^*)\Big|..., s_{t-h-d+1:}^{t-h} = g_{t-h-d+1:}^{t-h}\right] \\
&+ P_{\neq g_{t'}} \hat{P}_{\neq g_{t'}} \mathbb{E}_{G'}\left[\mathcal{L}(\pi_{(c+d,h)}(..., \hat{s}_{t-h-d+1:}^{t-h} \neq g_{t-h-d+1:}^{t-h}), \pi^*)\Big|..., s_{t-h-d+1:}^{t-h} \neq g_{t-h-d+1:}^{t-h}\right]
\end{aligned}
$$

Now, we group all the terms into two - one representing where the learner's simulation matches the reality and one where it does not. Continuing from where we left off, first, define $P^f_{\hat{s}=s} := P(s_{t-h-d+1:}^{t-h} | s_{t-h-d}, a_{t-h-d:}^{t-h-1})$

$$
\begin{aligned}
&\min_{\pi_{(c,h+d)}} \mathbb{E}_G\left[\mathcal{L}(\pi_{(c,h+d)}, \pi^*) | C\right] \\
&\leq \min_{\pi_{(c+d,h)}} \mathbb{E}_{a_t} \\
&\Big[P_f^d\ P_f^d \mathbb{E}\left[\mathcal{L}(\pi_{(c+d,h)}(a_t|..., \hat{s}_{t-h-d+1:}^{t-h} = g_{t-h-d+1:}^{t-h} = s_{t-h-d+1:}^{t-h}), \pi^*)\Big|..., s_{t-h-d+1:}^{t-h} = g_{t-h-d+1:}^{t-h}\right] \\
&+ P_{\neq g_{t'}}\ P^f_{\hat{s}=s} \mathbb{E}\left[\mathcal{L}(\pi_{(c+d,h)}(a_t|..., \hat{s}_{t-h-d+1:}^{t-h} = s_{t-h-d+1:}^{t-h} \neq g_{t-h-d+1:}^{t-h}), \pi^*)\Big|..., s_{t-h-d+1:}^{t-h} \neq g_{t-h-d+1:}^{t-h}\right] \\
&+ P_f^d \hat{P}_{\neq g_{t'}} \mathbb{E}\left[\mathcal{L}(\pi_{(c+d,h)}(a_t|..., \hat{s}_{t-h-d+1:}^{t-h} \neq g_{t-h-d+1:}^{t-h} = s_{t-h-d+1:}^{t-h}), \pi^*)\Big|..., s_{t-h-d+1:}^{t-h} = g_{t-h-d+1:}^{t-h}\right] \\
&+ P_{\neq g_{t'}} P_f^d \mathbb{E}\left[\mathcal{L}(\pi_{(c+d,h)}(a_t|..., \hat{s}_{t-h-d+1:}^{t-h} = g_{t-h-d+1:}^{t-h} \neq s_{t-h-d+1:}^{t-h}), \pi^*)\Big|..., s_{t-h-d+1:}^{t-h} \neq g_{t-h-d+1:}^{t-h}\right] \\
&+ P_{\neq g_{t'}} P(\hat{s}_{t-h-d+1:}^{t-h} \neq s_{t'} | s_{t-h-d}, a_{t-h-d:}^{t-h-1}) \\
&\mathbb{E}\left[\mathcal{L}(\pi_{(c+d,h)}(a_t|..., \hat{s}_{t-h-d+1:}^{t-h} \neq s_{t-h-d+1:}^{t-h} \neq g_{t-h-d+1:}^{t-h}), \pi^*)\Big|..., s_{t-h-d+1:}^{t-h} \neq g_{t-h-d+1:}^{t-h}\right] | C, a_t\ ] \mid |C]
\end{aligned}
$$

For the match terms, we use the fact that $P_f^d \leq 1$ and $P^f_{\hat{s}=s} = P(\hat{s}_{t-h-d+1:}^{t-h} = s_{t-h-d+1:}^{t-h} | s_{t-h-d}, a_{t-h-d:}^{t-h-1}) \leq 1$. For the mismatch terms, we use the definition of ϵ_f and Lemma 5. Then, we continue:

$$
\begin{aligned}
&\leq \min_{\pi_{(c+d,h)}} \mathbb{E}_G\left[\mathcal{L}(\pi_{(c+d,h)}, \pi^*)\Big|C\right] && \text{(Simulation matches reality)} \\
&+ P_{\neq g_{t'}}\left[P_f^d \epsilon_f + \hat{P}_{\neq g_{t'}, s_{t'}} \epsilon_f\right] + P_f^d\left[\hat{P}_{\neq g_{t'}} \epsilon_f\right]. && \text{(Simulation does not match reality)}
\end{aligned}
$$

We simplify the mismatch terms further:

$$
\begin{aligned}
&\leq P_{\neq g_{t'}}\left[P_f^d \epsilon_f + (1 - P_f^d)\epsilon_f\right] + P_f^d\left[\hat{P}_{\neq g_{t'}} \epsilon_f\right] \\
&= P_{\neq g_{t'}} \epsilon_f + P_f^d \hat{P}_{\neq g_{t'}} \epsilon_f \\
&= P_{\neq g_{t'}} \epsilon_f + P_f^d\left[(1 - P_f^d)\right]\epsilon_f \\
&= (1 - P_f^d)\epsilon_f + P_f^d\left[(1 - P_f^d)\right]\epsilon_f \\
&= \epsilon_f \cdot \left[1 - P_f^d + P_f^d - P_f^{2d}\right] \\
&= \epsilon_f \cdot \left[1 - P_f^{2d}\right].
\end{aligned}
$$

Next, we simplify the match terms by using the definition of α_b:

$$
\min_{\pi_{(c+d,h)}} \mathbb{E}_G\left[\mathcal{L}(\pi_{(c+d,h)}, \pi^*)\Big|C\right] = \min_{\pi_{(c,h)}} \mathbb{E}_G\left[\mathcal{L}(\pi_{(c,h)}, \pi^*) | C\right] - \alpha_b.
$$

Substituting these two terms back in, we conclude. □

Now, we prove the lower bound:

Proposition 1 (Consistency-Reactivity Inequalities - Lower Bound). Let $\mathcal{L}$ be a non-linear, convex loss function. Let $\mathcal{S}^+ \subset \{s_{t-k:t}\}$ be the states both models observe and let $\mathcal{S}^- := \{s_{t-k:t}\} \setminus \mathcal{S}^+$.

Let $C \coloneqq \{a_{t-h-d:t-1}\} \cup \mathcal{S}^+$, $G \coloneqq \{a_t, z_{t-k:t}\} \cup \mathcal{S}^-$. Then, we can bound the expected loss of the $(c, h+d)$-policy and the (c, h)-policy as:

$$\min_{\pi_h} \mathbb{E}_G \left[\mathcal{L}(\pi_h, \pi^*) | C\right] \leq \min_{\pi_{h+d}} \mathbb{E}_G \left[\mathcal{L}(\pi_{h+d}, \pi^*) | C\right] - \alpha_f + \epsilon_b(1 - P_b^{2d}).$$

Proof. We proceed in a manner similar to the proof of the upper bound. For ease of notation, we will write $x_{a:}^{b}$ to mean $x_{a:b}$. Additionally, for greater clarity, we will explicitly include the context length of each model, so $\pi_{(c,h)} = \pi_h$ and $\pi_{(c,h+d)} = \pi_{h+d}$. We start by writing, using Assumption 1,

$$\begin{aligned}
&\min_{\pi_{(c,h)}} \mathbb{E}_G \left[\mathcal{L}(\pi_{(c,h)}, \pi^*) \mid C\right] \\
&= \min_{\pi_{(c+d,h)}} \mathbb{E}_G \left[\mathcal{L}(P(g_{t-h-d-c:}^{t-h-c-1} | s_{t-h-c}) \pi_{(c+d,h)} + \sum_{\substack{\hat{s}_{t-h-d-c:}^{t-h-c-1}, \\ \text{not all } g_{t'}}} P(\hat{s}_{t-h-d-c:}^{t-h-c-1} | s_{t-h-c}) \pi_{(c+d,h)}^{t}, \pi^*) \Big| C \right]. \\
&\leq \min_{\pi_{(c+d,h)}} \mathbb{E}_G \left[\mathcal{L}(P_b^d \pi_{(c+d,h)} + \sum_{\substack{\hat{s}_{t-h-d-c:}^{t-h-c-1}, \\ \text{not all } g_{t'}}} P(\hat{s}_{t-h-d-c:}^{t-h-c-1} | s_{t-h-c}) \pi_{(c+d,h)}^{t}, \pi^*) \Big| C \right].
\end{aligned}$$

We introduce the following notation here

$$\hat{P}_{\neq g_{t'}} \coloneqq \sum_{\substack{\hat{s}_{t-h-d-c:t-h-c-1} \\ \text{not all } g_{t'}}} P(\hat{s}_{t-h-d-c:}^{t-h-c-1} | s_{t-h-c}).$$

Similarly,

$$P_{\neq g_{t'}} \coloneqq \sum_{\substack{s_{t-h-d-c:t-h-c-1} \\ \text{not all } g_{t'}}} P(s_{t-h-d-c:}^{t-h-c-1} | s_{t-h-c}).$$

With this notation, we continue our expansion:

$$\begin{aligned}
&\min_{\pi_{(c,h)}} \mathbb{E}_G \left[\mathcal{L}(\pi_{(c,h)}, \pi^*) | C\right] \\
&\leq \min_{\pi_{(c+d,h)}} \mathbb{E}_G \Big[\mathcal{L}(P_b^d \pi_{(c+d,h)}(a_t | s_{t-h-c:}^{t-h}, g_{t-h-d-c:}^{t-h-c-1}, a_{t-h:}^{t-1}) + \\
&\qquad \hat{P}_{\neq g_{t'}} \pi_{(c+d,h)}(a_t | s_{t-h-c:}^{t-h}, \hat{s}_{t-h-d-c:}^{t-h-c-1}, a_{t-h:}^{t-1}), \pi^*) \mid C\Big] \\
&\leq \min_{\pi_{(c+d,h)}} \mathbb{E}_G \Big[P_b^d \mathcal{L}(\pi_{(c+d,h)}(a_t | \ldots, g_{t-h-d-c:}^{t-h-c-1}), \pi^*) \mid C\Big] + \\
&\qquad \mathbb{E}_G \Big[\hat{P}_{\neq g_{t'}} \mathcal{L}(\ \pi_{(c+d,h)}(a_t | \ldots, \hat{s}_{t-h-d-c:}^{t-h-c-1} \neq g_{t-h-d-c:}^{t-h-c-1}), \pi^*) \mid C\Big]
\end{aligned}$$

where we got the inequality using the fact that $\mathcal{L}$ is a convex function. Next, we take the expectation over $s_{t-h-d-c:t-h-c-1}$ by grouping the terms into two: one where every $s_{t-h-d-c:t-h-c-1} = g_{t-h-d-c:t-h-c-1}$ and one where there is at least one term $s_i \neq g_i$. Then, again suppressing some terms inside the expression of $\pi_{(c+d,h)}$:

$$
\begin{aligned}
&\min_{\pi_{(c,h)}} \mathbb{E}_G \left[\mathcal{L}(\pi_{(c,h)}, \pi^*)|C\right] \\
&\leq \min_{\pi_{(c+d,h)}} \mathbb{E}_{a_t} \\
&\Big[P_b^d \; P_b^d \mathbb{E}\left[\mathcal{L}(\pi_{(c+d,h)}(..., \hat{s}_{t-h-d-c:}^{t-h-c-1} = g_{t-h-d-c:}^{t-h-c-1} = s_{t-h-d-c:}^{t-h-c-1}), \pi^*)\Big|..., s_{t-h-d-c:}^{t-h-c-1} = g_{t-h-d-c:}^{t-h-c-1}\right] \\
&+ P_{\neq g_{t'}} \; P_b^d \mathbb{E}\left[\mathcal{L}(\pi_{(c+d,h)}(..., \hat{s}_{t-h-d-c:}^{t-h-c-1} = g_{t-h-d-c:}^{t-h-c-1} \neq s_{t-h-d-c:}^{t-h-c-1}), \pi^*)\Big|..., s_{t-h-d-c:}^{t-h-c-1} \neq g_{t-h-d-c:}^{t-h-c-1}\right] \\
&+ P_b^d \hat{P}_{\neq g_{t'}} \; \mathbb{E}\left[\mathcal{L}(\pi_{(c+d,h)}(..., \hat{s}_{t-h-d-c:}^{t-h-c-1} \neq g_{t-h-d-c:}^{t-h-c-1} = s_{t-h-d-c:}^{t-h-c-1}), \pi^*)\Big|..., s_{t-h-d-c:}^{t-h-c-1} = g_{t-h-d-c:}^{t-h-c-1}\right] \\
&+ P_{\neq g_{t'}} \hat{P}_{\neq g_{t'}} \; \mathbb{E}\left[\mathcal{L}(\pi_{(c+d,h)}(..., \hat{s}_{t-h-d-c:}^{t-h-c-1} \neq g_{t-h-d-c:}^{t-h-c-1}), \pi^*)\Big|..., s_{t-h-d-c:}^{t-h-c-1} \neq g_{t-h-d-c:}^{t-h-c-1} \;] \mid C, a_t\right] \mid C\Big]
\end{aligned}
$$

Now, we group all the terms into two - one representing where the learner's simulation matches the reality and one where it does not. Continuing from where we left off and defining $P_{\hat{s}=s}^b := P(\hat{s}_{t-h-d-c:}^{t-h-c-1} = s_{t-h-d-c:}^{t-h-c-1} | s_{t-h-c})$:

$$
\begin{aligned}
&\min_{\pi_{(c,h)}} \mathbb{E}_G \left[\mathcal{L}(\pi_{(c,h)}, \pi^*)|C\right] \\
&\leq \min_{\pi_{(c+d,h)}} \mathbb{E}_{a_t} \\
&\Big[P_b^d \; P_b^d \mathbb{E}\left[\mathcal{L}(\pi_{(c+d,h)}(a_t|..., \hat{s}_{t-h-d-c:}^{t-h-c-1} = g_{t-h-d-c:}^{t-h-c-1}), \pi^*)\Big|..., s_{t-h-d-c:}^{t-h-c-1} = g_{t-h-d-c:}^{t-h-c-1}\right] \\
&+ P_{\neq g_{t'}} \; P_{\hat{s}=s}^b \mathbb{E}\left[\mathcal{L}(\pi_{(c+d,h)}(a_t|..., \hat{s}_{t-h-d-c:}^{t-h-c-1} = s_{t-h-d-c:}^{t-h-c-1}), \pi^*)\Big|..., s_{t-h-d-c:}^{t-h-c-1} \neq g_{t-h-d-c:}^{t-h-c-1}\right] \\
&+ P_b^d \hat{P}_{\neq g_{t'}} \; \mathbb{E}\left[\mathcal{L}(\pi_{(c+d,h)}(a_t|..., s_{t-h-d-c:}^{t-h-c-1} \neq g_{t-h-d-c:}^{t-h-c-1}), \pi^*)\Big|..., s_{t-h-d-c:}^{t-h-c-1} = g_{t-h-d-c:}^{t-h-c-1}\right] \\
&+ P_{\neq g_{t'}} P_b^{2d} \; \mathbb{E}\left[\mathcal{L}(\pi_{(c+d,h)}(a_t|..., \hat{s}_{t-h-d-c:}^{t-h-c-1} = g_{t-h-d-c:}^{t-h-c-1}), \pi^*)\Big|..., s_{t-h-d-c:}^{t-h-c-1} \neq g_{t-h-d-c:}^{t-h-c-1} \;] \mid C, a_t\right] \mid C\Big] \\
&+ P_{\neq g_{t'}} P(\hat{s}_{t-h-d-c:}^{t-h-c-1} \neq g_{t-h-d-c:}^{t-h-c-1}, s_{t-h-d-c:}^{t-h-c-1} \mid s_{t-h-c}) \\
&\mathbb{E}\left[\mathcal{L}(\pi_{(c+d,h)}(a_t|..., \hat{s}_{t-h-d-c:}^{t-h-c-1} \neq s_{t-h-d-c:}^{t-h-c-1}), \pi^*)\Big|..., s_{t-h-d-c:}^{t-h-c-1} \neq g_{t-h-d-c:}^{t-h-c-1} \;] \mid C, a_t\right] \mid C\Big]
\end{aligned}
$$

For the match terms, we use the fact that $P_b^d \leq 1$ and $P(\hat{s}_{t-h-d-c:}^{t-h-c-1} = s_{t-h-d-c:}^{t-h-c-1} | s_{t-h-c:}^{t-h}) \leq 1$. For the mismatch terms, we use the definition of ϵ_b and Lemma 5. Then, we continue:

$$
\begin{aligned}
&\leq \min_{\pi_{(c+d,h)}} \mathbb{E}_G \left[\mathcal{L}(\pi_{(c+d,h)}, \pi^*)\Big|C\right] && \text{(Simulation matches reality)} \\
&+ P_{\neq g_{t'}} \left[P_b^d \epsilon_b + \hat{P}_{\neq g_{t'}, s_{t'}} \epsilon_b\right] + P_b^d \left[\hat{P}_{\neq g_{t'}} \epsilon_b\right]. && \text{(Simulation does not match reality)}
\end{aligned}
$$

We simplify the mismatch terms further:

$$
\begin{aligned}
&\leq P_{\neq g_{t'}} \left[P_b^d \epsilon_b + (1 - P_b^d)\epsilon_b\right] + P_b^d \left[\hat{P}_{\neq g_{t'}} \epsilon_b\right] \\
&= P_{\neq g_{t'}} \epsilon_b + P_b^d \hat{P}_{\neq g_{t'}} \epsilon_b \\
&= P_{\neq g_{t'}} \epsilon_b + P_b^d \left[(1 - P_b^d)\right] \epsilon_b \\
&= (1 - P_b^d)\epsilon_b + P_b^d \left[(1 - P_b^d)\right] \epsilon_b \\
&= \epsilon_b \cdot \left[1 - P_b^d + P_b^d - P_b^{2d}\right] \\
&= \epsilon_b \cdot \left[1 - P_b^{2d}\right].
\end{aligned}
$$

Next, we simplify the match terms by using the definition of α_f which follows from Proposition 7 in a manner similar to the definition of α_b:

$$\begin{aligned}&\min_{\pi_{(c+d,h)}} \mathbb{E}_G\left[\mathcal{L}(\pi_{(c+d,h)},\pi^*)\Big|C\right]\\ &= \min_{\pi_{(c,h+d)}} \mathbb{E}_G\left[\mathcal{L}(\pi_{(c,h+d)},\pi^*)|C\right]-\alpha_f. && \text{(Proposition 7)}\end{aligned}$$

We substitute these terms back in to get the desired bound. □

C.4 Discussion on Assumption 1

We assumed that $c+h<k$ so that the larger action chunk model can condition on more states from the distant past that are temporally correlated with a_t. In the case where $c+h\geq k$, the larger action chunk model will not get any advantage (so, $\alpha=0$) and can only suffer from not having observed the recent past states. However, this is a very unlikely scenario because we expect human demonstrators to have a large memory horizon *i.e.*k is expected to be large.

C.5 Proof of Corollary 2 and Corollary 3

Now, we prove Corollary 2 as a direct consequence of the Consistency-Reactivity Inequalities. In a near-deterministic environment, P_f is close to 1. This is because, conditioned on the state and action at time $t'-1$, we can confidently infer the state visited at time t' as the environment lacks noise.

Corollary 2 (Restated) In a near-deterministic environment, if a_t is temporally dependent on at least one state in $\{s_{t-h-c-d:t-h-c-1}\}$ and ϵ_f is finite,

$$\min_{\pi_{h+d}} \mathbb{E}_G\left[\mathcal{L}(\pi_{h+d},\pi^*)|C\right] < \min_{\pi_h} \mathbb{E}_G\left[\mathcal{L}(\pi_h,\pi^*)|C\right]$$

Proof. This follows from Proposition 1 by taking $P_f\approx 1$, ϵ_f not large and $\alpha_b>0$ (since a_t is temporally dependent on at least one state in $\{s_{t-h-c-d:t-h-c-1}\}$ and the states in $\mathcal{T}_b$ cannot be deterministically inferred from the context of π_h). □

Lastly, we prove Corollary 3. We assume that, in a highly stochastic environment, P_b is small. This is because, for any time step t', the agent could reach $s_{t'+1}$ from many different states at time t' due to the noise in the environment.

Corollary 3 (Restated) In a highly stochastic environment, if temporal dependency decreases over the number of time steps *i.e.* $\alpha_f>\epsilon_b$, then

$$\min_{\pi_{h+d}} \mathbb{E}_G\left[\mathcal{L}(\pi_{h+d},\pi^*)|C\right] > \min_{\pi_h} \mathbb{E}_G\left[\mathcal{L}(\pi_h,\pi^*)|C\right]$$

Proof. Starting from Proposition 1, with P_b small (since the environment is highly stochastic), we get

$$\min_{\pi^t_{(c,h)}} \mathbb{E}_G\left[\mathcal{L}(\pi^t_{(c,h)}-\pi_E)|C\right] \leq \min_{\pi^t_{(c,h+d)}} \mathbb{E}_G\left[\mathcal{L}(\pi^t_{(c,h+d)}-\pi^t_E)|C\right]-\alpha_f+\epsilon_b.$$

Since temporal dependency reduces as number of steps grow, a_t is more temporally dependent on the recent past states than on the distant past, so $\alpha_f>\epsilon_b$. With this observation, we get:

$$\min_{\pi^t_{(c,h)}} \mathbb{E}_G\left[\mathcal{L}(\pi^t_{(c,h)}-\pi_E)|C\right] < \min_{\pi^t_{(c,h+d)}} \mathbb{E}_G\left[\mathcal{L}(\pi^t_{(c,h+d)}-\pi^t_E)|C\right].$$

□

C.6 Closed-Loop versus Open-Loop in Highly Stochastic and Near-Deterministic Environments

The Consistency-Reactivity Inequalities allow us to make an even stronger statement when we compare strictly closed-loop policies with open-loop ones. Consider the same set-up as before with $h=0$. Thus, $\pi_{(c,0)}$ represents a closed-loop policy whereas $\pi_{(c,d)}$ represents an open-loop one. We can compare these policies' divergences with the expert across the entire trajectory in the limiting cases of the environment stochasticity.

Corollary 9. *In a highly stochastic environment, if temporal dependency decreases such that $\alpha_f > \epsilon_b$ at all time steps, then divergence between the closed-loop policy over the full trajectory is lower than that between the open-loop policy and the expert. In a near deterministic environment, if there is at least one time step t such that a_t depends on some state in $s_{t-d-c:t-c-1}$, then the divergence between the closed-loop policy over the full trajectory is greater than that between the open-loop policy and the expert.*

Proof. At any arbitrary time step t, the chunks of the two policies may be aligned in one of two ways:

Case 1: $\pi_{(c,1)}$ is executing a_t as the first action in its action chunk and $\pi_{(c,1+d)}$ is also executing a_t as the first action in its action chunk.

Case 2: $\pi_{(c,1)}$ is executing a_t as the first action in its action chunk and $\pi_{(c,1+d)}$ is also executing a_t as the k-th action, where $k \in (1, 1+d]$ in its action chunk.

Using the Consistency-Reactivity Inequalities, in Case 1, both policies have equal divergence.

However, in case 2, using Corollary 3, we know that the closed-loop policy will outperform the open-loop one in the first setting of the statement and open-loop will outperform in the second. From this, we can conclude the divergence across the full trajectory. □