Probabilistic Inference in Language Models via Twisted Sequential Monte Carlo

Stephen Zhao [1 2 *] **Rob Brekelmans** [2 *] **Alireza Makhzani** [1 2 **] **Roger Grosse** [1 2 **]

Abstract

Numerous capability and safety techniques of Large Language Models (LLMs), including RLHF, automated red-teaming, prompt engineering, and infilling, can be cast as sampling from an unnormalized target distribution defined by a given reward or potential function over the full sequence. In this work, we leverage the rich toolkit of Sequential Monte Carlo (SMC) for these probabilistic inference problems. In particular, we use learned *twist functions* to estimate the expected future value of the potential at each timestep, which enables us to focus inference-time computation on promising partial sequences. We propose a novel contrastive method for learning the twist functions, and establish connections with the rich literature of soft reinforcement learning. As a complementary application of our twisted SMC framework, we present methods for evaluating the accuracy of language model inference techniques using novel *bidirectional* SMC bounds on the log partition function. These bounds can be used to estimate the KL divergence between the inference and target distributions in both directions. We apply our inference evaluation techniques to show that twisted SMC is effective for sampling undesirable outputs from a pretrained model (a useful component of harmlessness training and automated red-teaming), generating reviews with varied sentiment, and performing infilling tasks.

1. Introduction

A wide range of language model learning and inference tasks can be viewed as steering a model's generations to satisfy a specified property. In particular, traditional reinforcement learning from human feedback (RLHF) pipelines (Ziegler et al., 2019; Stiennon et al., 2020; Ouyang et al., 2022; Bai et al., 2022; Rafailov et al., 2023) prioritize responses that score highly according to a terminal reward function reflecting human feedback (Korbak et al., 2022b). Red-teaming techniques such as prompt-engineering and infilling may seek target outputs with low reward or (high probability of) undesirable responses (Zou et al., 2023; Perez et al., 2022). In reasoning tasks, we may seek to target outputs which are likely to be deemed valid by a 'verifier' (Cobbe et al., 2021; Anil et al., 2021; Dohan et al., 2022; Hu et al., 2023). Specific properties of generated responses might also be enforced (Khalifa et al., 2020; Yang & Klein, 2021; Lew et al., 2023).

We view the above tasks as instances of *probabilistic inference*: sampling from a target unnormalized density and estimating its intractable (log) normalization constant. Consider a pretrained base model $p_0(\mathbf{s}_{1:T}|\mathbf{s}_0)$ which generates responses $\mathbf{s}_{1:T}$ of maximum length T based on a variable-length prompt $\mathbf{s}_0$. We consider defining the target distribution of interest using the base model modulated by a potential function $\phi(\mathbf{s}_{1:T})$ which evaluates full sequences,

$$\sigma(\mathbf{s}_{1:T}|\mathbf{s}_0) := \frac{1}{\mathcal{Z}_\sigma(\mathbf{s}_0)} p_0(\mathbf{s}_{1:T}|\mathbf{s}_0)\phi(\mathbf{s}_{1:T}), \tag{1}$$

$$\text{where } \mathcal{Z}_\sigma(\mathbf{s}_0) := \sum_{\mathbf{s}_{1:T}} \tilde{\sigma}(\mathbf{s}_{1:T}|\mathbf{s}_0) = \sum_{\mathbf{s}_{1:T}} p_0(\mathbf{s}_{1:T}|\mathbf{s}_0)\phi(\mathbf{s}_{1:T}),$$

where $\tilde{\sigma}(\mathbf{s}_{1:T}|\mathbf{s}_0)$ denotes the unnormalized density. We refer to $\mathcal{Z}_\sigma(\mathbf{s}_0)$ as the normalization constant or partition function, which is intractable due to the summation over $\mathbf{s}_{1:T}$. We drop dependence on $\mathbf{s}_0$ to avoid clutter, but note that each prompt induces a different partition function. In the context of the aforementioned applications, $\phi(\mathbf{s}_{1:T})$ may be derived from a human preference model (for RLHF), an indication of bad behavior (for automated red-teaming), or a verifier's prediction of correctness (for reasoning tasks). We refer to Table 4 or Korbak et al. (2022b); Dohan et al. (2022); Phan et al. (2023); Hu et al. (2023) for further examples and discussion of probabilistic inference in language models.

Twisted Sequential Monte Carlo in Language Models In this work, we leverage tools from (twisted) Sequential Monte Carlo (SMC) (Doucet et al., 2001; Del Moral et al., 2006; Briers et al., 2010; Chopin et al., 2020)

[*] Joint first authorship, [**] Joint senior authorship. [1]University of Toronto [2]Vector Institute. Correspondence to: {stephenzhao, makhzani, rgrosse} @cs.toronto.edu, rob.brekelmans@vectorinstitute.ai.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

to perform and evaluate inference in the language modeling setting (Sec. 3). A particular challenge in sampling from Eq. (1) is that the target distribution $\sigma(\mathbf{s}_{1:T})$ is non-causal. In order to sample tokens sequentially, one needs to infer the marginal distribution $\sigma(\mathbf{s}_{1:t}) = \sum_{\mathbf{s}_{t+1:T}} \sigma(\mathbf{s}_{1:T}) \propto \sum_{\mathbf{s}_{t+1:T}} p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t})\phi(\mathbf{s}_{1:T})$, which involves an intractable marginalization. To address this problem, we propose to learn *twist functions* $\psi_t(\mathbf{s}_{1:t})$ which modulate the base model such that $p_0(\mathbf{s}_{1:t})\psi_t(\mathbf{s}_{1:t})$ matches the target marginals $\sigma(\mathbf{s}_{1:t})$, up to normalization. The twist functions can be used to focus each step of language model generation on promising partial sequences.

Evaluating Inference in Language Modeling Sampling from the target distribution is closely intertwined with bounding the log partition function. Similarly to variational inference or traditional RLHF objectives (Korbak et al., 2022b), SMC algorithms yield lower bounds on $\log \mathcal{Z}_\sigma$, where tighter bounds typically coincide with more accurate target sampling. However, *upper* bounds may often be obtained when an exact target sample is available (Grosse et al., 2015; 2016; Brekelmans et al., 2021). The difference between upper and lower bounds on $\log \mathcal{Z}_\sigma$ in fact yields an upper bound on the symmetrized KL divergence between the distribution of inference samples and the target distribution (Grosse et al., 2016). For these reasons, we argue in Sec. 5 that log partition function estimates are a powerful tool for evaluating language model inference techniques.

Contributions Our probabilistic inference perspective leads to the following contributions:

- *Twisted Sequential Monte Carlo for Language Modeling*: We view *twisted* SMC as a general framework for sampling and evaluation of language models. While twisted SMC is well-known and Lew et al. (2023) consider SMC with fixed, few-step-ahead target information in the language modeling setting, we propose to *learn* intermediate twist functions for target distributions defined by terminal potential only.
- *Contrastive Twist Learning*: We develop probabilistic methods for learning intermediate twist functions, presenting a novel *contrastive twist learning* (CTL) method inspired by energy-based modeling and density ratio estimation in Sec. 4.1. Further, we adapt existing twisted SMC methods (Lawson et al., 2018; 2022; Lioutas et al., 2022) to the language modeling setting, and highlight connections with inference techniques inspired by (soft) reinforcement learning (RL).
- *Evaluating Inference in Language Models*: Finally, we demonstrate that twisted SMC provides a rich set of tools for evaluating language model fine-tuning or controlled generation techniques. We propose a novel SMC upper bound on $\log \mathcal{Z}_\sigma$ which is applicable when an exact target sample is available and may be of independent interest. We apply these bounds to evaluate inference quality by measuring the KL divergence to the target $\sigma(\mathbf{s}_{1:T})$ in *both* directions, which can be used to diagnose mode-dropping behavior of methods such as proximal policy optimization (PPO) (Schulman et al., 2017) which optimize a mode-seeking divergence.

We describe background on importance sampling and SMC in Sec. 2, before presenting our framework for twisted SMC in the language model setting in Sec. 3. We propose methods to learn the twist functions in Sec. 4 and methods to evaluate inference in Sec. 5. Our experiments in Sec. 7 showcase the ability of twisted SMC to improve controlled generation and lend insights into inference quality in existing methods.

2. Background

Suppose we are given access to an unnormalized density $\tilde{\sigma}(\mathbf{s}_{1:T})$ which can be efficiently evaluated. We focus on estimation of the partition function or normalization constant $\mathcal{Z}_\sigma := \sum_{\mathbf{s}_{1:T}} \tilde{\sigma}(\mathbf{s}_{1:T})$, since unbiased estimators with low variance yield approximate sampling techniques which closely approximate the target distribution (Finke, 2015; Maddison et al., 2017). We review simple importance sampling (SIS) and SMC techniques in this section.

2.1. Simple Importance Sampling

Simple importance sampling (SIS) provides an unbiased estimator of $\mathcal{Z}_\sigma$ by calculating importance weights for any normalized proposal distribution $q(\mathbf{s}_{1:T})$,

$$w(\mathbf{s}^i_{1:T}) \coloneqq \frac{\tilde{\sigma}(\mathbf{s}^i_{1:T})}{q(\mathbf{s}^i_{1:T})}\,, \tag{2}$$

which is unbiased since $\mathcal{Z}_\sigma = \mathbb{E}_{q(\mathbf{s}_{1:T})}[w(\mathbf{s}_{1:T})]$. The importance weights also yield an an unbiased K-sample estimator of the partition function,

$$\hat{\mathcal{Z}}^{\text{SIS}}_\sigma \coloneqq \frac{1}{K}\sum_{i=1}^K w(\mathbf{s}^i_{1:T})\,, \qquad \mathbf{s}^i_{1:T} \sim q(\mathbf{s}_{1:T})\,. \tag{3}$$

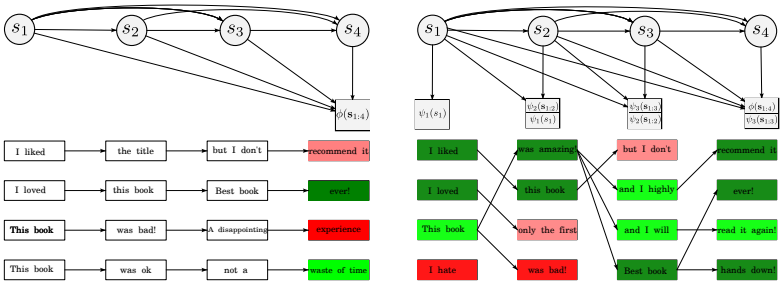
By normalizing the weights in Eq. (2) over K samples from $q(\mathbf{s}_{1:T})$, we can obtain (biased) estimators of expectations under $\sigma(\mathbf{s}_{1:T})$,

$$\mathbb{E}_{\sigma(\mathbf{s}_{1:T})}\big[f(\mathbf{s}_{1:T})\big] \approx \sum_{k=1}^K \frac{w(\mathbf{s}^k_{1:T})}{\sum_{j=1}^K w(\mathbf{s}^j_{1:T})} f(\mathbf{s}^k_{1:T}) \tag{4}$$

or select an approximate target sample $\mathbf{s}^\sigma_{1:T}$ from a categorical distribution with the self-normalized importance weights

$$\mathbf{s}^\sigma_{1:T} \leftarrow \mathbf{s}^\omega_{1:T}, \qquad \omega \sim \mathsf{cat}\left(\left\{\frac{w(\mathbf{s}^i_{1:T})}{\sum_{j=1}^K w(\mathbf{s}^j_{1:T})}\right\}_{i=1}^K\right). \tag{5}$$

The quality of the approximations in Eq. (3)-(5) depends crucially on how well the proposal $q(\mathbf{s}_{1:T})$ (which may be



(a) Simple Importance Sampling (b) Twisted SMC

Figure 2: Illustrative example of SIS and (Twisted) SMC for sampling book reviews conditioned on positive sentiment $\phi(\mathbf{s}_{1:T})$. SIS only performs resampling after observing the entire sequence, while SMC can kill or clone partial sequences $\mathbf{s}_{1:t}$ based on incremental importance weights induced by twist functions $\psi_t(\mathbf{s}_{1:t})$. Green/red indicate high/low importance weights at each intermediate step of SMC, or at the final step of SIS. For SMC with the base model proposal p_0 and the optimal twists, the incremental weights ψ_t^*/ψ_{t-1}^* (Alg. 1 or Eq. (6)) are directly correlated with sentiment.

Algorithm 1 (Twisted) SMC Sampling (q_{SMC})

SMC-PROPOSAL $(p_0, q, \{\psi_t\}_{t=1}^{T-1}, \phi, K)$:
for $t = 1, ..., T$ **do**
  for $k = 1, ..., K$ **do**
    Sample $s_t^k \sim q(s_t \mid \mathbf{s}_{1:t-1}^k)$
    $\mathbf{s}_{1:t}^k \leftarrow \texttt{concat}(\mathbf{s}_{1:t-1}^k, s_t^k)$
    if $t < T$ **then**
      $w_t^k \leftarrow \frac{p_0(s_t^k \mid \mathbf{s}_{1:t-1}^k)}{q(s_t^k \mid \mathbf{s}_{1:t-1}^k)} \frac{\psi_t(\mathbf{s}_{1:t}^k)}{\psi_{t-1}(\mathbf{s}_{1:t-1}^k)}$
    else
      $w_t^k \leftarrow \frac{p_0(s_t^k \mid \mathbf{s}_{1:t-1}^k)}{q(s_t^k \mid \mathbf{s}_{1:t-1}^k)} \frac{\phi(\mathbf{s}_{1:t}^k)}{\psi_{t-1}(\mathbf{s}_{1:t-1}^k)}$
    end if
  end for
  if $t < T$ **then**
    $\bar{\mathbf{s}}_{1:t}^{1:K} \leftarrow \mathbf{s}_{1:t}^{1:K}.\texttt{copy}()$
    for $k = 1, ..., K$ **do**
      $\omega_t^k \sim \texttt{cat}\left(\left\{\frac{w_t^i}{\sum_{j=1}^K w_t^j}\right\}_{i=1}^K\right)$
      $\mathbf{s}_{1:t}^k \leftarrow \bar{\mathbf{s}}_{1:t}^{\omega_t^k}$
    end for
  end if
end for
return $\{\mathbf{s}_{1:T}^k, w_T^k\}_{k=1}^K$
  $\hat{\mathcal{Z}}_\sigma^{\text{SMC}} = \prod_{t=1}^T \frac{1}{K} \sum_{k=1}^K w_t^k$

learned, Sec. 3.2) matches the target $\sigma(\mathbf{s}_{1:T})$. While we discuss evaluation methods in Sec. 5, note that if inference is exact (i.e., $q(\mathbf{s}_{1:T}) = \sigma(\mathbf{s}_{1:T})$), then the variance of the importance weights is zero, as $w(\mathbf{s}_{1:T}) = \mathcal{Z}_\sigma$ for all $\mathbf{s}_{1:T}$.

2.2. Sequential Monte Carlo

SMC improves inference by decomposing it into easier subproblems involving a set of unnormalized intermediate target distributions $\{\tilde{\pi}_t(\mathbf{s}_{1:t})\}_{t=1}^T$, where $\tilde{\pi}_t$ is the unnormalized density of $\pi_t = \tilde{\pi}_t/\mathcal{Z}_t$. A key observation is that as long as $\tilde{\pi}_T(\mathbf{s}_{1:T}) = \tilde{\sigma}(\mathbf{s}_{1:T})$, we obtain an unbiased estimate of the partition function $\mathcal{Z}_T = \mathcal{Z}_\sigma$, regardless of the intermediate distributions π_t and proposal $q(s_t|\mathbf{s}_{1:t-1})$.

We begin by defining the *incremental* importance weights

$$w_t(\mathbf{s}_{1:t}) \coloneqq \frac{\tilde{\pi}_t(\mathbf{s}_{1:t})}{\tilde{\pi}_{t-1}(\mathbf{s}_{1:t-1}) q(s_t|\mathbf{s}_{1:t-1})}. \tag{6}$$

SMC maintains a set of K partial sequences, by first sampling from the proposal $q(s_t^k|\mathbf{s}_{1:t-1}^k)$ in each index k. Optional resampling steps may be performed to clone sequences with high incremental importance weights using

$$\mathbf{s}_{1:t}^k \leftarrow \mathbf{s}_{1:t}^{\omega_t^k}, \qquad \omega_t^k \sim \texttt{cat}\left(\left\{\frac{w_t(\mathbf{s}_{1:t}^i)}{\sum_{j=1}^K w_t(\mathbf{s}_{1:t}^j)}\right\}_{i=1}^K\right), \tag{7}$$

similarly to Eq. (5). Since resampling is performed *with* replacement, sequences with high weights may be cloned multiple times. The resulting $\mathbf{s}_{1:t}^{\omega_t^k}$ are used as prefixes for the next step of proposal sampling in index k (see Alg. 1).

We can show that SMC yields an unbiased estimator $\hat{\mathcal{Z}}_\sigma^{\text{SMC}}$ of the normalization constant $\mathcal{Z}_\sigma$, by considering the extended state space $\boldsymbol{S} \coloneqq \{s_t^k, \omega_t^k\}_{k,t=1}^{K,T}$ of token and index random variables from the sampling procedure $\boldsymbol{S} \sim q_{\text{SMC}}(\boldsymbol{S})$ in Alg. 1. Assuming resampling at every step, [1]

$$\mathcal{Z}_\sigma = \mathbb{E}\left[\hat{\mathcal{Z}}_\sigma^{\text{SMC}}\right] = \mathbb{E}_{q_{\text{SMC}}(\boldsymbol{S})}\left[\prod_{t=1}^T \frac{1}{K} \sum_{k=1}^K w_t(\mathbf{s}_{1:t}^k)\right]. \tag{8}$$

To show $\hat{\mathcal{Z}}_\sigma^{\text{SMC}}$ is unbiased, we view Eq. (8) as performing simple importance sampling $\mathcal{Z}_\sigma = \mathbb{E}_{q_{\text{SMC}}(\boldsymbol{S})}\left[\frac{\tilde{\sigma}_{\text{SMC}}(\boldsymbol{S})}{q_{\text{SMC}}(\boldsymbol{S})}\right]$ in the extended state space, for appropriate definitions of $\sigma_{\text{SMC}}(\boldsymbol{S})$ and $q_{\text{SMC}}(\boldsymbol{S})$ in App. F or (Andrieu et al., 2010; Maddison et al., 2017). Intuitively, we may view the average incremental importance weights at each step as estimating the ratio $\mathcal{Z}_t/\mathcal{Z}_{t-1} \approx \frac{1}{K}\sum_{k=1}^K w_t(\mathbf{s}_{1:t}^k)$. Eq. (8) composes intermediate partition function ratio estimators to obtain an estimate of the final $\mathcal{Z}_T = \mathcal{Z}_\sigma = \prod_{t=1}^T \mathcal{Z}_t/\mathcal{Z}_{t-1}$, with $\mathcal{Z}_0 = 1$.

With no resampling, SMC reduces to SIS with target $\sigma(\mathbf{s}_{1:T}) = \pi_T(\mathbf{s}_{1:T})$ and proposal $q(\mathbf{s}_{1:T})$. Using the final-step SMC weights, we may estimate expectations or draw approximate samples $\mathbf{s}_{1:T}^\sigma$ as in Eq. (4)-(5).

Fig. 2 illustrates the key advantage of SMC resampling over SIS. While a suboptimal $q(\mathbf{s}_{1:T})$ may produce sequences with low probability under the target $\sigma(\mathbf{s}_{1:T})$, SMC resampling with well-chosen intermediate targets π_t clones the most promising partial sequences $\mathbf{s}_{1:t}$ at step t. Since later sampling proceeds from these prefixes, we expect to obtain final sequences which better cover the high-probability regions of the target distribution. We discuss techniques to evaluate the quality of SMC or SIS sampling in Sec. 5.

[1] The decision to resample may be based on an adaptive condition such as Effective Sample Size (ESS) (Chopin et al., 2020). For $R \leq T$, let $\{t_r\}_{r=1}^R$ index times where resampling occurs and fix $t_0 = 0$ and $t_R = T$. The estimator becomes $\hat{\mathcal{Z}}_\sigma^{\text{SMC}} = \prod_{r=1}^R \frac{1}{K} \sum_{i=1}^K \left(\prod_{t=t_{r-1}+1}^{t_r} w_t(\mathbf{s}_{1:t}^i)\right)$, and the final-step weights used in Eq. (4) or (5) are $\prod_{t=t_{R-1}+1}^T w_t(\mathbf{s}_{1:t}^i)$.

3. Twisted Sequential Monte Carlo for Language Modeling

A key design choice in the SMC procedure above is the intermediate targets $\{\pi_t\}_{t=1}^{T-1}$, where we assume $\pi_T(\mathbf{s}_{1:T}) = \sigma(\mathbf{s}_{1:T})$ is always the target distribution. In state-space models with observation likelihoods or environments with intermediate rewards, *filtering* SMC considers target information collected from times $\tau \leq t$ to define π_t. (Chopin et al., 2020). Previous work on SMC for language models (Lew et al., 2023) has considered per-token or few-step-ahead statistics to define tractable intermediate π_t. However, we are often interested in target distributions which are determined by a terminal potential $\phi(\mathbf{s}_{1:T})$ only, as in Eq. (1).

In such settings, *twisted* SMC methods (Briers et al., 2010; Whiteley & Lee, 2014; Lawson et al., 2022) consider the *full* target information (until time T) to define $\{\pi_t\}_{t=1}^{T-1}$. In other words, our desired intermediate targets are the true marginals $\sigma(\mathbf{s}_{1:t})$ of the target distribution. Intuitively, note that in order to exactly sample $\mathbf{s}_{1:T} \sim \sigma(\mathbf{s}_{1:T})$, we need to ensure partial sequences are distributed according to the intermediate marginals $\mathbf{s}_{1:t} \sim \sigma(\mathbf{s}_{1:t})$. In Sec. 3.1, we will represent the intermediate targets $\{\pi_t\}_{t=1}^{T-1}$ using *twist* functions $\psi_t \colon \mathbf{s}_{1:t} \to \mathbb{R}$ which modulate the base model to (approximately) match the target marginals, thereby summarizing future information relevant to sampling at time t.

3.1. Twist Functions

We represent the intermediate target distributions $\{\pi_t\}_{t=1}^{T-1}$ for SMC sampling using the following general form.

Definition 3.1 (Twisted (Intermediate) Targets). *Using approximate twist functions $\{\psi_t\}_{t=1}^{T-1}$ and the final target ϕ, we define the twisted intermediate target distributions*

$$\pi_t(\mathbf{s}_{1:t}) = \begin{cases} \frac{1}{Z_t^\psi}\, p_0(\mathbf{s}_{1:t})\, \psi_t(\mathbf{s}_{1:t}) & t \neq T \\ \frac{1}{Z_\sigma}\, p_0(\mathbf{s}_{1:T})\, \phi(\mathbf{s}_{1:T}) & t = T \end{cases} \tag{9}$$

For an arbitrary proposal q and the unnormalized targets in Eq. (9), the incremental importance weights are given by

$$w_t(\mathbf{s}_{1:t}) = \frac{p_0(s_t|\mathbf{s}_{1:t-1})}{q(s_t|\mathbf{s}_{1:t-1})} \frac{\psi_t(\mathbf{s}_{1:t})}{\psi_{t-1}(\mathbf{s}_{1:t-1})}. \tag{10}$$

While uninformed twist functions ψ_t may result in $\pi_t(\mathbf{s}_{1:t})$ which are no closer to the target marginal $\sigma(\mathbf{s}_{1:t})$ than the base model $p_0(\mathbf{s}_{1:t})$ (for example, in early stages of learning), the crucial fact is that our final target distribution in Eq. (9) reflects the target potential $\phi(\mathbf{s}_{1:T})$. As in Sec. 2.2, this ensures that, regardless of the intermediate twists, our resulting importance sampling estimators will be unbiased.

Finally, the optimal twists $\psi_t^*(\mathbf{s}_{1:t})$ recover the intermediate marginals $\pi_t^*(\mathbf{s}_{1:t}) = \sigma(\mathbf{s}_{1:t})$ of the target distribution. We state the sense in which π_t^* and ψ_t^* are optimal in App. A.1, and prove the following proposition in App. B Prop. B.1.

Proposition 3.2 (Optimal Twists). *For a given target distribution $\sigma(\mathbf{s}_{1:T})$ in Eq. (1), the optimal twist functions $\psi_t^*(\mathbf{s}_{1:t})$ (in regions where $p_0(\mathbf{s}_{1:t}) > 0$) correspond to*

$$\pi_t^*(\mathbf{s}_{1:t}) = \sigma(\mathbf{s}_{1:t}) = \frac{1}{\mathcal{Z}_t^{\psi^*}}\, p_0(\mathbf{s}_{1:t})\, \psi_t^*(\mathbf{s}_{1:t}). \tag{11}$$

Up to a constant independent of $\mathbf{s}_{1:t}$, the optimal twists are

$$\psi_t^*(\mathbf{s}_{1:t}) \propto \sum_{\mathbf{s}_{t+1:T}} p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t})\phi(\mathbf{s}_{1:T}). \tag{12}$$

and satisfy the recursion

$$\psi_t^*(\mathbf{s}_{1:t}) \propto \sum_{s_{t+1}} p_0(s_{t+1} \,|\, \mathbf{s}_{1:t})\psi_t^*(\mathbf{s}_{1:t+1}). \tag{13}$$

Since the optimal twist functions are unavailable due to the need to marginalize over future timesteps, we consider learning approximate twist functions using methods in Sec. 4.

3.2. Proposal Distribution

For a given set of targets $\{\pi_t\}_{t=1}^{T}$, the importance weights in Eq. (10) depend crucially on the choice of proposal.

Base Model as Proposal The most straightforward choice of proposal is the base pre-trained model, $q = p_0$. While we demonstrate in Sec. 7 that SMC resampling with learned twists and the base model proposal can closely approximate the target distribution, this may require large K. We can achieve greater efficiency using better choices of proposal.

Twist-Induced Proposal For given targets $\{\pi_t\}_{t=1}^{T}$, the optimal proposal minimizes the variance of the importance weights (App. A.1). In the language model setting with a terminal potential only, we will in fact be able to sample from the optimal proposal for the one-step importance weights.

Proposition 3.3. (Twist-Induced Proposal). *For a given set of intermediate twisted targets $\pi_t(\mathbf{s}_{1:t})$ in Eq. (9), the proposal which minimizes the variance of the one-step incremental importance weights w_t is given by*

$$\begin{aligned} q_t^\pi(s_t|\mathbf{s}_{1:t-1}) &\propto \frac{\pi_t(\mathbf{s}_{1:t})}{\pi_{t-1}(\mathbf{s}_{1:t-1})} \\ &= \frac{1}{Z_t^\pi(\mathbf{s}_{1:t-1})} p_0(s_t|\mathbf{s}_{1:t-1})\psi_t(\mathbf{s}_{1:t}). \end{aligned} \tag{14}$$

See proof in App. A.2. For $t < T$, we can construct a parameterization of $\psi_t(\mathbf{s}_{1:t})$ such that the proposal is tractable to sample in transformer architectures, where the normalization $Z_t^\pi(\mathbf{s}_{1:t-1}) = \sum_{s_t} p_0(s_t|\mathbf{s}_{1:t-1})\psi_t(\mathbf{s}_{1:t})$ sums over the discrete vocabulary of next tokens $s_t \in \mathcal{V}$. However, at the final step, note that $\phi(\mathbf{s}_{1:T})$ may require calls to a different neural network such as a reward model or classifier.

We thus consider an approximate $\psi_T(\mathbf{s}_{1:T}) \approx \phi(\mathbf{s}_{1:T})$ for the proposal $q_T(s_T|\mathbf{s}_{1:T-1}) \propto p_0(s_T|\mathbf{s}_{1:T-1})\psi_T(\mathbf{s}_{1:T})$ at

the final timestep. With slight abuse of notation, we let $q^\pi(\mathbf{s}_{1:T})$ denote this tractable proposal over full sequences,

$$q^\pi(\mathbf{s}_{1:T}) \coloneqq \Big(\prod_{t=1}^{T-1} q_t^\pi(s_t|\mathbf{s}_{1:t-1})\Big)\, q_T(s_T|\mathbf{s}_{1:T-1})\,. \quad (15)$$

Using this proposal, the incremental weights become

$$w_t(\mathbf{s}_{1:t}) = \begin{cases} \frac{\sum_{s_t} p_0(s_t|\mathbf{s}_{1:t-1})\psi_t(\mathbf{s}_{1:t})}{\psi_{t-1}(\mathbf{s}_{1:t-1})} & t < T \\ \frac{\sum_{s_T} p_0(s_T|\mathbf{s}_{1:T-1})\psi_T(\mathbf{s}_{1:T})}{\psi_{T-1}(\mathbf{s}_{1:T-1})}\frac{\phi(\mathbf{s}_{1:T})}{\psi_T(\mathbf{s}_{1:T})} & t = T \end{cases}, \quad (16)$$

which are independent of s_t for $t < T$.

Variational Proposal As noted in Sec. 2.1, SMC with no resampling steps reduces to SIS with the full target distribution $\sigma(\mathbf{s}_{1:T})$. Policy gradient methods (Schulman et al., 2017; Parshakova et al., 2019; Korbak et al., 2022a; Go et al., 2023) which directly learn a tractable approximation $q(\mathbf{s}_{1:T})$ to the target distribution may thus be viewed as a particularly simple instance of SMC, or inference more generally (see Korbak et al. (2022b)). We may also evaluate these inference methods using our proposed tools in Sec. 5. See Table 5 and App. E for detailed losses and discussion.

Finally, we might also learn a separate proposal q alongside the twisting targets $\{\pi_t\}_{t=1}^{T-1}$. This may be useful to approximate the variance-minimizing proposal for multi-step or adaptive resampling beyond the tractable optimal one-step proposal in Prop. 3.3 (see Prop. A.5). We discuss training losses based on multi-step importance weights in App. C.1.

3.3. Conditional Target Distributions

More generally, we may consider *conditional* target distributions, obtained by conditioning on an observation random variable o_T. This mirrors the standard setting of SMC in state-space models (Doucet et al., 2001; Briers et al., 2010; Gu et al., 2015; Heng et al., 2020; Lawson et al., 2022).

Defining $\phi(\mathbf{s}_{1:T}, o_T) = \sigma(o_T|\mathbf{s}_{1:T})$ as a probabilistic model of o_T, our target distribution is the posterior $\sigma(\mathbf{s}_{1:T}|o_T)$,

$$\sigma(\mathbf{s}_{1:T}|o_T) = \frac{1}{\mathcal{Z}_\sigma(o_T)} p_0(\mathbf{s}_{1:T})\sigma(o_T|\mathbf{s}_{1:T})\,, \quad (17)$$

where the partition function $\mathcal{Z}_\sigma(o_T) = \sigma(o_T) = \sum_{\mathbf{s}_{1:T}} p_0(\mathbf{s}_{1:T})\sigma(o_T|\mathbf{s}_{1:T})$ is the marginal of the given o_T.

Using Prop. 3.2, the optimal twists matching the marginals $\sigma(\mathbf{s}_{1:t}|o_T)$, are conditional likelihoods of o_T given $\mathbf{s}_{1:t}$,

$$\begin{aligned}\psi_t^*(\mathbf{s}_{1:t}, o_T) &\propto \sum_{\mathbf{s}_{t+1:T}} p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t})\phi(\mathbf{s}_{1:T}, o_T) \\ &= \sigma(o_T|\mathbf{s}_{1:t})\,,\end{aligned} \quad (18)$$

since $\sigma(o_T|\mathbf{s}_{1:t}) = \sum_{\mathbf{s}_{t+1:T}} \sigma(o_T, \mathbf{s}_{t+1:T}|\mathbf{s}_{1:t})$. We proceed to construct intermediate target distributions and proposals in the previous sections, where $\psi_t(\mathbf{s}_{1:t}, o_T)$ and even $q_t(s_t|\mathbf{s}_{1:t-1}, o_T)$ may condition on a particular value of o_T. To recover the unconditional setting, we fix a binary observational variable $\sigma(o_T = 1|\mathbf{s}_{1:T}) \coloneqq \phi(\mathbf{s}_{1:T})$ (Levine, 2018) and omit explicit conditioning, showing that conditional twist learning generalizes our previous exposition (see App. B.2 for detailed discussion).

Exact Target Sampling on Simulated Data Assuming $\sigma(o_T|\mathbf{s}_{1:T})$ is tractable to sample, we may obtain an exact sample from the target posterior for simulated o_T using ancestral sampling. In particular, we obtain a sample $\mathbf{s}_{1:T}, o_T \sim p_0(\mathbf{s}_{1:T})\sigma(o_T|\mathbf{s}_{1:T})$ from the joint distribution, which also factorizes as $\sigma(o_T, \mathbf{s}_{1:T}) = \sigma(o_T)\sigma(\mathbf{s}_{1:T}|o_T)$. Using the latter factorization, we may interpret $\mathbf{s}_{1:T}$ as an exact sample from the target posterior for the given o_T. We refer to this as the Bidirectional Monte Carlo (BDMC) trick (Grosse et al., 2015; 2016), and will use it to draw exact samples for training in Sec. 4.1.2 or evaluation in Sec. 5.

3.4. Connections with Reinforcement Learning

Twisted SMC shares close connections with (soft) reinforcement learning (Levine, 2018; Piché et al., 2018; Lawson et al., 2018; Heng et al., 2020; Lioutas et al., 2022), which we develop with detailed discussion in App. B.3 and App. D. Here, we briefly mention two distinct RL interpretations of the SMC twists in relation to the reward function.

Base Model Policy Evaluation Viewing the final potential $\phi(\mathbf{s}_{1:T})$ as the reward function, the optimality condition $\psi_t^*(\mathbf{s}_{1:t}) = \sum_{\mathbf{s}_{t+1:T}} p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t})\phi(\mathbf{s}_{1:T})$ in Eq. (12) corresponds to exact *policy evaluation* of the future reward under the *fixed* base model policy $p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t})$. Mudgal et al. (2023) adopt this perspective for controlled decoding, and refer to the twist functions as 'prefix scorers'.

Soft RL with KL Regularization Alternatively, we may consider the KL-regularized RL target distributions commonly used in language modeling (Levine, 2018; Korbak et al., 2022b) as a special case of our twisted SMC framework. For a regularization strength β, define the potential as

$$\phi(\mathbf{s}_{1:T}) = e^{\beta r(\mathbf{s}_{1:T})}. \quad (19)$$

In this case, intermediate twist functions in Def. 3.1 correspond to Q-values $\psi_t(\mathbf{s}_{1:t}) = e^{\beta Q(s_t, \mathbf{s}_{1:t-1})}$, and taking the log of both sides in the optimal twist condition Eq. (13) yields a soft Bellman recursion $Q^*(s_t, \mathbf{s}_{1:t-1}) = \frac{1}{\beta}\log \sum_{s_{t+1}} p_0(s_{t+1}|\mathbf{s}_{1:t}) e^{\beta Q^*(s_{t+1}, \mathbf{s}_{1:t})}$ with no intermediate reward (see App. B.3). From the soft RL perspective, the twists are analogous to a critic, while the proposal plays the role of an actor (Levine, 2018; Haarnoja et al., 2018).

4. Learning the Twist Functions

We next consider methods to learn twist functions $\psi_t^{\boldsymbol{\theta}}$ parameterized by neural networks, presenting a novel *contrastive twist learning* (CTL) approach in Sec. 4.1. We summarize twist learning methods from related work in Sec. 4.2.

4.1. Contrastive Twist Learning

To match our approximate $\pi_t^{\boldsymbol{\theta}}$ to the target marginals, we propose to minimize T separate KL divergences,

$$\min_{\boldsymbol{\theta}} \mathcal{L}_{\text{CTL}}(\boldsymbol{\theta}) \coloneqq \min_{\boldsymbol{\theta}} \sum_{t=1}^{T} D_{\text{KL}}\big(\sigma(\mathbf{s}_{1:t}) \,\big\|\, \pi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})\big) \,. \quad (20)$$

While other divergences could be used to learn $\pi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})$, we argue that the mass-covering behavior of Eq. (20) is a desirable property for twist learning. Since we separately match each $\sigma(\mathbf{s}_{1:t})$, our hope is that suboptimal learning in early timesteps does not lead to aggressive pruning of partial sequences that would achieve high final target likelihood.

Using Eq. (9), the negative gradient of Eq. (20) at each t is

$$\mathbb{E}_{\sigma(\mathbf{s}_{1:t})}\big[\nabla_{\boldsymbol{\theta}} \log \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})\big] - \mathbb{E}_{\pi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})}\big[\nabla_{\boldsymbol{\theta}} \log \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})\big], \quad (21)$$

which allows us to learn from exact target samples of $\sigma(\mathbf{s}_{1:t})$ in the first term when they are available.

Note the similarity of the objective in Eq. (20) and gradient in Eq. (21) to maximum likelihood training of energy-based models (EBM). Due to the form of the gradient update, we refer to this method as *contrastive twist learning* (CTL). We now proceed to describe approximate techniques for positive sampling (first term) and negative sampling (second term).

4.1.1. Approximate Negative Sampling

A common challenge in energy-based modeling is that the second term in Eq. (21) involves sampling from the target π_t with intractable normalization constant $\mathcal{Z}_t^{\psi}$. We proceed to estimate the expectation using SIS as in Eq. (4), using a proposal $q(\mathbf{s}_{1:t})$ such as the base model or the twist-induced proposal from Sec. 3.2. Note that SMC resampling with learned intermediate twist functions could also be used.

4.1.2. (Approximate) Positive Sampling

In contrast to traditional EBM settings, we do not necessarily have exact samples available from a 'data' distribution. We describe several settings related to availability of positive samples, which are explored in our experiments in Sec. 7.

Exact Target Samples If exact posterior samples are available, for example using the BDMC trick in Sec. 3.3, we may use them directly in the gradient update in Eq. (21).

Rejection Sampling Rejection sampling can yield exact target samples $\mathbf{s}_{1:T}^{\sigma}$ if an upper bound on the likelihood ratio $\frac{\tilde{\sigma}(\mathbf{s}_{1:T})}{q(\mathbf{s}_{1:T})} \leq M$ is known. When the target $\tilde{\sigma}(\mathbf{s}_{1:T})$ is defined by thresholding or an indicator function $p_0(\mathbf{s}_{1:T})\mathbb{I}(\mathbf{s}_{1:t} \in \mathcal{C})$ or joint distribution $p_0(\mathbf{s}_{1:T})\sigma(o_T|\mathbf{s}_{1:T})$, we can clearly take $M = 1$ for the base model proposal $p_0(\mathbf{s}_{1:T})$. If the base model yields posterior samples in reasonable time, we can obtain exact samples for training using rejection sampling, and use our twist learning procedures to greatly improve sampling efficiency at generation time.

While an improved proposal q may more efficiently draw samples meeting the target conditions, exact rejection sampling would require estimating the corresponding M. Approximate or quasi rejection sampling might be used in this case, as analysed in Eikema et al. (2022).

Approximate Positive Sampling using SIS or SMC In cases where exact samples are unavailable and rejection sampling is inefficient or inexact, we leverage SMC sampling with twist targets $\{\pi_t^{\boldsymbol{\theta}}\}_{t=1}^{T}$ and any proposal $q(\mathbf{s}_{1:T})$ to first draw a set of K full sequences $\mathbf{s}_{1:T}^{1:K}$. As in Eq. (4), we can use the normalized SMC weights since the last resampling step to estimate the expected gradient in the first term of Eq. (21). Without resampling, we recover SIS estimation.

While both our approximate positive and negative sampling for estimating the expectations in Eq. (21) rely on SMC or SIS weights (often with the same proposal), the crucial distinction is that weights for *positive* sampling are based on the *true target potential* $\phi(\mathbf{s}_{1:T})$ over *full* sequences.

Truncation to Partial Sequences For an exact positive sample, we use its truncation to a partial sequence of length t (which corresponds to a sample from the desired marginal σ_t) to perform the gradient update in Eq. (21). For approximate positive sampling, we use the same set of K final weights to estimate the expected gradient at each timestep.

4.2. Twist Learning Methods from Related Work

We briefly describe alternative approaches for twist learning, with detailed discussion in App. C and a summary of the loss functions for methods used in our experiments in Table 5.

Soft Q-Learning (RL) Enforcing the recursion in Eq. (13) using a squared error loss is analogous to soft Q-learning (see App. B.3, C.1.1), and has been used for twisted SMC in Lioutas et al. (2022). We interpret path consistency losses (Nachum et al., 2017), which were derived for soft RL and have been used for language modeling in Guo et al. (2021); Hu et al. (2023), from an importance sampling perspective in App. C.1, E.1. Mudgal et al. (2023) consider a similar squared Bellman error loss, but using the policy evaluation interpretation in Sec. 3.4 instead of a soft RL interpretation.

SIXO The SIXO loss proposed by Lawson et al. (2022) learns twist functions using a binary classification task to distinguish samples from the target marginal $\sigma(\mathbf{s}_{1:t}|o_T)$ and base model $p_0(\mathbf{s}_{1:t})$ at each step, which corresponds to noise contrastive estimation (Gutmann & Hyvärinen, 2010) for learning energy-based models. See App. C.3.

FUDGE Yang & Klein (2021) learn twists by constructing a binary classification task to instead learn the conditional likelihood $\sigma(o_T|\mathbf{s}_{1:t})$ (Eq. (18)). This may be viewed as enforcing the $T-t$ step optimality equation in Eq. (12) or Eq. (18), where rollouts should be obtained using the base

model $p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t})$ (see Table 5 or App. C.4). Mudgal et al. (2023); Deng & Raffel (2023) similarly propose to enforce the $T-t$ step optimality condition using a squared-error loss, $\sum_t \mathbb{E}_{p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t})}[(\phi(\mathbf{s}_{1:T}) - \psi_t(\mathbf{s}_{1:t}))^2]$.

5. Evaluating Inference in Language Models

Our SMC framework yields a rich set of tools to evaluate language model inference techniques using well-studied quantities such as the log partition function $\log \mathcal{Z}_\sigma$ and KL divergence to the target distribution. Remarkably, with access to a single exact sample from the target distribution, we show in Prop. 5.1 that we can obtain *upper* bounds on $\log \mathcal{Z}_\sigma$ in addition to lower bounds. These bounds can tightly sandwich $\log \mathcal{Z}_\sigma$ with increasing K, thereby ensuring reliable conclusions regarding inference quality.

5.1. Applications of $\log \mathcal{Z}_\sigma$ Estimation

Evaluating Fine-Tuned Models To motivate this section and present an important application of our SMC methods, consider evaluating how well a given $q(\mathbf{s}_{1:T})$ matches a target distribution for controlled generation or fine-tuning. Assume that q is tractable to sample and evaluate. To calculate the KL divergence to σ in either direction, we also require an estimate of the log partition function $\log \mathcal{Z}_\sigma$,

$$D_{\mathrm{KL}}(q(\mathbf{s}_{1:T}) \,\|\, \sigma(\mathbf{s}_{1:T})) = \mathbb{E}_q\left[\log \frac{q(\mathbf{s}_{1:T})}{p_0(\mathbf{s}_{1:T})\phi(\mathbf{s}_{1:T})}\right] + \log \mathcal{Z}_\sigma$$
$$D_{\mathrm{KL}}(\sigma(\mathbf{s}_{1:T}) \,\|\, q(\mathbf{s}_{1:T})) = \mathbb{E}_\sigma\left[\log \frac{p_0(\mathbf{s}_{1:T})\phi(\mathbf{s}_{1:T})}{q(\mathbf{s}_{1:T})}\right] - \log \mathcal{Z}_\sigma \tag{22}$$

For $D_{\mathrm{KL}}(\sigma \,\|\, q)$, note that we also require samples from the target σ, which may be readily available using the BDMC trick when σ is defined as a Bayesian posterior (Sec. 3.3). In such cases, we argue that SMC can be used to accurately bound the value of $\log \mathcal{Z}_\sigma$ and estimate each KL divergence above. Estimation of $D_{\mathrm{KL}}(\sigma \,\|\, q)$ may be particularly important to diagnose mode-dropping in inference techniques such as PPO which optimize the mode-seeking $D_{\mathrm{KL}}(q \,\|\, \sigma)$ during fine-tuning (Korbak et al., 2022b).

Evaluating Twisted SMC Sampling After running SIS or SMC with K samples, we can sample a single index as in Eq. (5) to return a single approximate target sample $\mathbf{s}^\sigma_{1:T}$. However, the marginal distribution of this sample, which we denote as $\mathbf{s}^\sigma_{1:T} \sim q_{\mathrm{SMC}}(\mathbf{s}_{1:T})$, is not tractable due to the need to sum over all possible sets of K samples. Nevertheless, we will show below that the tightness of our $\log \mathcal{Z}_\sigma$ lower or upper bounds in Prop. 5.1 provides upper bounds on the KL divergences $D_{\mathrm{KL}}(q_{\mathrm{SMC}}(\mathbf{s}_{1:T}) \,\|\, \sigma(\mathbf{s}_{1:T}))$ or $D_{\mathrm{KL}}(\sigma(\mathbf{s}_{1:T}) \,\|\, q_{\mathrm{SMC}}(\mathbf{s}_{1:T}))$, respectively.

Alternatively, we can also use the single-sample KL divergences in Eq. (22) for the twist-induced proposal q^π in Eq. (15) to evaluate a set of twist functions ψ_t (Sec. 7.2).

5.2. Bidirectional SMC Bounds on $\log \mathcal{Z}_\sigma$

Given the importance of $\log \mathcal{Z}_\sigma$ estimation as motivated above, we propose a *bidirectional SMC* stochastic upper bound which is novel (to the best of our knowledge), and may be of interest outside of the language modeling setting.

Recall from Sec. 2.2 that SMC admits an interpretation as SIS in an extended state space $\boldsymbol{S} := \{s_t^k, \omega_t^k\}_{k=1,t=1}^{K,T}$ which includes all tokens and resampling indices. We derive lower and upper bounds on $\log \mathcal{Z}_\sigma$ in Prop. 5.1 below, with proof and detailed description of the extended state space target $\sigma_{\mathrm{SMC}}(\boldsymbol{S})$ and proposal $q_{\mathrm{SMC}}(\boldsymbol{S})$ distributions in App. F.

Proposition 5.1. (Bidirectional SMC Bounds) *The log partition function $\log \mathcal{Z}_\sigma$ of a target distribution $\sigma(\mathbf{s}_{1:T})$ can be lower and upper bounded by*

$$\mathbb{E}_{q_{\mathrm{SMC}}(\boldsymbol{S})}\left[\log \prod_{t=1}^{T} \frac{1}{K} \sum_{i=1}^{K} w_t(\mathbf{s}^i_{1:t})\right] \le \log \mathcal{Z}_\sigma$$
$$\log \mathcal{Z}_\sigma \le \mathbb{E}_{\sigma_{\mathrm{SMC}}(\boldsymbol{S})}\left[\log \prod_{t=1}^{T} \frac{1}{K} \sum_{i=1}^{K} w_t(\mathbf{s}^i_{1:t})\right]. \tag{23}$$

The gap in the lower bound is $D_{\mathrm{KL}}(q_{\mathrm{SMC}}(\boldsymbol{S}) \,\|\, \sigma_{\mathrm{SMC}}(\boldsymbol{S}))$, and the gap in the upper bound is $D_{\mathrm{KL}}(\sigma_{\mathrm{SMC}}(\boldsymbol{S}) \,\|\, q_{\mathrm{SMC}}(\boldsymbol{S}))$.

The proof in App. F adapts the general approach for extended state space log partition function bounds from Brekelmans et al. (2021) using the probabilistic interpretation of SMC (Andrieu et al., 2010; Maddison et al., 2017). With no resampling, the SIS case recovers the Importance Weighted Autoencoder (IWAE) lower (Burda et al., 2015) and upper (Sobolev & Vetrov, 2019; Brekelmans et al., 2021) bounds.

Sampling from σ_{SMC} for SMC Upper Bounds We now discuss sampling from $\sigma_{\mathrm{SMC}}(\boldsymbol{S})$ for the expectation in the upper bound, which requires a single, *exact* sample from the target distribution $\sigma(\mathbf{s}_{1:T})$. This sample may be obtained, for example, using the BDMC trick in Sec. 3.3. Note that Sec. 2.2 and Alg. 1 describe sampling from $q_{\mathrm{SMC}}(\boldsymbol{S})$.

Sampling from $\sigma_{\mathrm{SMC}}(\boldsymbol{S})$ differs from sampling from $q_{\mathrm{SMC}}(\boldsymbol{S})$ by its treatment of the exact target sample. In particular, the partial sequence corresponding to the exact target sample is guaranteed to be cloned once at each resampling step. In other indices, resampling proceeds as in Sec. 2.2, where the exact sample may be cloned additional times based on its incremental importance weights. Finally, we sample $K-1$ next tokens from the proposal, while the value of the remaining chain is fixed by the exact target sample. See App. F and Alg. 2 for detailed discussion.

Tightness of the Bidirectional Bounds Since the bounds in Prop. 5.1 become exact as $K \to \infty$ for any proposal (Burda et al., 2015; Maddison et al., 2017), we can use SMC or IWAE with large K to sandwich the log partition function when σ samples are available.

For a given K, the gaps in the $\log \mathcal{Z}_\sigma$ bounds in Prop. 5.1 provide further insight into the quality of twisted SMC sampling via the marginal distribution of the sample $\mathbf{s}_{1:T}^\sigma$ (Sec. 5.1). The data processing inequality suggests that $D_{\mathrm{KL}}(q_{\mathrm{SMC}}(\mathbf{s}_{1:T}) \,\|\, \sigma(\mathbf{s}_{1:T})) \leq D_{\mathrm{KL}}(q_{\mathrm{SMC}}(\boldsymbol{S}) \,\|\, \sigma_{\mathrm{SMC}}(\boldsymbol{S}))$ and $D_{\mathrm{KL}}(\sigma(\mathbf{s}_{1:T}) \,\|\, q_{\mathrm{SMC}}(\mathbf{s}_{1:T})) \leq D_{\mathrm{KL}}(\sigma_{\mathrm{SMC}}(\boldsymbol{S}) \,\|\, q_{\mathrm{SMC}}(\boldsymbol{S}))$ (Grosse et al., 2015; 2016). Thus, if the difference between upper and lower bounds on $\log \mathcal{Z}_\sigma$ is small, then we can conclude that the K-sample SMC or SIS procedures in Sec. 2.2 yield a single approximate sample $\mathbf{s}_{1:T}^\sigma$ whose distribution $q_{\mathrm{SMC}}(\mathbf{s}_{1:T})$ is close to the target $\sigma(\mathbf{s}_{1:T})$ in symmetrized KL divergence.[2]

6. Related Work

In the previous sections, we have discussed related work as it fit within our SMC framework for language modeling. Note that Lew et al. (2023) consider SMC sampling for language models, but do not learn twist functions or proposals.

Decoding from language models to obtain diverse (Holtzman et al., 2019; Vilnis et al., 2023) or controlled generation (Zhang et al., 2023; Dathathri et al., 2019; Krause et al., 2020; Yang & Klein, 2021; Guo et al., 2021; Qin et al., 2022; Snell et al., 2022; Hu et al., 2023) is an active area of research. Our SMC resampling approach may be viewed as a principled *probabilistic* extension of best-of-K decoding methods. Mudgal et al. (2023) propose a K-way $\arg\max$ decoding scheme based on 'prefix scorers' ψ_t learned using Eq. (13), but also consider using these twists as logits for softmax sampling in the proposal. However, neither of these decoding schemes are aligned with our proposed SMC framework, as we discuss in App. D. For example, greedy $\arg\max$ decoding with respect to the optimal twists in Prop. 3.2 does not yield samples from the target $\sigma(\mathbf{s}_{1:T})$.

Finally, RL-based methods such as PPO maintain both a policy or proposal network and value network or advantage estimator during training. From the soft RL perspective in Sec. 3.4 and App. B.3, the soft values play a similar role as our twist functions for SMC resampling. Liu et al. (2023) consider using Monte Carlo Tree Search (MCTS) based on

7. Experiments

We now illustrate empirically how our framework can be used to evaluate inference through $\log \mathcal{Z}_\sigma$ bounds and KL divergences between the sampling and target distributions, providing meaningful quantitative comparison between various learning methods. We consider a range of tasks throughout this section, including toxic story generation (as an example of uncovering rare undesirable behavior), generating reviews with varied sentiment, and infilling. For the toxicity and infilling tasks, we consider the TinyStories model

[2]Note that the difference between upper and lower bound yields $D_{\mathrm{KL}}(\sigma_{\mathrm{SMC}}(\boldsymbol{S}) \,\|\, q_{\mathrm{SMC}}(\boldsymbol{S})) + D_{\mathrm{KL}}(q_{\mathrm{SMC}}(\boldsymbol{S}) \,\|\, \sigma_{\mathrm{SMC}}(\boldsymbol{S}))$.

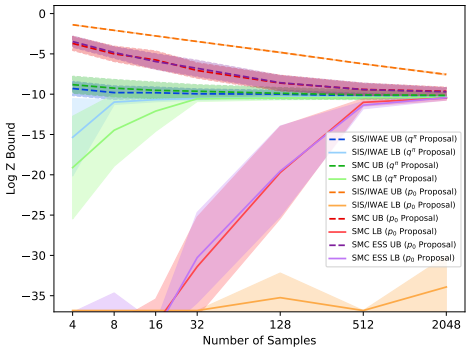


Figure 3: Comparison of SIS (IWAE) and SMC bounds on $\log \mathcal{Z}_\sigma$ for base proposal p_0 and twist-induced proposal q^π, with twists learned with CTL. With the twist-induced proposal, both SIS and SMC bounds are tight; with the base proposal, resampling with learned twists is needed. Resampling based on ESS instead of every-step resampling yields similar results.

(Eldan & Li, 2023) as a small-scale model where the generation is coherent, and use the prompt of 'Once upon a time, there was a'. For the toxicity task, we elicit responses judged to be toxic by the classifier from Corrêa (2023). For the sentiment task, we consider the GPT2-Medium model (Radford et al., 2019) and a classifier trained on Amazon reviews (Li, 2023). Our code is available at `https://github.com/Silent-Zebra/twisted-smc-lm`.

7.1. Comparing SIS and SMC for $\log \mathcal{Z}_\sigma$ Estimation

We first use our $\log \mathcal{Z}_\sigma$ bounds to test how twisted SMC can improve upon SIS and efficiently sample rare events. We consider the task of toxic story generation. The target is defined as $\sigma(\mathbf{s}_{1:T}) \propto p_0(\mathbf{s}_{1:T}) \mathbb{I}[\mathbf{s}_{1:T} \in \mathcal{C}]$ where $\mathcal{C} := \{\mathbf{s}_{1:T} \,|r(\mathbf{s}_{1:T}) \leq \eta\}$, $r(\mathbf{s}_{1:T})$ is the non-toxic logit, and the threshold $\eta = -5$ corresponds to a greater than 99% chance of being toxic. Rejection sampling under p_0 yields exact samples for $\log \mathcal{Z}_\sigma$ UB estimation, but can require hundreds of thousands of samples. Thus, this setting also allows us to test the effectiveness of approximate positive sampling for twist training when target samples are rare.

Fig. 3 demonstrates that training twists with CTL and approximate positive sampling can significantly improve log partition function estimation and sampling efficiency. We first note that both upper and lower bounds tighten as K increases, as expected, for both SIS and SMC. Using p_0 as proposal, the SIS LB (orange) generally fails to draw any samples meeting the threshold. By contrast, SMC resampling (red) with p_0 proposal eventually achieves *tight* $\log \mathcal{Z}_\sigma$ upper and lower bounds, yielding near-exact target samples (small KL divergence between the distribution over samples and the target distribution) by the reasoning in Sec. 5.

However, both SMC and SIS with the twist-induced proposal achieve tight estimation and near-exact sampling of σ with orders of magnitude lower K. Resampling does not appear to help or hurt these bounds, as the effect of the

Proposal q	Twist Learning	$D_{\mathrm{KL}}(q \,\|\, \sigma)$	$D_{\mathrm{KL}}(\sigma \,\|\, q)$
Twisted	Contrastive	1.11 ± 0.05	$\mathbf{1.07 \pm 0.02}$
Twisted	RL	1.52 ± 0.09	1.42 ± 0.03
Twisted	SIXO	1.71 ± 0.06	1.98 ± 0.04
Twisted	FUDGE	3.24 ± 0.26	2.00 ± 0.13
DPG	–	1.09 ± 0.05	1.12 ± 0.03
PPO	–	$\mathbf{0.98 \pm 0.01}$	1.32 ± 0.04

Table 1: Toxicity (Sec. 7.2.1)

Proposal q	Twist Learning	$D_{\mathrm{KL}}(q \,\|\, \sigma)$	$D_{\mathrm{KL}}(\sigma \,\|\, q)$
Twisted	Contrastive	$\mathbf{0.55 \pm 0.03}$	$\mathbf{0.47 \pm 0.01}$
Twisted	RL	0.94 ± 0.04	0.81 ± 0.02
Twisted	SIXO	0.73 ± 0.03	0.59 ± 0.02
Twisted	FUDGE	1.01 ± 0.07	0.77 ± 0.07
DPG	–	0.72 ± 0.04	0.57 ± 0.01
PPO	–	1.04 ± 0.31	0.87 ± 0.20

Table 2: Sentiment (Sec. 7.2.2)

Proposal q_{o_T}	Twist Learning	$\mathbb{E}_{o_T}[D_{\mathrm{KL}}(q_{o_T} \,\|\, \sigma_{o_T})]$	$\mathbb{E}_{o_T}[D_{\mathrm{KL}}(\sigma_{o_T} \,\|\, q_{o_T})]$
Twisted	Contrastive	23.93 ± 0.34	8.87 ± 0.05
Twisted	RL	31.35 ± 2.33	14.96 ± 1.69
Twisted	SIXO	20.34 ± 0.36	7.43 ± 0.04
Twisted	FUDGE	60.93 ± 2.82	19.85 ± 0.51
DPG	–	$\mathbf{13.27 \pm 0.44}$	$\mathbf{4.90 \pm 0.03}$
PPO	–	19.37 ± 0.41	14.07 ± 0.50

Table 3: Infilling (Sec. 7.2.3)

twists has been incorporated in the proposal q^π in Eq. (15). We conclude that the twist-induced proposal can provide significant efficiency gains over base model sampling.

7.2. Evaluating Twist-Induced or Variational Proposals

We next use our $\log \mathcal{Z}_\sigma$ bounds to evaluate single-sample inference using $D_{\mathrm{KL}}(q \,\|\, \sigma)$ and $D_{\mathrm{KL}}(\sigma \,\|\, q)$ (Sec. 5.1). We consider two SIS proposal-learning methods: PPO (Schulman et al., 2017) which minimizes $D_{\mathrm{KL}}(q \,\|\, \sigma)$ during training, and distributional policy gradient (DPG), which minimizes $D_{\mathrm{KL}}(\sigma \,\|\, q)$ (Parshakova et al., 2019) (see App. E).

We consider four twist learning methods, including CTL, soft Q-learning (RL), SIXO (Lawson et al., 2022), and FUDGE (Yang & Klein, 2021) (see App. C and Table 5). For each, we measure KL divergences involving the twist-induced proposal q^π. Thus, *these experiments showcase two complementary applications of SMC*: as a novel inference method yielding a tractable q^π, and as an evaluation method for any other inference method (such as PPO) using K-sample bounds on $\log \mathcal{Z}_\sigma$ to estimate the KL divergence.

7.2.1. Generating Toxic Stories

We consider toxic story generation as in Sec. 7.1, but using a target $\sigma(\mathbf{s}_{1:T}) \propto p_0(\mathbf{s}_{1:T})p(a = 1|\mathbf{s}_{1:T})$, where $p(a = 1|\mathbf{s}_{1:T})$ denotes the probability of the text being judged as toxic by a classifier. Compared to the thresholding target, this task provides a smoother gradient signal for learning but still allows for exact sampling via rejection sampling. We train using approximate positive sampling, but provide an ablation with exact positive sampling results in App. H.3.

We report KL divergences in Table 1. We observe that PPO performs best with respect to $D_{\mathrm{KL}}(q \,\|\, \sigma)$ while our CTL method performs best with respect to $D_{\mathrm{KL}}(\sigma \,\|\, q)$. This is consistent with the divergences minimized during training. In App. H.1 we provide a qualitative example of a toxic story generated with CTL for $\sigma(\mathbf{s}_{1:T}) \propto p_0(\mathbf{s}_{1:T})p(a = 1|\mathbf{s}_{1:T})^\beta$ with $\beta = 10$, a case where no exact samples are available.

7.2.2. Generation with Varied Sentiment

For the sentiment setting described earlier, we consider a prompt 'I bought this' and target $\sigma(\mathbf{s}_{1:T}) \propto p_0(\mathbf{s}_{1:T})p(a = 1|\mathbf{s}_{1:T})$, where $a = 1$ indicates a 1-star review and exact samples are available by rejection sampling. We train using approximate positive sampling (see App. H.3 for comparison with exact). While all methods achieve low KL divergences in Table 2, CTL performs best for both directions.

7.2.3. Infilling

Finally, we consider a *conditional twist learning* setting, where $\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}, o_T)$ takes input o_T that identifies the target distribution $\sigma(\mathbf{s}_{1:T}|o_T)$ (Sec. 3.3). We consider an infilling task (Lew et al., 2023; Hu et al., 2023), where the observation variables $o_T \coloneqq \mathbf{s}_{T+1:T+c}$ are continuation tokens with likelihoods $\sigma(o_T|\mathbf{s}_{1:T}) \coloneqq p_0(\mathbf{s}_{T+1:T+c}|\mathbf{s}_{1:T})$ evaluated under the base model given $\mathbf{s}_{1:T}$. Our target is the posterior $\sigma(\mathbf{s}_{1:T}|o_T)$. Instead of training separate $\psi_t^{\boldsymbol{\theta}}$ for each o_T, we amortize training a conditional twist network $\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}, o_T)$.

A second distinctive feature of this setting is that we *train* from exact posterior or target samples, which are readily available using the BDMC trick in Sec. 3.3. In particular, we may sample sequences of length $T + c$ from the base model $\mathbf{s}_{1:T+c} \sim p_0(\mathbf{s}_{1:T+c}) = \sigma(\mathbf{s}_{1:T}, o_T = \mathbf{s}_{T+1:T+c})$, and interpret the prefix $\mathbf{s}_{1:T} \sim \sigma(\mathbf{s}_{1:T}|o_T = \mathbf{s}_{T+1:T+c})$ as a target sample. Note that we do not explicitly control the continuations tokens o_T defining the tasks. We evaluate average KL divergences over 2000 different $o_T = \mathbf{s}_{T+1:T+c}$, with $T = 15$ and $c = 10$, and report results in Table 3.

We find DPG performs best for both directions of the KL divergence in this setting, likely due to its ability to leverage exact positive samples by minimizing $D_{\mathrm{KL}}(\sigma_{o_T} \,\|\, q_{o_T})$. CTL also learns from exact positive samples, but requires approximate negative sampling and only performs comparably to SIXO, which uses exact positive samples and performs exact negative sampling under p_0. Finally, PPO trains from q_{o_T} samples only, and performs relatively poorly with respect to $D_{\mathrm{KL}}(\sigma_{o_T} \,\|\, q_{o_T})$. To correlate these results with sample quality, we show qualitative results in App. H.1.

Using our KL divergence evaluation methods, we conclude DPG may be preferable when exact target samples are available (Sec. 7.2.3, App. H.3), while CTL may be preferable with approximate positive sampling (Sec. 7.2.1, Sec. 7.2.2).

8. Conclusion

In this work, we have presented twisted SMC as a principled probabilistic inference framework for solving numerous capability and safety tasks in LLMs. After discussing different design choices for twisted SMC and their relation to related work, we proposed a novel contrastive method for twist learning. Further, we proposed novel bidirectional SMC bounds for evaluating LLM inference methods. We demonstrated the effectiveness of our methods for both sampling and evaluation across a variety of experimental settings.

Acknowledgments

AM and RG acknowledge support from the Canada CIFAR AI Chairs program and from Open Philanthropy. SZ thanks Juhan Bae for helping debug memory issues in the code. Resources used in this research were provided, in part, by the Province of Ontario, the Government of Canada, and companies sponsoring the Vector Institute. We thank the anonymous reviewers for helpful comments on earlier versions of this paper.

Impact Statement

This paper is motivated by the social consequences of recent advances in the field of machine learning. Controlled generation from language models has the potential to improve safety through better steering of generation to human preferences, more efficient automated red-teaming, and the ability to estimate or bound probabilities of rare behaviors. Any such work is inherently a double-edged sword; the same techniques used to generate samples from a harmless distribution of text could, with a single sign change, be repurposed for generating samples from a harmful distribution of text. Thus, better controlled generation (in our framework, better sampling from target distributions) can provide benefits in the hands of responsible users but can also magnify harms in the hands of malevolent users (who have access to model weights).

Overall, we believe the potential positive social benefits of our work in evaluation and steering language model output towards desired target distributions outweigh the potential negatives stemming primarily from misuse.

References

Andrieu, C., Doucet, A., and Holenstein, R. Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 72(3): 269–342, 2010.

Anil, C., Zhang, G., Wu, Y., and Grosse, R. Learning to give checkable answers with prover-verifier games. *arXiv preprint arXiv:2108.12099*, 2021.

Bae, J., Zhang, M. R., Ruan, M., Wang, E., Hasegawa, S., Ba, J., and Grosse, R. B. Multi-rate vae: Train once, get the full rate-distortion curve. In *The Eleventh International Conference on Learning Representations*, 2022.

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Banerjee, A., Guo, X., and Wang, H. On the optimality of conditional expectation as a bregman predictor. *IEEE Transactions on Information Theory*, 51(7), 2005.

Brekelmans, R., Huang, S., Ghassemi, M., Ver Steeg, G., Grosse, R. B., and Makhzani, A. Improving mutual information estimation with annealed and energy-based bounds. In *International Conference on Learning Representations*, 2021.

Briers, M., Doucet, A., and Maskell, S. Smoothing algorithms for state–space models. *Annals of the Institute of Statistical Mathematics*, 62:61–89, 2010.

Burda, Y., Grosse, R., and Salakhutdinov, R. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

Chopin, N., Papaspiliopoulos, O., et al. *An introduction to sequential Monte Carlo*, volume 4. Springer, 2020.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Corrêa, N. K. Aira, 2023. URL https://huggingface.co/nicholasKluge/ToxicityModel.

Dathathri, S., Madotto, A., Lan, J., Hung, J., Frank, E., Molino, P., Yosinski, J., and Liu, R. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*, 2019.

Del Moral, P., Doucet, A., and Jasra, A. Sequential monte carlo samplers. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 68(3):411–436, 2006.

Deng, H. and Raffel, C. Reward-augmented decoding: Efficient controlled text generation with a unidirectional reward model. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Dohan, D., Xu, W., Lewkowycz, A., Austin, J., Bieber, D., Lopes, R. G., Wu, Y., Michalewski, H., Saurous, R. A., Sohl-Dickstein, J., et al. Language model cascades. *arXiv preprint arXiv:2207.10342*, 2022.

Domke, J. and Sheldon, D. R. Importance weighting and variational inference. *Advances in neural information processing systems*, 31, 2018.

Doucet, A., De Freitas, N., Gordon, N. J., et al. *Sequential Monte Carlo methods in practice*, volume 1. Springer, 2001.

Eikema, B., Kruszewski, G., Dance, C. R., Elsahar, H., and Dymetman, M. An approximate sampler for energy-based models with divergence diagnostics. *Transactions on Machine Learning Research*, 2022.

Eldan, R. and Li, Y. Tinystories: How small can language models be and still speak coherent english? *arXiv preprint arXiv:2305.07759*, 2023. URL https://huggingface.co/roneneldan/TinyStories-33M.

Finke, A. *On extended state-space constructions for Monte Carlo methods*. PhD thesis, University of Warwick, 2015.

Go, D., Korbak, T., Kruszewski, G., Rozen, J., Ryu, N., and Dymetman, M. Aligning foundation models for language with preferences through f-divergence minimization. In *International Conference on Machine Learning*, 2023.

Grosse, R. B., Ghahramani, Z., and Adams, R. P. Sandwiching the marginal likelihood using bidirectional monte carlo. *arXiv preprint arXiv:1511.02543*, 2015.

Grosse, R. B., Ancha, S., and Roy, D. Measuring the reliability of mcmc inference with bidirectional monte carlo. *Advances in Neural Information Processing Systems*, 2016.

Gu, S. S., Ghahramani, Z., and Turner, R. E. Neural adaptive sequential monte carlo. *Advances in neural information processing systems*, 28, 2015.

Guo, H., Tan, B., Liu, Z., Xing, E. P., and Hu, Z. Efficient (soft) q-learning for text generation with limited good data. *arXiv preprint arXiv:2106.07704*, 2021.

Gutmann, M. and Hyvärinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *International conference on artificial intelligence and statistics*, pp. 297–304. JMLR Workshop and Conference Proceedings, 2010.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*. PMLR, 2018.

Heng, J., Bishop, A., Deligiannidis, G., and Doucet, A. Controlled sequential monte carlo. *Annals of Statistics*, 48(5), 2020.

Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2019.

Hu, E. J., Jain, M., Elmoznino, E., Kaddar, Y., Lajoie, G., Bengio, Y., and Malkin, N. Amortizing intractable inference in large language models. *arXiv preprint arXiv:2310.04363*, 2023.

Khalifa, M., Elsahar, H., and Dymetman, M. A distributional approach to controlled text generation. *arXiv preprint arXiv:2012.11635*, 2020.

Khanov, M., Burapacheep, J., and Li, Y. ARGS: Alignment as reward-guided search. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=shgx0eqdw6.

Korbak, T., Elsahar, H., Kruszewski, G., and Dymetman, M. Controlling conditional language models without catastrophic forgetting. In *International Conference on Machine Learning*, pp. 11499–11528. PMLR, 2022a.

Korbak, T., Perez, E., and Buckley, C. L. Rl with kl penalties is better viewed as bayesian inference. *arXiv preprint arXiv:2205.11275*, 2022b.

Krause, B., Gotmare, A. D., McCann, B., Keskar, N. S., Joty, S., Socher, R., and Rajani, N. F. Gedi: Generative discriminator guided sequence generation. *arXiv preprint arXiv:2009.06367*, 2020.

Lawson, D., Tucker, G., Naesseth, C. A., Maddison, C., Adams, R. P., and Teh, Y. W. Twisted variational sequential monte carlo. In *Third workshop on Bayesian Deep Learning (NeurIPS)*, 2018.

Lawson, D., Raventós, A., Warrington, A., and Linderman, S. Sixo: Smoothing inference with twisted objectives, 2022.

Levine, S. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

Lew, A. K., Zhi-Xuan, T., Grand, G., and Mansinghka, V. K. Sequential monte carlo steering of large language models using probabilistic programs. *arXiv preprint arXiv:2306.03081*, 2023.

Li, Y. Distilbert-base-uncased-finetuned-mnli-amazon-query-shopping, 2023. URL https://huggingface.co/LiYuan/amazon-review-sentiment-analysis.

Lioutas, V., Lavington, J. W., Sefas, J., Niedoba, M., Liu, Y., Zwartsenberg, B., Dabiri, S., Wood, F., and Scibior, A. Critic sequential monte carlo. In *The Eleventh International Conference on Learning Representations*, 2022.

Liu, A., Sap, M., Lu, X., Swayamdipta, S., Bhagavatula, C., Smith, N. A., and Choi, Y. Dexperts: Decoding-time controlled text generation with experts and anti-experts. In *59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 2021.

Liu, J., Cohen, A., Pasunuru, R., Choi, Y., Hajishirzi, H., and Celikyilmaz, A. Don't throw away your value model! making ppo even better via value-guided monte-carlo tree search decoding. *arXiv e-prints*, pp. arXiv–2309, 2023.

Maddison, C. J., Lawson, J., Tucker, G., Heess, N., Norouzi, M., Mnih, A., Doucet, A., and Teh, Y. Filtering variational objectives. *Advances in Neural Information Processing Systems*, 30, 2017.

Mudgal, S., Lee, J., Ganapathy, H., Li, Y., Wang, T., Huang, Y., Chen, Z., Cheng, H.-T., Collins, M., Strohman, T., et al. Controlled decoding from language models. *arXiv preprint arXiv:2310.17022*, 2023.

Nachum, O., Norouzi, M., Xu, K., and Schuurmans, D. Bridging the gap between value and policy based reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Parshakova, T., Andreoli, J.-M., and Dymetman, M. Distributional reinforcement learning for energy-based sequential models. *arXiv preprint arXiv:1912.08517*, 2019.

Perez, E., Huang, S., Song, F., Cai, T., Ring, R., Aslanides, J., Glaese, A., McAleese, N., and Irving, G. Red teaming language models with language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 3419–3448, 2022.

Phan, D., Hoffman, M. D., Douglas, S., Le, T. A., Parisi, A. T., Sountsov, P., Sutton, C., Vikram, S., Saurous, R. A., et al. Training chain-of-thought via latent-variable inference. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Piché, A., Thomas, V., Ibrahim, C., Bengio, Y., and Pal, C. Probabilistic planning with sequential monte carlo methods. In *International Conference on Learning Representations*, 2018.

Qin, L., Welleck, S., Khashabi, D., and Choi, Y. Cold decoding: Energy-based constrained text generation with langevin dynamics. *Advances in Neural Information Processing Systems*, 35:9538–9551, 2022.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. 2019. URL https://huggingface.co/gpt2-medium.

Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023.

Scharth, M. and Kohn, R. Particle efficient importance sampling. *Journal of Econometrics*, 190(1):133–147, 2016.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Shih, A., Sadigh, D., and Ermon, S. Long horizon temperature scaling. *arXiv preprint arXiv:2302.03686*, 2023.

Snell, C. V., Kostrikov, I., Su, Y., Yang, S., and Levine, S. Offline rl for natural language generation with implicit language q learning. In *The Eleventh International Conference on Learning Representations*, 2022.

Sobolev, A. and Vetrov, D. P. Importance weighted hierarchical variational inference. *Advances in Neural Information Processing Systems*, 32, 2019.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33: 3008–3021, 2020.

Vilnis, L., Zemlyanskiy, Y., Murray, P., Passos, A. T., and Sanghai, S. Arithmetic sampling: parallel diverse decoding for large language models. In *International Conference on Machine Learning*. PMLR, 2023.

Whiteley, N. and Lee, A. Twisted particle filters. *The Annals of Statistics*, 42(1):115–141, 2014.

Yang, K. and Klein, D. Fudge: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3511–3535, 2021.

Zhang, H., Song, H., Li, S., Zhou, M., and Song, D. A survey of controllable text generation using transformer-based pre-trained language models. *ACM Computing Surveys*, 56(3):1–37, 2023.

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

Zou, A., Wang, Z., Kolter, J. Z., and Fredrikson, M. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

Appendix

Table of Contents

Table 4: Examples of Target Posteriors in Language Model Finetuning and Controlled Generation

Type	Target	References / Examples
Reward	$\sigma(\mathbf{s}_{1:T}) \propto p_0(\mathbf{s}_{1:T}) e^{\pm \beta r(\mathbf{s}_{1:T})}$	RLHF (Ziegler et al., 2019; Ouyang et al., 2022; Korbak et al., 2022b)
Continuation	$\sigma(\mathbf{s}_{1:T}) \propto p_0(\mathbf{s}_{1:T}) p_0(\mathbf{s}_{T+1:T+c} \vert \mathbf{s}_{1:T})^{\beta}$	Generates tokens based on likelihood of future tokens $p(\mathbf{s}_{T+1:T+c} \vert \mathbf{s}_{1:T})$ For $\beta = 1$, this is in-filling (Lew et al., 2023). As $\beta \to \infty$, disregard $p_0(\mathbf{s}_{1:T})$, focus on $\arg\max$ of continuation prob. - similar to adversarial prompt generation (Zou et al., 2023)
Indicator	$\sigma(\mathbf{s}_{1:T}) \propto p_0(\mathbf{s}_{1:T}) \mathbb{I}[\mathbf{s}_{1:T} \in \mathcal{C}]$ where $\mathbb{I}$ is indicator of set $\mathcal{C}$: $\mathbb{I}[\mathbf{s}_{1:T} \in \mathcal{C}] = 1$ if $[\mathbf{s}_{1:T} \in \mathcal{C}]$ $\mathbb{I}[\mathbf{s}_{1:T} \in \mathcal{C}] = 0$ if $[\mathbf{s}_{1:T} \notin \mathcal{C}]$	Generations $\mathbf{s}_{1:T}$ from this target must satisfy the properties of set $\mathcal{C}$. - Meeting reward threshold $\mathcal{C}_{r \leq \eta} \coloneqq \{\mathbf{s}_{1:T} \mid \pm r(\mathbf{s}_{1:T}) \leq \eta\}$ - Containing topical or specific words in $\mathbf{s}_{1:T}$ - Having certain structure or rhyme (Yang & Klein, 2021), - Valid output according to verifier (Cobbe et al., 2021; Dohan et al., 2022))
Classifier	$\sigma(\mathbf{s}_{1:T}) \propto p_0(\mathbf{s}_{1:T}) p(y \vert \mathbf{s}_{1:T})^{\beta}$	Class y can be a binary (e.g. toxicity) or multinomial (e.g. 1-5 star reviews) Bayesian posterior for $\beta = 1$: $\sigma(\mathbf{s}_{1:T}) = p(\mathbf{s}_{1:T} \vert y) \propto p_0(\mathbf{s}_{1:T}) p(y \vert \mathbf{s}_{1:T})$ (Dathathri et al., 2019; Krause et al., 2020; Liu et al., 2021)
Global Temperature	$\sigma(\mathbf{s}_{1:T}) \propto p_0(\mathbf{s}_{1:T})^{\beta}$	Tempering on entire sequences (long-horizon) vs. per-token (myopic) - yields higher quality generation in Shih et al. (2023)
Distributional	$\sigma(\mathbf{s}_{1:T}) \propto p_0(\mathbf{s}_{1:T}) e^{\boldsymbol{\beta} \cdot \boldsymbol{T}(\mathbf{s}_{1:T})}$	KL minimization subj. expectation constraints on $\boldsymbol{T} = \{T_i\}$ $\arg\min D_{\mathrm{KL}}(q(\mathbf{s}_{1:T}) \Vert p_0(\mathbf{s}_{1:T}))$ s.t. $\mathbb{E}_q[\boldsymbol{T}] = \boldsymbol{\eta}_{\boldsymbol{\beta}}$ ($\boldsymbol{\beta}$ = optimal Lagrange multipliers for constraints $\boldsymbol{\eta}$) e.g. gender roles/references (Khalifa et al., 2020)
Intermediate		References / Examples
Indicator	$\phi_t(\mathbf{s}_{1:t}) = \mathbb{I}[s_t \in \mathcal{C}]$ or $\mathbb{I}[\mathbf{s}_{1:t} \in \mathcal{C}]$	words of specific length, or specific sets of tokens (Khalifa et al., 2020; Lew et al., 2023)
Product of Experts	$\sigma(\mathbf{s}_{1:T}) \propto \prod_{m=1}^{M} \prod_{t=1}^{T} p_0(s_t \vert \mathbf{s}_{1:t-1}, \mathbf{s}_0^{(m)})$	prompt intersection (Lew et al., 2023)

Table 5: Losses for twist (top) and proposal (bottom) learning, where $\pi_s(\cdot)$ indicates an arbitrary sampling distribution. See App. C for detailed discussion and additional losses.

Name	***Loss***	***Learning Principle***
CTL	$\sum_{t=1}^{T} \mathbb{E}_{\pi_s(o_T)}\left[D_{\mathrm{KL}}(\sigma(\mathbf{s}_{1:t} \vert o_T) \,\|\, \pi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t} \vert o_T))\right]$ *(Gradient:)* $-\mathbb{E}_{\pi_s(o_T)}\left[\mathbb{E}_{\sigma(\mathbf{s}_{1:t} \vert o_T)}\left[\nabla_{\boldsymbol{\theta}} \log \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}, o_T)\right] - \mathbb{E}_{\pi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t} \vert o_T)}\left[\nabla_{\boldsymbol{\theta}} \log \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}, o_T)\right]\right]$	Marginal Matching with MLE
Soft RL	$\sum_{t=1}^{T-1} \mathbb{E}_{\pi_s(\mathbf{s}_{1:t}, o_T)}\left[\left(\log \sum_{s_{t+1}} p_0(s_{t+1} \vert \mathbf{s}_{1:t}) \mathrm{sg}\left(\psi_{t+1}^{\boldsymbol{\theta}}(\mathbf{s}_{1:t+1}, o_T)\right) - \log \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}, o_T)\right)^2\right] + \mathbb{E}_{\pi_s(\mathbf{s}_{1:T}, o_T)}\left[\left(\log \phi(\mathbf{s}_{1:T}, o_T) - \log \psi_T^{\boldsymbol{\theta}}(\mathbf{s}_{1:T}, o_T)\right)^2\right]$	Twist Consistency / Soft Q-Learning
SIXO	$\sum_{t=1}^{T} -\left[\mathbb{E}_{\pi_s(o_T)\sigma(\mathbf{s}_{1:t} \vert o_T)}\left[\log \mathtt{sigmoid}\left(\log \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}, o_T)\right)\right] + \mathbb{E}_{p_0(\mathbf{s}_{1:t})\pi_s(o_T)}\left[\log\left(1 - \mathtt{sigmoid}\left(\log \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}, o_T)\right)\right)\right]\right]$	Noise Contrastive Estimation
FUDGE	$\sum_{t=1}^{T} -\mathbb{E}_{\pi_s(\mathbf{s}_{1:t}, o_T)}\mathbb{E}_{p_0(\mathbf{s}_{t+1:T} \vert \mathbf{s}_{1:t})}\left[\sigma(o_T \vert \mathbf{s}_{1:T}) \log \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}, o_T) + \left(1 - \sigma(o_T \vert \mathbf{s}_{1:T})\right) \log\left(1 - \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}, o_T)\right)\right)\right]$	Binary Classification
DPG	$\mathbb{E}_{\pi_s(o_T)}\left[D_{\mathrm{KL}}(\sigma(\mathbf{s}_{1:T} \vert o_T) \,\|\, q^{\boldsymbol{\xi}}(\mathbf{s}_{1:T} \vert o_T))\right]$	Maximum Likelihood (MLE)
PPO	$\mathbb{E}_{\pi_s(o_T)}\left[D_{\mathrm{KL}}(q^{\boldsymbol{\xi}}(\mathbf{s}_{1:T} \vert o_T) \,\|\, \sigma(\mathbf{s}_{1:T} \vert o_T))\right]$	Variational Inference

A. Proofs

In this section, we present the sense in which the target marginals correspond to the *optimal* intermediate distributions in twisted SMC. We defer proof of Prop. 3.2 from the main text to slightly more general version in App. B.1 Prop. B.1, although Prop. A.4 provides the analogous statement in terms of the intermediate target distributions $\pi_t^*(\mathbf{s}_{1:t}) = \sigma(\mathbf{s}_{1:t})$ instead of the optimal twists ψ_t^*.

We also prove Prop. 3.3 from the main text in App. A.2 and derive the gradient of the CTL loss (Eq. (21)) in App. A.3.

A.1. Proof for Optimal Intermediate Target Distributions

In order to achieve sampling from the full joint distribution $\sigma(\mathbf{s}_{1:T})$, each intermediate target $\sigma(\mathbf{s}_{1:t})$ must match the intermediate marginal $\sigma(\mathbf{s}_{1:t})$. To formalize this notion, we provide the following definition of optimality, justified by the fact that it yields an exact partition function estimator.

To do so, we will consider the multi-step importance weights

$$w_{t:t+c-1}(\mathbf{s}_{1:t+c-1}) = \prod_{\tau=t}^{t+c-1} w_\tau(\mathbf{s}_{1:\tau}) = \prod_{\tau=t}^{t+c-1} \frac{\tilde{\pi}_\tau(\mathbf{s}_{1:\tau})}{\tilde{\pi}_{\tau-1}(\mathbf{s}_{1:\tau-1})q(s_\tau|\mathbf{s}_{1:\tau-1})} = \frac{\tilde{\pi}_{t+c-1}(\mathbf{s}_{1:t+c-1})}{\tilde{\pi}_{t-1}(\mathbf{s}_{1:t-1})q(\mathbf{s}_{t:t+c-1}|\mathbf{s}_{1:t-1})}$$

(c-Step SMC Weights)

using a telescoping cancellation in the final equality. The one-step weights correspond to $c = 1$, denoted simply as w_t.

Definition A.1 (Optimal Twisted SMC Sampling). *For a given target distribution $\sigma(\mathbf{s}_{1:T}) \propto p_0(\mathbf{s}_{1:T})\phi(\mathbf{s}_{1:T})$, we refer to a twisted SMC procedure,* $\text{SMC}(\{\pi_t\}_{t=1}^T, q, K)$ *or* $\text{SMC}(p_0, \{\psi_t\}_{t=1}^T, q, K)$ *(with $\pi_T = \sigma$ or $\psi_T = \phi$), as* optimal *if c-step importance weights $w_{t:t+c-1}(\mathbf{s}_{1:t+c-1}) = \mathcal{Z}^\psi_{t+c-1}/\mathcal{Z}^\psi_{t-1}$ for all $1 \le t \le T$ and $0 \le c \le T - t + 1$.*

Note, that the role of ψ_t and $\mathcal{Z}^\psi_t$ is specified in Def. 3.1. We assume $\pi_T = \sigma$ for the goal of estimating $\mathcal{Z}_\sigma$, and show below that an optimal twisted SMC procedure yields an exact partition function estimator.

Proposition A.2 (Optimal SMC yields Exact Partition Function Estimation). *For any optimal twisted SMC procedure, the resulting estimator of the partition function $\mathcal{Z}_\sigma$ has zero bias and zero variance.*

Proof. As in Footnote 1 or App. F Alg. 2, consider $\{t_r\}_{r=1}^R$ index timesteps where resampling occurs and fix $t_0 = 0$ and $t_R = T$. The SMC estimator of $\mathcal{Z}_\sigma = \mathcal{Z}^\psi_T$ becomes $\hat{\mathcal{Z}}^{\text{SMC}}_\sigma = \prod_{r=1}^R \frac{1}{K}\sum_{i=1}^K \left(\prod_{t=t_{r-1}+1}^{t_r} w_t(\mathbf{s}^i_{1:t})\right)$ for $\boldsymbol{S} \sim q_{\text{SMC}}(\boldsymbol{S})$. Using the optimality definition in Def. A.1, we have $w_t(\mathbf{s}_{1:t}) = \mathcal{Z}^\psi_t/\mathcal{Z}^\psi_{t-1}$ for all partial sequences $\mathbf{s}_{1:t}$. Noting the telescoping multiplicative cancellation and the fact that $w_t(\mathbf{s}^i_{1:t}) = \mathcal{Z}^\psi_t/\mathcal{Z}^\psi_{t-1}$ is constant with respect to indices $i \in [1, K]$, we have the following estimator for a single run of an optimal SMC procedure,

$$\hat{\mathcal{Z}}^{\text{SMC}}_\sigma = \prod_{r=1}^R \frac{1}{K}\sum_{i=1}^K \left(\prod_{t=t_{r-1}+1}^{t_r} w_t\left(\mathbf{s}^i_{1:t}\right)\right) = \prod_{r=1}^R \frac{\mathcal{Z}^\psi_{t_r}}{\mathcal{Z}^\psi_{t_{r-1}}} = \frac{\mathcal{Z}^\psi_{t_R}}{\mathcal{Z}^\psi_{t_0}} = \frac{\mathcal{Z}^\psi_T}{\mathcal{Z}^\psi_0} = \mathcal{Z}_\sigma \tag{24}$$

as desired, assuming $\mathcal{Z}^\psi_0 = 1$. Since $\hat{\mathcal{Z}}^{\text{SMC}}_\sigma = \mathcal{Z}_\sigma$ is independent of $\boldsymbol{S}$, we conclude $\hat{\mathcal{Z}}^{\text{SMC}}_\sigma$ has zero bias and zero variance.

Note that we could also define optimality in Def. A.1 using the condition that $w_{t:t+c-1}(\mathbf{s}_{1:t+c-1}) = \text{const}$ for all $1 \le t \le T$ and $0 \le c \le T - t + 1$. Following similar derivations as above would yield $\hat{\mathcal{Z}}^{\text{SMC}}_\sigma = \text{const}$. As we will show in App. F, $\hat{\mathcal{Z}}^{\text{SMC}}_\sigma$ is unbiased with $\mathbb{E}[\hat{\mathcal{Z}}^{\text{SMC}}_\sigma] = \mathcal{Z}_\sigma$. We thus conclude that $\hat{\mathcal{Z}}^{\text{SMC}}_\sigma = \mathcal{Z}_\sigma$ with zero variance, and thus Prop. A.2 holds. □

With this notion of optimality in mind, we demonstrate the following necessary and sufficient conditions.

Proposition A.3 (Optimality Conditions). *The following conditions are necessary and sufficient for twisted SMC optimality,*

$$\begin{aligned} (i): &\qquad \pi^*_t(\mathbf{s}_{1:t}) = \sigma(\mathbf{s}_{1:t}) &&\forall \quad 1 \le t \le T \\ (ii): &\qquad q^*_t(s_t|\mathbf{s}_{1:t-1}) = \sigma(s_t|\mathbf{s}_{1:t-1}) &&\forall \quad 1 \le t \le T\,. \end{aligned} \tag{25}$$

Proof. (Necessary) Optimal Twisted SMC $\implies$ $(i), (ii)$: We begin by writing the marginalization of the unnormalized density $\tilde{\pi}^*_{t+c}$ over prefixes of length t as

$$\tilde{\pi}^*_{t+c}(\mathbf{s}_{1:t}) = \sum_{\mathbf{s}_{t+1:t+c}} \tilde{\pi}^*_{t+c}(\mathbf{s}_{1:t+c}) = \sum_{\mathbf{s}_{t+1:t+c}} p_0(\mathbf{s}_{1:t+c})\psi_{t+c}(\mathbf{s}_{1:t+c}) = p_0(\mathbf{s}_{1:t}) \sum_{\mathbf{s}_{t+1:t+c}} p_0(\mathbf{s}_{t+1:t+c}|\mathbf{s}_{1:t})\psi_{t+c}(\mathbf{s}_{1:t+c})$$

The normalization constant of $\tilde{\pi}^*_{t+c}(\mathbf{s}_{1:t})$ can easily be seen to be $\mathcal{Z}^{\psi^*}_{t+c}$ after summing over $\mathbf{s}_{1:t}$ above, which yields $\pi^*_{t+c}(\mathbf{s}_{1:t}) = \tilde{\pi}^*_{t+c}(\mathbf{s}_{1:t})/\mathcal{Z}^{\psi^*}_{t+c}$. We now factorize the c-step incremental importance weights (at step $t+1$, see Eq. (c-Step SMC Weights)) using the above identities, which imply that $\tilde{\pi}^*_{t+c}(\mathbf{s}_{1:t+c}) = \mathcal{Z}^{\psi^*}_{t+c}\pi^*_{t+c}(\mathbf{s}_{1:t+c}) = \mathcal{Z}^{\psi^*}_{t+c}\pi^*_{t+c}(\mathbf{s}_{1:t})\pi^*_{t+c}(\mathbf{s}_{t+1:t+c}|\mathbf{s}_{1:t})$ and

$$w_{t+1:t+c}(\mathbf{s}_{1:t+c}) = \frac{\tilde{\pi}^*_{t+c}(\mathbf{s}_{1:t+c})}{\tilde{\pi}^*_t(\mathbf{s}_{1:t})q^*(\mathbf{s}_{t+1:t+c}|\mathbf{s}_{1:t})} = \frac{\mathcal{Z}^{\psi^*}_{t+c}}{\mathcal{Z}^{\psi^*}_t}\frac{\pi^*_{t+c}(\mathbf{s}_{1:t})}{\pi^*_t(\mathbf{s}_{1:t})}\frac{\pi^*_{t+c}(\mathbf{s}_{t+1:t+c}|\mathbf{s}_{1:t})}{q^*(\mathbf{s}_{t+1:t+c}|\mathbf{s}_{1:t})} \tag{26}$$

In order to have $w_{t+1:t+c}(\mathbf{s}_{1:t+c}) = \mathcal{Z}^{\psi^*}_{t+c}/\mathcal{Z}^{\psi^*}_t$ in general, we thus require $\pi^*_{t+c}(\mathbf{s}_{1:t}) = \pi^*_t(\mathbf{s}_{1:t})$ and $\pi^*_{t+c}(\mathbf{s}_{t+1:t+c}|\mathbf{s}_{1:t}) = q^*(\mathbf{s}_{t+1:t+c}|\mathbf{s}_{1:t})$ for all t and $c \leq T-t$.

(Sufficient) $(i), (ii) \implies$ *Optimal Twisted SMC:* Consider the incremental importance weights using (i) and (ii),

$$w_t(\mathbf{s}_{1:t}) = \frac{\tilde{\pi}^*_t(\mathbf{s}_{1:t})}{\tilde{\pi}^*_{t-1}(\mathbf{s}_{1:t-1})q^{\pi^*}_t(s_t|\mathbf{s}_{1:t-1})} = \frac{\mathcal{Z}^\psi_t \sigma(\mathbf{s}_{1:t})}{\mathcal{Z}^\psi_{t-1}\sigma(\mathbf{s}_{1:t-1})\sigma(s_t|\mathbf{s}_{1:t-1})} = \frac{\mathcal{Z}^\psi_t}{\mathcal{Z}^\psi_{t-1}} \tag{27}$$

which matches the optimality definition in Def. A.1. □

Proposition A.4 (Optimal Intermediate Target Distributions). *For a given target distribution $\sigma(\mathbf{s}_{1:T})$ (Eq. (30)), the following conditions are equivalent, and are necessary for optimality of a twisted SMC procedure involving $\{\pi^*_t\}_{t=1}^T$,*

$$\begin{aligned}
(i): &\quad \pi^*_t(\mathbf{s}_{1:t}) = \sum_{s_{t+1}} \pi^*_{t+1}(\mathbf{s}_{1:t+1}) && \forall \quad 1 \leq t \leq T-1\,, \\
(ii): &\quad \pi^*_t(\mathbf{s}_{1:t}) = \sum_{\mathbf{s}_{t+1:t+c}} \pi^*_{t+c}(\mathbf{s}_{1:t+c}) && \forall \quad 1 \leq t \leq T-1,\ 1 \leq c \leq T-t\,, \\
(iii): &\quad \pi^*_t(\mathbf{s}_{1:t}) = \sigma(\mathbf{s}_{1:t}) && \forall \quad 1 \leq t \leq T\,.
\end{aligned} \tag{28}$$

Conditions (i) and (iii) directly correspond to the recursions for the optimal twist functions given in Prop. 3.2 and Prop. B.1.

Proof. $(i) \iff (ii)$*:* It is clear that $(ii) \implies (i)$ as a special case for $c = 1$. To show $(i) \implies (ii)$, we have

$$\pi^*_t(\mathbf{s}_{1:t}) = \sum_{s_{t+1}} \pi^*_{t+1}(\mathbf{s}_{1:t+1}) = \sum_{s_{t+1}}\sum_{s_{t+2}} \pi^*_{t+2}(\mathbf{s}_{1:t+2}) = \ldots = \sum_{\mathbf{s}_{t+1:t+c}} \pi^*_{t+c}(\mathbf{s}_{1:t+c}).$$

$(i) \implies (iii)$: Recursively applying (i) until time T suggests that

$$\pi^*_t(\mathbf{s}_{1:t}) = \sum_{s_{t+1}} \pi^*_{t+1}(\mathbf{s}_{1:t+1}) = \sum_{s_{t+1}}\sum_{s_{t+2}} \pi^*_{t+2}(\mathbf{s}_{1:t+2}) = \ldots = \sum_{\mathbf{s}_{t+1:T}} \pi^*_T(\mathbf{s}_{1:T}) = \sum_{\mathbf{s}_{t+1:T}} \sigma(\mathbf{s}_{1:T}) = \sigma(\mathbf{s}_{1:t}).$$

$(iii) \implies (i)$: The target marginals clearly satisfy the recursion

$$\sigma(\mathbf{s}_{1:t}) \coloneqq \sum_{\mathbf{s}_{t+1:T}} \sigma(\mathbf{s}_{1:T}) = \sum_{s_{t+1}}\sum_{\mathbf{s}_{t+2:T}} \sigma(\mathbf{s}_{1:T}) = \sum_{s_{t+1}} \sigma(\mathbf{s}_{1:t+1}).$$

□

A.2. Proof of Twist-Induced Proposal

Proposition 3.3. (Twist-Induced Proposal). *For a given set of intermediate twisted targets $\pi_t(\mathbf{s}_{1:t})$ in Eq. (9), the proposal which minimizes the variance of the one-step incremental importance weights w_t is given by*

$$\begin{aligned}
q^\pi_t(s_t|\mathbf{s}_{1:t-1}) &\propto \frac{\pi_t(\mathbf{s}_{1:t})}{\pi_{t-1}(\mathbf{s}_{1:t-1})} \\
&= \frac{1}{Z^\pi_t(\mathbf{s}_{1:t-1})} p_0(s_t|\mathbf{s}_{1:t-1})\psi_t(\mathbf{s}_{1:t}).
\end{aligned} \tag{14}$$

Proof. We seek to minimize the variance of the resulting importance weights, subject to a constraint on the proposal probabilities summing to 1. Introducing a Lagrange multiplier $\lambda(\mathbf{s}_{1:t-1})$, we have

$$\min_{q(s_t|\mathbf{s}_{1:t-1})} \mathbb{E}_{q(s_t|\mathbf{s}_{1:t-1})}\left[\left(\frac{\tilde{\pi}_t(\mathbf{s}_{1:t})}{\tilde{\pi}_{t-1}(\mathbf{s}_{1:t-1})q(s_t|\mathbf{s}_{1:t-1})}\right)^2\right] - \left(\mathbb{E}_{q(s_t|\mathbf{s}_{1:t-1})}\left[\left(\frac{\tilde{\pi}_t(\mathbf{s}_{1:t})}{\tilde{\pi}_{t-1}(\mathbf{s}_{1:t-1})q(s_t|\mathbf{s}_{1:t-1})}\right)\right]\right)^2 + \lambda(\mathbf{s}_{1:t-1})\left(\textstyle\sum_{s_t} q(s_t|\mathbf{s}_{1:t-1}) - 1\right)$$

Taking $\frac{\delta}{\delta q}(\cdot) = 0$ implies

$$0 = \left(\frac{\tilde{\pi}_t(\mathbf{s}_{1:t})}{\tilde{\pi}_{t-1}(\mathbf{s}_{1:t-1})q(s_t|\mathbf{s}_{1:t-1})}\right)^2 - 2q(s_t|\mathbf{s}_{1:t-1})\left(\frac{\tilde{\pi}_t(\mathbf{s}_{1:t})}{\tilde{\pi}_{t-1}(\mathbf{s}_{1:t-1})q(s_t|\mathbf{s}_{1:t-1})}\right)\frac{\tilde{\pi}_t(\mathbf{s}_{1:t})}{\tilde{\pi}_{t-1}(\mathbf{s}_{1:t-1})q(s_t|\mathbf{s}_{1:t-1})^2} + \lambda(\mathbf{s}_{1:t-1})$$

where the derivative in the second term is zero since the $q(s_t|\mathbf{s}_{1:t-1})$ cancel. Finally, we have

$$\frac{\tilde{\pi}_t(\mathbf{s}_{1:t})^2}{\tilde{\pi}_{t-1}(\mathbf{s}_{1:t-1})^2 q(s_t|\mathbf{s}_{1:t-1})^2} = \lambda(\mathbf{s}_{1:t-1})$$

$$q^*(s_t|\mathbf{s}_{1:t-1}) = \frac{1}{\sqrt{\lambda(\mathbf{s}_{1:t-1})}} \frac{\tilde{\pi}_t(\mathbf{s}_{1:t})}{\tilde{\pi}_{t-1}(\mathbf{s}_{1:t-1})} = \frac{1}{Z_t^\pi(\mathbf{s}_{1:t-1})} p_0(s_t|\mathbf{s}_{1:t-1})\psi_t(\mathbf{s}_{1:t})$$

where $Z_t^\pi(\mathbf{s}_{1:t-1})$ (or λ) is chosen to enforce normalization. □

We focused on the one-step twist-induced proposal in Prop. 3.3. However, this proposal is *not optimal* for resampling every c steps (as would also occur, for example, with adaptive resampling).

Proposition A.5 (Multi-Step Twist Induced Proposal (Generalization of Prop. 3.3)). *For resampling c-steps ahead, the optimal proposal (over $\mathbf{s}_{t+1:t+c-1}$) which minimizes the variance of the importance weights $w_{t:t+c-1}(\mathbf{s}_{1:t+c-1})$ is given by*

$$q^\pi(\mathbf{s}_{t:t+c-1}|\mathbf{s}_{1:t-1}) = \frac{p_0(\mathbf{s}_{t:t+c-1}|\mathbf{s}_{1:t-1})\psi_{t+c-1}(\mathbf{s}_{1:t+c-1})}{\sum_{\mathbf{s}_{t:t+c-1}} p_0(\mathbf{s}_{t:t+c-1}|\mathbf{s}_{1:t-1})\psi_{t+c-1}(\mathbf{s}_{1:t+c-1})}.$$

The proof follows the same reasoning as in the proof of Prop. 3.3 above, using the multistep weights $w_{t:t+c-1}(\mathbf{s}_{1:t+c-1}) = \frac{\tilde{\pi}_{t+c-1}(\mathbf{s}_{1:t+c-1})}{\tilde{\pi}_{t-1}(\mathbf{s}_{1:t-1})q(\mathbf{s}_{t:t+c-1}|\mathbf{s}_{1:t-1})}$ from Eq. (c-Step SMC Weights).

Note that the denominator is not usually tractable for $c > 1$ in language modeling applications.

A.3. Derivation of CTL Gradient

Lemma A.6 (Derivation of CTL Gradient). *For the* CTL *loss* $\min_{\boldsymbol{\theta}} \mathcal{L}_{CTL}(\boldsymbol{\theta}) := \min_{\boldsymbol{\theta}} \sum_{t=1}^T D_{\mathrm{KL}}\big(\sigma(\mathbf{s}_{1:t}) \,\big\|\, \pi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})\big)$, *the (negative) gradient with respect to the parameters $\boldsymbol{\theta}$ is given by*

$$-\nabla_{\boldsymbol{\theta}} \mathcal{L}_{CTL}(\boldsymbol{\theta}) = \sum_{t=1}^T \mathbb{E}_{\sigma(\mathbf{s}_{1:t})}\Big[\nabla_{\boldsymbol{\theta}} \log \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})\Big] - \mathbb{E}_{\pi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})}\Big[\nabla_{\boldsymbol{\theta}} \log \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})\Big] \tag{29}$$

Proof. Consider expanding the form of $\pi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})$ using Eq. (9), noting that the normalization $\log \mathcal{Z}_t^\psi$ is independent of $\mathbf{s}_{1:t}$. Taking the gradient with respect to $\boldsymbol{\theta}$ using the log derivative identity $\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) = f(\boldsymbol{\theta})\nabla_{\boldsymbol{\theta}} \log f(\boldsymbol{\theta})$, we have

$$\begin{aligned}
-\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathrm{CTL}}(\boldsymbol{\theta}) &= -\nabla_{\boldsymbol{\theta}} \left(\sum_{t=1}^T \mathbb{E}_{\sigma(\mathbf{s}_{1:t})}\Big[\log \sigma(\mathbf{s}_{1:t}) - \log p_0(\mathbf{s}_{1:t}) - \log \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})\Big] + \log \sum_{\mathbf{s}_{1:t}} p_0(\mathbf{s}_{1:t})\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}) \right) \\
&= \sum_{t=1}^T \mathbb{E}_{\sigma(\mathbf{s}_{1:t})}\Big[\nabla_{\boldsymbol{\theta}} \log \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})\Big] - \sum_{t=1}^T \sum_{\mathbf{s}_{1:t}} \frac{p_0(\mathbf{s}_{1:t})\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})}{\sum_{\mathbf{s}_{1:t}} p_0(\mathbf{s}_{1:t})\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})} \nabla_{\boldsymbol{\theta}} \Big(\log p_0(\mathbf{s}_{1:t}) + \log \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}) \Big) \\
&= \sum_{t=1}^T \Big(\mathbb{E}_{\sigma(\mathbf{s}_{1:t})}\Big[\nabla_{\boldsymbol{\theta}} \log \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})\Big] - \mathbb{E}_{\pi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})}\Big[\nabla_{\boldsymbol{\theta}} \log \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})\Big] \Big)
\end{aligned}$$

□

B. SMC with Intermediate Potentials and Connection with Soft Reinforcement Learning

In the main text, we focused on settings where the target distribution is defined by a potential $\phi(\mathbf{s}_{1:T})$ depending on full sequences only, as in Eq. (1). This setting highlights the need for (learned) twist functions to summarize the future expected value of the potential in the absence of intermediate target information.

In this appendix, we generalize our exposition to show how our twisted SMC framework can accommodate settings with intermediate potentials, which is evocative of connections with soft reinforcement learning (Levine, 2018). We leverage intuition from soft RL while introducing our general probabilistic interpretation, by using $\overset{\text{(sRL)}}{=}$ to instantiate the soft RL special case. In particular, soft RL will correspond to the terminal potential

$$\phi_t(\mathbf{s}_{1:t}) \overset{\text{(sRL)}}{=} e^{\beta\, r_t(\mathbf{s}_{1:t})} \qquad \text{(soft RL } \phi_t \text{ Definition)}$$

which corresponds to $\phi(\mathbf{s}_{1:T}) = e^{\beta r_T(\mathbf{s}_{1:T})}$ if the potential is given at the final step only (as in RLHF, Korbak et al. (2022b)). However, we defer detailed discussion of soft RL to App. B.3. See Table 4 for several examples of intermediate potentials.

Finally, we formalize a notion of conditional target distributions and twist functions in App. B.2, which generalizes the exposition in the main text and captures our conditional twist learning experiments in Sec. 7.2.3.

B.1. Twisted SMC with Intermediate Potentials

To generalize the exposition in the main text, we might consider defining the target as

$$\sigma(\mathbf{s}_{1:T}) \coloneqq \frac{1}{\mathcal{Z}_\sigma} p_0(\mathbf{s}_{1:T}) \left(\prod_{t=1}^{T} \phi_t(\mathbf{s}_{1:t}) \right) \overset{\text{(sRL)}}{=} \frac{1}{\mathcal{Z}_\sigma} p_0(\mathbf{s}_{1:T}) e^{\beta \sum_{t=1}^{T} r_t(\mathbf{s}_{1:t})} \tag{30}$$

where Eq. (1) and the main text exposition corresponds to $\phi_t(\mathbf{s}_{1:t}) = 1$ for $t < T$.

Optimal Twists with Intermediate Potentials Using Eq. (30), the marginal distribution $\sigma(\mathbf{s}_{1:t}) = \sum_{\mathbf{s}_{t+1:T}} \sigma(\mathbf{s}_{1:T})$ over t tokens becomes

$$\begin{aligned} \sigma(\mathbf{s}_{1:t}) =& \frac{1}{\mathcal{Z}_\sigma} p_0(\mathbf{s}_{1:t}) \left(\prod_{\tau=1}^{t} \phi_\tau(\mathbf{s}_{1:\tau}) \right) \left(\sum_{\mathbf{s}_{t+1:T}} p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t}) \prod_{\tau=t+1}^{T} \phi_\tau(\mathbf{s}_{1:\tau}) \right) \\ \overset{\text{(sRL)}}{=}& \frac{1}{\mathcal{Z}_\sigma} p_0(\mathbf{s}_{1:t}) e^{\beta \sum_{\tau=1}^{t} r_\tau(\mathbf{s}_{1:\tau})} \left(\sum_{\mathbf{s}_{t+1:T}} p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t}) e^{\beta \sum_{\tau=t+1}^{T} r_\tau(\mathbf{s}_{1:\tau})} \right) \end{aligned} \tag{31}$$

(soft RL special case)

As in Prop. 3.2, the goal of the optimal twist functions is to facilitate sampling from the intermediate marginals $\sigma(\mathbf{s}_{1:t})$ of the target distribution $\sigma(\mathbf{s}_{1:T})$.

We consider two different quantities involved in defining the optimal twists, which differ in their treatment of the intermediate reward. For the soft RL setting, this corresponds to the natural distinction between Q-values and (soft) value functions V_t.

$$\begin{aligned} \sigma(\mathbf{s}_{1:t}) =& \frac{1}{\mathcal{Z}_\sigma} p_0(\mathbf{s}_{1:t}) \left(\prod_{\tau=1}^{t-1} \phi_\tau(\mathbf{s}_{1:\tau}) \right) \underbrace{\phi_t(\mathbf{s}_{1:t}) \Big(\underbrace{\sum_{\mathbf{s}_{t+1:T}} p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t}) \prod_{\tau=t+1}^{T} \phi_\tau(\mathbf{s}_{1:\tau})}_{\Phi_t^*(\mathbf{s}_{1:t}) :\propto} \Big)}_{\psi_t^*(\mathbf{s}_{1:t}) :\propto} \\ \overset{\text{(sRL)}}{=}& \frac{1}{\mathcal{Z}_\sigma} p_0(\mathbf{s}_{1:t}) \left(e^{\beta \sum_{\tau=1}^{t-1} r_\tau(\mathbf{s}_{1:\tau})} \right) \underbrace{e^{\beta\, r_t(\mathbf{s}_{1:t})} \Big(\underbrace{\sum_{\mathbf{s}_{t+1:T}} p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t}) e^{\beta \sum_{\tau=t+1}^{T} r_\tau(\mathbf{s}_{1:\tau})}}_{\Phi_t^*(\mathbf{s}_{1:t}) :\propto e^{\beta V_t^*(\mathbf{s}_{1:t})} =} \Big)}_{\psi_t^*(\mathbf{s}_{1:t}) :\propto e^{\beta r_t(\mathbf{s}_{1:t}) + \beta V_t^*(\mathbf{s}_{1:t})} =} \end{aligned} \tag{32}$$

where $:\propto$ means 'defined to be proportional to' and $Q_t^*(s_t, \mathbf{s}_{1:t-1}) = r_t(\mathbf{s}_{1:t}) + V_t^*(\mathbf{s}_{1:t})$ in RL notation. See App. B.3 for detailed derivations in the soft RL special case. In general, Φ_t captures the expectation of *future* potentials from $t+1:T$, analogous to the (soft) value function. The twists ψ_t play a role analogous to a Q-value, estimating both the immediate ϕ_t and future value Φ_t. In particular,

$$\psi_t^*(\mathbf{s}_{1:t}) \propto \phi_t(\mathbf{s}_{1:t}) \Phi_t^*(\mathbf{s}_{1:t}) \qquad \text{where} \quad \Phi_t^*(\mathbf{s}_{1:t}) :\propto \sum_{\mathbf{s}_{t+1:T}} p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t}) \prod_{\tau=t+1}^{T} \phi_\tau(\mathbf{s}_{1:\tau}) \tag{33}$$

We continue to refer to ψ_t as the *twist functions* and focus on probabilistic interpretations based on ψ_t instead of Φ_t^* (see App. B.4 for additional discussion).

To show that this notation is consistent with the main text, consider the optimal twists $\psi_t^*(\mathbf{s}_{1:t}) = \phi_t(\mathbf{s}_{1:t}) \Phi_t^*(\mathbf{s}_{1:t})$ with no intermediate potentials, $\phi_t(\mathbf{s}_{1:t}) = 1$ for $t < T$. For $t < T$, $\psi_t^*(\mathbf{s}_{1:t}) = \Phi_t^*(\mathbf{s}_{1:t})$ reflect the future expected potential and for $t = T$, the terminal potential is $\psi_T^*(\mathbf{s}_{1:T}) = \phi_T(\mathbf{s}_{1:T})$, with no future potentials after step T, i.e. $\Phi_T = 1$.

Building on Eq. (31)-(32) above, the following generalization of Prop. 3.2 defines the 'optimal' twists so as to obtain the intermediate target marginals $\sigma(\mathbf{s}_{1:t})$ (see Prop. A.4).

Proposition B.1 (Optimal Twists). *For a given target distribution $\sigma(\mathbf{s}_{1:T})$ in Eq. (30), the optimal twist functions yield intermediate $\{\pi_t\}_{t=1}^{T-1}$ which match the target marginals. In regions where $p_0(\mathbf{s}_{1:t}) > 0$, the optimal twists are given by*

$$\pi_t^*(\mathbf{s}_{1:t}) = \sigma(\mathbf{s}_{1:t}) = \frac{1}{\mathcal{Z}_t^{\psi^*}} p_0(\mathbf{s}_{1:t}) \left(\prod_{\tau=1}^{t-1} \phi_\tau(\mathbf{s}_{1:\tau})\right) \psi_t^*(\mathbf{s}_{1:t}) \quad = \frac{1}{\mathcal{Z}_t^{\Phi^*}} p_0(\mathbf{s}_{1:t}) \left(\prod_{\tau=1}^{t-1} \phi_\tau(\mathbf{s}_{1:\tau})\right) \phi_t(\mathbf{s}_{1:t}) \Phi_t^*(\mathbf{s}_{1:t}). \quad (34)$$

Up to a constant c_t independent of $\mathbf{s}_{1:t}$, the optimal twists ψ_t^* are given by

$$\psi_t^*(\mathbf{s}_{1:t}) = c_t \; \phi_t(\mathbf{s}_{1:t}) \sum_{\mathbf{s}_{t+1:T}} p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t}) \prod_{\tau=t+1}^{T} \phi_\tau(\mathbf{s}_{1:\tau}) \quad (35)$$

where c_t is absorbed into the normalization constant $\mathcal{Z}_t^{\psi^*}$. The optimal twists satisfy the recursion

$$\psi_t^*(\mathbf{s}_{1:t}) = \frac{\mathcal{Z}_t^{\psi^*}}{\mathcal{Z}_{t+1}^{\psi^*}} \phi_t(\mathbf{s}_{1:t}) \sum_{s_{t+1}} p_0(s_{t+1}|\mathbf{s}_{1:t}) \psi_{t+1}^*(\mathbf{s}_{1:t+1}). \quad (36)$$

Proof. Substituting Eq. (35) into Eq. (34), we obtain the desired marginal Eq. (31),

$$\pi_t^*(\mathbf{s}_{1:t}) = \frac{c_t}{\mathcal{Z}_t^{\psi^*}} p_0(\mathbf{s}_{1:t}) \prod_{\tau=1}^{t} \phi_\tau(\mathbf{s}_{1:\tau}) \left(\sum_{\mathbf{s}_{t+1:T}} p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t}) \prod_{\tau=t+1}^{T} \phi_\tau(\mathbf{s}_{1:\tau}) \right) = \sigma(\mathbf{s}_{1:t})$$

where the final equality follows from absorbing the constant c_t into $\mathcal{Z}_t^{\psi^*}$, with $\frac{1}{\mathcal{Z}_\sigma} = \frac{c_t}{\mathcal{Z}_t^{\psi^*}}$ and $\mathcal{Z}_\sigma$ which normalizes $\tilde{\sigma}(\mathbf{s}_{1:t})$.

We will now use $c_t = \frac{\mathcal{Z}_t^{\psi^*}}{\mathcal{Z}_\sigma}$ to show the recursion in Eq. (36). Note that Eq. (35) implies

$$\psi_t^*(\mathbf{s}_{1:t}) = c_t \; \phi_t(\mathbf{s}_{1:t}) \sum_{s_{t+1}} p_0(s_{t+1}|\mathbf{s}_{1:t}) \underbrace{\left(\phi_{t+1}(\mathbf{s}_{1:t+1}) \sum_{\mathbf{s}_{t+2:T}} p_0(\mathbf{s}_{t+2:T}|\mathbf{s}_{1:t+1}) \prod_{\tau=t+2}^{T} \phi_\tau(\mathbf{s}_{1:\tau}) \right)}_{\frac{1}{c_{t+1}} \psi_{t+1}^*(\mathbf{s}_{1:t+1})}$$

$$= \frac{\mathcal{Z}_t^{\psi^*}}{\mathcal{Z}_{t+1}^{\psi^*}} \phi_t(\mathbf{s}_{1:t}) \sum_{s_{t+1}} p_0(s_{t+1}|\mathbf{s}_{1:t}) \psi_{t+1}^*(\mathbf{s}_{1:t+1})$$

where the second line follows from $\frac{c_t}{c_{t+1}} = \frac{\mathcal{Z}_t^{\psi^*}/\mathcal{Z}_\sigma}{\mathcal{Z}_{t+1}^{\psi^*}/\mathcal{Z}_\sigma}$. This demonstrates Eq. (36). □

Remark B.2 (Equivalence Class of ψ_t and Φ_t). Note that any rescaling of $\psi_t \leftarrow c_t \bar{\psi}_t$ by a constant with respect to $\mathbf{s}_{1:t}$ will yield the same intermediate marginals $\pi_t(\mathbf{s}_{1:t})$, due to the normalization constant $\mathcal{Z}_t^{\psi}$ which scales with ψ_t. This defines an equivalence class in the space of functions. The same statement holds for Φ_t. We express results such as Eq. (35) using proportionality $\propto$. We define ψ_t and Φ_t as particular members of their equivalence classes whose normalization $\mathcal{Z}_t^{\psi}$ and $\mathcal{Z}_t^{\Phi}$ are equal, such that $\psi_t(\mathbf{s}_{1:t}) = \phi_t(\mathbf{s}_{1:t}) \Phi_t(\mathbf{s}_{1:t})$.

This leads to the following definition of the intermediate twisting targets (we defer the soft RL special case to App. B.3).

Definition B.3 (Twisted Intermediate Targets). *Using approximate twist functions $\{\psi_t\}_{t=1}^{T-1}$, we define the twisted intermediate target distributions*

$$\pi_t(\mathbf{s}_{1:t}) = \begin{cases} \frac{1}{\mathcal{Z}_t^{\psi}} p_0(\mathbf{s}_{1:t}) \left(\prod_{\tau=1}^{t-1} \phi_\tau(\mathbf{s}_{1:\tau})\right) \psi_t(\mathbf{s}_{1:t}) & (t < T) \\ \frac{1}{\mathcal{Z}_\sigma} p_0(\mathbf{s}_{1:T}) \prod_{t=1}^{T} \phi_t(\mathbf{s}_{1:t}) & (t = T) \end{cases} \quad \text{(Twist Targets } (\psi)\text{)}$$

One-Step Twist-Induced Proposal Using Prop. 3.3 and Def. B.3 and noting that $\phi_{t-1}(\mathbf{s}_{1:t-1})$ is independent of s_t, we have the optimal one-step proposal

$$\begin{aligned} q_t^\pi(s_t|\mathbf{s}_{1:t-1}) \propto \frac{\pi_t(\mathbf{s}_{1:t})}{\pi_{t-1}(\mathbf{s}_{1:t-1})} &= \frac{\mathcal{Z}_{t-1}^\psi}{\mathcal{Z}_t^\psi} p_0(s_t|\mathbf{s}_{1:t-1}) \frac{\phi_{t-1}(\mathbf{s}_{1:t-1})\psi_t(\mathbf{s}_{1:t})}{\psi_{t-1}(\mathbf{s}_{1:t-1})} \\ &=: \frac{1}{Z_t^\pi(\mathbf{s}_{1:t-1})} p_0(s_t|\mathbf{s}_{1:t-1})\psi_t(\mathbf{s}_{1:t}) \qquad \text{(Twist-Induced Proposal } (\psi)\text{)} \\ &= \frac{p_0(s_t|\mathbf{s}_{1:t-1})\psi_t(\mathbf{s}_{1:t})}{\sum_{s_t} p_0(s_t|\mathbf{s}_{1:t-1})\psi_t(\mathbf{s}_{1:t})} \end{aligned}$$

where in the second line, we absorb terms which depend only on $\mathbf{s}_{1:t-1}$ (and not s_t) into the normalization. In the soft RL special case, we have $q_t^\pi(s_t|\mathbf{s}_{1:t-1}) \propto p_0(s_t|\mathbf{s}_{1:t-1})e^{\beta Q_t(s_t, \mathbf{s}_{1:t-1})}$ (see Eq. (Twist-Induced Proposal (soft RL)) below).

B.2. Conditional Twisted SMC

To formalize our notion of conditional twists in the infilling experiments (Sec. 7.2.3), we generalize our above framework to explicitly depend on 'observation' random variables $\{o_t\}_{t=1}^T$. This matches the common setting of SMC in state-space models (Briers et al., 2010; Gu et al., 2015; Lawson et al., 2022; Chopin et al., 2020). Our derivations in this section also emphasize that the optimal twist functions in Prop. B.1 learn functions proportional to *conditional likelihoods* of the future observation variables given the current sequence (see Eq. (39) below)). We recover the unconditional targets in the main text for fixed $o_T = 1$.

Consider a target distribution $\sigma(\mathbf{s}_{1:T}|\boldsymbol{o}_{1:T})$ conditioned on particular observation random variables $\boldsymbol{o}_{1:T} \coloneqq \{o_t\}_{t=1}^T$. We define a probabilistic model over observations $\sigma(o_t|\mathbf{s}_{1:t}) = \phi_t(o_t, \mathbf{s}_{1:t})$ as the intermediate potential,[3] which yields the target posterior

$$\sigma(\mathbf{s}_{1:T}|\boldsymbol{o}_{1:T}) = \frac{p_0(\mathbf{s}_{1:T})\left(\prod_{t=1}^T \sigma(o_t|\mathbf{s}_{1:t})\right)}{\sum_{\mathbf{s}_{1:T}} p_0(\mathbf{s}_{1:T})\left(\prod_{t=1}^T \sigma(o_t|\mathbf{s}_{1:t})\right)} = \frac{1}{\mathcal{Z}_\sigma(\boldsymbol{o}_{1:T})} p_0(\mathbf{s}_{1:T})\left(\prod_{t=1}^T \phi_t(o_t, \mathbf{s}_{1:t})\right) = \frac{p_0(\mathbf{s}_{1:T})\sigma(\boldsymbol{o}_{1:T}|\mathbf{s}_{1:T})}{\sigma(\boldsymbol{o}_{1:T})} \tag{37}$$

where we interpret $\sigma(\boldsymbol{o}_{1:T}|\mathbf{s}_{1:T}) = \prod_{t=1}^T \sigma(o_t|\mathbf{s}_{1:t})$ and $\mathcal{Z}_\sigma(\boldsymbol{o}_{1:T}) = \sigma(\mathbf{s}_{1:T})$ to make the Bayesian posterior explicit in the last equality. Note, we now seek to estimate a different partition function $\mathcal{Z}_\sigma(\boldsymbol{o}_{1:T})$ for each set of observation variables.

Using our infilling experiments in Sec. 7.2.3 as an example, consider (a sequence of) subsequent tokens $o_T = \mathbf{s}_{T+1:T+c}$ as observation variables, where the observation model is simply the base language model $\sigma(o_T|\mathbf{s}_{1:T}) \coloneqq p_0(\mathbf{s}_{T+1:T+c}|\mathbf{s}_{1:T})$.

Using Eq. (37), the intermediate marginals become

$$\begin{aligned} \sigma(\mathbf{s}_{1:t}|\boldsymbol{o}_{1:T}) &= \sum_{\mathbf{s}_{t+1:T}} \sigma(\mathbf{s}_{1:T}|\boldsymbol{o}_{1:T}) \\ &= \sum_{\mathbf{s}_{t+1:T}} \frac{1}{\sigma(\boldsymbol{o}_{1:T})} p_0(\mathbf{s}_{1:t}) p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t}) \Big(\prod_{t=1}^T \sigma(o_t|\mathbf{s}_{1:t})\Big) \\ &= \frac{1}{\mathcal{Z}_\sigma(\boldsymbol{o}_{1:T})} p_0(\mathbf{s}_{1:t}) \left(\prod_{\tau=1}^t \phi_\tau(o_\tau, \mathbf{s}_{1:\tau})\right) \sum_{\mathbf{s}_{t+1:T}} p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t}) \Big(\prod_{\tau=t+1}^T \phi_\tau(o_\tau, \mathbf{s}_{1:\tau})\Big) \\ &= \frac{1}{\mathcal{Z}_\sigma(\boldsymbol{o}_{1:T})} p_0(\mathbf{s}_{1:t}) \left(\prod_{\tau=1}^t \phi_\tau(o_\tau, \mathbf{s}_{1:\tau})\right) \sigma(\boldsymbol{o}_{t+1:T}|\mathbf{s}_{1:t}) \,, \end{aligned} \tag{38}$$

noting that $\sigma(\boldsymbol{o}_{t+1:T}|\mathbf{s}_{1:t}) = \sum_{\mathbf{s}_{t+1:T}} \sigma(\boldsymbol{o}_{t+1:T}, \mathbf{s}_{t+1:T}|\mathbf{s}_{1:t})$ matches the second to last line.

[3] Note, rescaling $\phi_t(\mathbf{s}_{1:t}, o_t = 1)$ by a constant c with respect to $o_t, \mathbf{s}_{1:t}$ does not affect the target posterior in Eq. (37). For example, with terminal potential only: $\sigma(\mathbf{s}_{1:T}|o_T) = \frac{p_0(\mathbf{s}_{1:T})\ \phi_T(\mathbf{s}_{1:T}, o_T)/c}{\sum_{\mathbf{s}_{1:T}} p_0(\mathbf{s}_{1:T})\ \phi_T(\mathbf{s}_{1:T}, o_T)/c} = \frac{1}{\mathcal{Z}_\sigma(o_T)} p_0(\mathbf{s}_{1:T})\phi_T(\mathbf{s}_{1:T}, o_T)$ as long as the scaling factor is independent of o_T and $\mathbf{s}_{1:T}$.

The optimal twists take a similar form as Prop. B.1, but now as a function of the future observation or conditioning information. Further, the optimal twists is proportional to the conditional likelihoods (e.g., $\sigma(\boldsymbol{o}_{t+1:T}|\mathbf{s}_{1:t})$) of future observations given $\mathbf{s}_{1:t}$, which marginalize over future tokens (e.g., $\mathbf{s}_{t+1:T}$),

$$\begin{aligned}\Phi_t^*(\mathbf{s}_{1:t}, \boldsymbol{o}_{t+1:T}) &\overset{\boldsymbol{o}_{t+1:T}}{\propto} \sigma(\boldsymbol{o}_{t+1:T}|\mathbf{s}_{1:t}) = \sum_{\mathbf{s}_{t+1:T}} p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t})\Big(\prod_{\tau=t+1}^{T} \phi_\tau(o_\tau, \mathbf{s}_{1:\tau})\Big)\,, \\ \psi_t^*(\mathbf{s}_{1:t}, \boldsymbol{o}_{t:T}) &\overset{\boldsymbol{o}_{t:T}}{\propto} \sigma(\boldsymbol{o}_{t:T}|\mathbf{s}_{1:t}) = \sum_{\mathbf{s}_{t+1:T}} p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t})\Big(\prod_{\tau=t}^{T} \phi_\tau(o_\tau, \mathbf{s}_{1:\tau})\Big)\,,\end{aligned} \tag{39}$$

where $f(x, \boldsymbol{o}) \overset{\boldsymbol{o}}{\propto} g(x, \boldsymbol{o})$ denotes proportionality up to a constant which depends on $\boldsymbol{o}$ only: $\exists c(\boldsymbol{o})\colon f(x, \boldsymbol{o}) = c(\boldsymbol{o})g(x, \boldsymbol{o})$. These equations can be confirmed by comparing Prop. B.1 with the last two lines in Eq. (38).

The intermediate marginals over partial sequences can finally be rewritten as either

$$\begin{aligned}\sigma(\mathbf{s}_{1:t}|\boldsymbol{o}_{1:T}) &\overset{\boldsymbol{o}_{1:T}}{\propto} p_0(\mathbf{s}_{1:t})\left(\prod_{\tau=1}^{t} \phi_\tau(o_\tau, \mathbf{s}_{1:\tau})\right)\Phi_t^*(\mathbf{s}_{1:t}, \boldsymbol{o}_{t+1:T})\,, \\ &= p_0(\mathbf{s}_{1:t})\left(\prod_{t=1}^{t-1} \phi_\tau(o_\tau, \mathbf{s}_{1:\tau})\right)\psi_t^*(\mathbf{s}_{1:t}, \boldsymbol{o}_{t:T})\,.\end{aligned} \tag{40}$$

We discuss the choice of parameterization using ψ_t versus Φ_t in App. B.4.

The conditional twist learning formulation matches the setting of Lawson et al. (2022), to which we refer the reader for additional discussion. We use this conditional perspective to derive classification losses for twist learning in App. C.3-C.4.

Unconditional Targets as a Special Case In cases where we are only learning twists for a single set of conditioning information such as a single classifier label or a reward model, note that we can omit explicit conditioning information in $\psi_t(\mathbf{s}_{1:t}, o_t)$ and consider setting $\{o_t = 1\}_{t=1}^T$. With terminal potential only as in the main text, we write $\sigma(o_T = 1|\mathbf{s}_{1:T}) = \phi(\mathbf{s}_{1:T})$ and the overall target distribution as $\sigma(\mathbf{s}_{1:T}) = \sigma(\mathbf{s}_{1:T}|o_T = 1) \propto p_0(\mathbf{s}_{1:T})\phi_T(\mathbf{s}_{1:T})$.

To given meaning to this probabilistic interpretation of $\sigma(o_T = 1|\mathbf{s}_{1:T}) = \phi(\mathbf{s}_{1:T})$ with a binary random variable o_T, note that we need to ensure $\phi(\mathbf{s}_{1:T}) \in [0, 1]$. As a result, sampling from the target $\sigma(\mathbf{s}_{1:T}|o_T = 1)$ or joint $\sigma(\mathbf{s}_{1:T}, o_T = 1)$ is no easier in this interpretation than in Eq. (1), which is intractable in general. For example, finding $\phi_{\max} = \max_{\mathbf{s}_{1:T}} \phi(\mathbf{s}_{1:T})$ and dividing $\phi(\mathbf{s}_{1:T}) \leftarrow \phi(\mathbf{s}_{1:T})/\phi_{\max}$ to rescale $\sigma(o_T = 1|\mathbf{s}_{1:T})$ is equivalent to being able to perform rejection sampling with the base model proposal $p_0(\mathbf{s}_{1:T})$ (see Sec. 4.1.2).

With this caveat in mind, the formulation in Eq. (37)-Eq. (39) strictly generalizes our exposition in the main text and App. B.1. With intermediate potentials, we set $\sigma(o_{1:T} = \mathbf{1}|\mathbf{s}_{1:T}) = \prod_{t=1}^T \phi_t(\mathbf{s}_{1:t})$.

Our notation also matches the exposition in Levine (2018) for the soft RL case with a binary observation or 'optimality' random variable $\sigma(o_t = 1|\mathbf{s}_{1:t-1}, s_t) = e^{\beta r_t(\mathbf{s}_{1:t-1}, s_t)}$, where the reward is a function of the state $x_t = \mathbf{s}_{1:t-1}$ and action $a_t = s_t$ pair (see the MDP interpretation in App. B.3). Levine (2018) do not explicitly discuss the need to rescale the reward or its relation to rejection sampling.

B.3. Connection with Soft Reinforcement Learning

In this section, we more explicitly describe the soft reinforcement learning setting (Levine, 2018) commonly used in RLHF (Korbak et al., 2022b) as a special case of our probabilistic framework. Again, we use notation $\overset{\text{(sRL)}}{=}$ to indicate that the expressions in this section correspond to a particular instance of our SMC framework where $\phi(\mathbf{s}_{1:T}) = e^{\beta r(\mathbf{s}_{1:T})}$.

Summary of Soft RL Notation To summarize the below derivations, we state the relevant assignments for the soft RL case. We focus on the optimal case for simplicity, but note that approximate versions play identical roles,

$$\phi_t(\mathbf{s}_{1:t}) = e^{\beta\, r_t(\mathbf{s}_{1:t})} \qquad \psi_t^*(\mathbf{s}_{1:t}) = e^{\beta r_t(\mathbf{s}_{1:t}) + \beta V_t^*(\mathbf{s}_{1:t})} = e^{\beta Q_t^*(s_t, \mathbf{s}_{1:t-1})} \qquad \Phi_t^*(\mathbf{s}_{1:t}) = e^{\beta V_t^*(\mathbf{s}_{1:t})} \quad \text{(Twist to Soft RL)}$$

where $\psi_t^*(\mathbf{s}_{1:t}) = \phi_t(\mathbf{s}_{1:t})\Phi_t^*(\mathbf{s}_{1:t})$ or $Q_t^*(s_t, \mathbf{s}_{1:t-1}) = r_t(\mathbf{s}_{1:t}) + V_t^*(\mathbf{s}_{1:t})$. In the other direction, we have

$$r_t(\mathbf{s}_{1:t}) = \frac{1}{\beta}\log \phi_t(\mathbf{s}_{1:t}) \qquad Q_t^*(s_t, \mathbf{s}_{1:t-1}) = \frac{1}{\beta}\log \psi_t^*(\mathbf{s}_{1:t}) \qquad V_t^*(\mathbf{s}_{1:t}) = \frac{1}{\beta}\log \Phi_t^*(\mathbf{s}_{1:t}) \qquad \text{(Soft RL to Twist)}$$

MDP Interpretation To draw connections with soft RL, we view language model controlled decoding as a Markov Decision Process (MDP), where the prompt is drawn from an initial state distribution $\mathbf{s}_0 \sim \nu_0$, an action policy $\pi(a_t|x_t) = q(s_t|\mathbf{s}_{1:t-1})$ selects the next token $a_t = s_t$ given a partial sequence $x_t = \mathbf{s}_{1:t-1}$ as the state, and the environment transitions are deterministic $P(x_{t+1} = \mathbf{s}_{1:t}|a_t = s_t, x_t = \mathbf{s}_{1:t-1}) = \delta(x_{t+1} = \text{concat}(s_t, \mathbf{s}_{1:t-1}))$ append the selected token to update the state. Discounting may also be included without difficulty. The reward is given by $r_t(\mathbf{s}_{1:t})$.

Final Target Distribution We define the target distribution as the solution to the following variational optimization which solves the regularized MDP described above,

$$\sigma(\mathbf{s}_{1:T}) \overset{\text{(sRL)}}{=} \frac{1}{\mathcal{Z}_\sigma} p_0(\mathbf{s}_{1:T}) e^{\beta \sum_{t=1}^{T} r_t(\mathbf{s}_{1:t})} = \arg\max_{q(\mathbf{s}_{1:T})} \mathbb{E}_{q(\mathbf{s}_{1:T})}\Big[\sum_{t=1}^{T} r_t(\mathbf{s}_{1:t})\Big] - \frac{1}{\beta} D_{\text{KL}}(q(\mathbf{s}_{1:T}) \,\|\, p_0(\mathbf{s}_{1:T})) \tag{41}$$

which corresponds to the choice $\phi_t(\mathbf{s}_{1:t}) = e^{\beta\, r_t(\mathbf{s}_{1:t})}$ as in Eq. (Twist to Soft RL). The soft value is defined as the maximum value of the above optimization for optimal $q^*(\mathbf{s}_{1:T})$, and corresponds to the scaled log partition function

$$V_0^*(\mathbf{s}_0) \coloneqq \frac{1}{\beta} \log \mathcal{Z}_\sigma = \frac{1}{\beta} \log \sum_{\mathbf{s}_{1:T}} p_0(\mathbf{s}_{1:T}) e^{\beta \sum_{t=1}^{T} r_t(\mathbf{s}_{1:t})} = \max_{q(\mathbf{s}_{1:T})} \mathbb{E}_{q(\mathbf{s}_{1:T})}\Big[\sum_{t=1}^{T} r_t(\mathbf{s}_{1:t})\Big] - \frac{1}{\beta} D_{\text{KL}}(q(\mathbf{s}_{1:T}) \,\|\, p_0(\mathbf{s}_{1:T})) \tag{42}$$

which can be confirmed by substituting $q(\mathbf{s}_{1:T}) = \sigma(\mathbf{s}_{1:T})$ from Eq. (41) into the maximization on the right side of Eq. (42). Although we omit the dependence of $\mathcal{Z}_\sigma(\mathbf{s}_0)$ on the prompt $\mathbf{s}_0$ for notational simplicity (see Eq. (1)), note that $V_0^* \coloneqq V^*(\mathbf{s}_0)$ naturally corresponds to the soft value of the prompt as the initial state in the MDP.

Optimal Intermediate Marginals and Soft Value Decomposing the maximization in Eq. (42) into optimizations over each $q(s_{t+1}|\mathbf{s}_{1:t})$, we define the intermediate soft value $V_t^*(\mathbf{s}_{1:t})$ as the maximum of the expected future regularized reward

$$V_t^*(\mathbf{s}_{1:t}) = \frac{1}{\beta} \log \Phi_t^*(\mathbf{s}_{1:t}) \overset{\text{(sRL)}}{=} \frac{1}{\beta} \log \sum_{\mathbf{s}_{t+1:T}} p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t})\, e^{\beta \sum_{\tau=t+1}^{T} r_\tau(s_{1:\tau})} \qquad \text{(Optimal Intermediate Soft Value)}$$

$$= \max_{q(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t})} \mathbb{E}_{q(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t})}\Big[\sum_{\tau=t+1}^{T} r_\tau(\mathbf{s}_{1:\tau})\Big] - \frac{1}{\beta} D_{\text{KL}}(q(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t}) \,\|\, p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t}))$$

$$= \max_{q(s_{t+1}|\mathbf{s}_{1:t})} \mathbb{E}_{q(s_{t+1}|\mathbf{s}_{1:t})}\Big[r_{t+1}(\mathbf{s}_{1:t+1}) + V_{t+1}^*(\mathbf{s}_{1:t+1})\Big] - \frac{1}{\beta} D_{\text{KL}}(q(s_{t+1}|\mathbf{s}_{1:t}) \,\|\, p_0(s_{t+1}|\mathbf{s}_{1:t}))$$

where, in the third line, we isolate the optimization over $q(s_t|\mathbf{s}_{1:t-1})$ by (i) assuming optimality at $\tau < t$ and (ii) substituting the optimal value $V_{t+1}^*(\mathbf{s}_{1:t+1}) = \max_{q(\mathbf{s}_{t+2:T}|\mathbf{s}_{1:t+1})}[...]$ of the maximization over $q(\mathbf{s}_{t+2:T}|\mathbf{s}_{1:t+1})$ (i.e. recursively applying the second line).

The optimal intermediate marginal can be written using either $V_t^*(\mathbf{s}_{1:t})$ or $Q_t^*(s_t, \mathbf{s}_{1:t-1})$ form (as in Eq. (32) above, or by substituting the optimal V_t^* or Q_t^* into the twist targets below).

Finally, note that twist consistency condition in Prop. 3.2 Eq. (13) or B.1 Eq. (36) implies

$$V_t^*(\mathbf{s}_{1:t}) = \frac{1}{\beta} \log \sum_{s_{t+1}} p_0(s_{t+1}|\mathbf{s}_{1:t})\, e^{\beta r_{t+1}(\mathbf{s}_{1:t+1}) + \beta V_{t+1}^*(\mathbf{s}_{1:t+1})} = \frac{1}{\beta} \log \sum_{s_{t+1}} p_0(s_{t+1}|\mathbf{s}_{1:t})\, e^{\beta Q_{t+1}^*(s_{t+1}, \mathbf{s}_{1:t})} \tag{43}$$

which can also be confirmed using Eq. (Optimal Intermediate Soft Value).

Twisted Intermediate Targets We state the approximate twisting targets for *both* V_t or Q_t parameterizations in order to make connections with soft RL losses in App. C. For approximate $V_t(\mathbf{s}_{1:t})$ or $Q_t(s_t, \mathbf{s}_{1:t-1})$, we have

$$\pi_t(\mathbf{s}_{1:t}) \overset{\text{(sRL)}}{=} \frac{1}{\mathcal{Z}_t^V} p_0(\mathbf{s}_{1:t}) e^{\beta \sum_{\tau=1}^{t-1} r_\tau(\mathbf{s}_{1:\tau})} e^{\beta r_t(\mathbf{s}_{1:t}) + \beta V_t(\mathbf{s}_{1:t})} \qquad (t < T) \qquad \text{(Twist Targets (Soft RL V))}$$

$$= \frac{1}{\mathcal{Z}_t^Q} p_0(\mathbf{s}_{1:t}) e^{\beta \sum_{\tau=1}^{t-1} r_\tau(\mathbf{s}_{1:\tau})} e^{\beta Q_t(s_t, \mathbf{s}_{1:t-1})} \qquad (t < T) \qquad \text{(Twist Targets (Soft RL Q))}$$

where the final twisting target is given by Eq. (41) and the optimal Q-values are defined as

$$Q_t^*(s_t, \mathbf{s}_{1:t-1}) = r_t(\mathbf{s}_{1:t}) + V_t^*(\mathbf{s}_{1:t}) \tag{44}$$

One-Step Proposal Finally, the optimal one-step proposal (e.g. in V_t form) can be derived either (i) as the twist-induced proposal from Eq. (Twist Targets (Soft RL V)) and Prop. B.1 or (ii) as the solution to the one-step optimization in the third line of Eq. (Optimal Intermediate Soft Value). As in Eq. (Twist-Induced Proposal (ψ)),

$$q_t^\pi(s_t|\mathbf{s}_{1:t-1}) \overset{\text{(sRL)}}{=} \frac{p_0(s_t|\mathbf{s}_{1:t-1})e^{\beta(r_t(\mathbf{s}_{1:t})+V_t(\mathbf{s}_{1:t}))}}{\sum_{s_t} p_0(s_t|\mathbf{s}_{1:t-1})e^{\beta(r_t(\mathbf{s}_{1:t})+V_t(\mathbf{s}_{1:t}))}} \propto p_0(s_t|\mathbf{s}_{1:t-1})e^{\beta Q_t(s_t,\mathbf{s}_{1:t-1})}.$$

(Twist-Induced Proposal (soft RL))

We define the one-step log normalization constant induced by an approximate V_t or Q_t as V_{V_t} or V_{Q_t}, respectively,

$$V_{V_t}(\mathbf{s}_{1:t-1}) \coloneqq \frac{1}{\beta}\log\sum_{s_t} p_0(s_t|\mathbf{s}_{1:t-1})e^{\beta(r_t(\mathbf{s}_{1:t})+V_t(\mathbf{s}_{1:t}))} \qquad V_{Q_t}(\mathbf{s}_{1:t-1}) \coloneqq \frac{1}{\beta}\log\sum_{s_t} p_0(s_t|\mathbf{s}_{1:t-1})e^{\beta Q_t(s_t,\mathbf{s}_{1:t-1})} \tag{45}$$

such that, for example, $q_t^\pi(s_t|\mathbf{s}_{1:t-1}) = p_0(s_t|\mathbf{s}_{1:t-1})e^{\beta Q_t(s_t,\mathbf{s}_{1:t-1})-\beta V_{Q_t}(\mathbf{s}_{1:t-1})}$. Note that, by the twist consistency condition in Eq. (43) or Prop. 3.2 and B.1, at optimality we have $V^*(\mathbf{s}_{1:t-1}) = V_{V_t^*}(\mathbf{s}_{1:t-1}) = V_{Q_t^*}(\mathbf{s}_{1:t-1})$.

RLHF Minimizes $D_{\text{KL}}(q \,\|\, \sigma)$ Note that, for a given suboptimal $q(\mathbf{s}_{1:T})$, the value of the variational optimization in Eq. (41) is a lower bound on the (scaled) log partition function $V_0^* = \frac{1}{\beta}\log\mathcal{Z}_\sigma$. Similarly to the standard Evidence Lower Bound, the gap in this lower bound is given by the KL divergence

$$\frac{1}{\beta}\log\mathcal{Z}_\sigma = \underbrace{\frac{1}{\beta}D_{\text{KL}}(q(\mathbf{s}_{1:T}) \,\|\, \sigma(\mathbf{s}_{1:T}))}_{\text{ELBO gap } (\geq 0)} + \underbrace{\left(\mathbb{E}_{q(\mathbf{s}_{1:T})}\left[\sum_{t=1}^T r_t(\mathbf{s}_{1:t})\right] - \frac{1}{\beta}D_{\text{KL}}(q(\mathbf{s}_{1:T}) \,\|\, p_0(\mathbf{s}_{1:T}))\right)}_{\text{'ELBO': Eq. (41)}} \tag{46}$$

In this sense, we consider soft RL or policy gradient methods such as PPO which optimize Eq. (41) as targeting $\sigma(\mathbf{s}_{1:T})$ by minimizing $D_{\text{KL}}(q(\mathbf{s}_{1:T}) \,\|\, \sigma(\mathbf{s}_{1:T}))$ (Korbak et al., 2022b).

B.4. Remarks on Parameterization

While the twisting targets (Eq. (Twist Targets (ψ))) and twist-induced proposal (Eq. (Twist-Induced Proposal (ψ))) may equivalently be parameterized using approximate Φ_t, we focus on the ψ_t parameterization to match the main text. In particular, recall that the optimal twists satisfy $\psi_t^*(\mathbf{s}_{1:t}) = \phi_t(\mathbf{s}_{1:t})\Phi_t^*(\mathbf{s}_{1:t})$ for all t. With no intermediate potential ($\phi_t = 1$ for $t < T$), our approximate twists estimate $\psi_t(\mathbf{s}_{1:t}) \approx \Phi_t^*(\mathbf{s}_{1:t}) \propto \sum_{\mathbf{s}_{t+1:T}} p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t})\phi_T(\mathbf{s}_{1:T})$ for $t < T$. In this section, we describe how the presence of intermediate potentials may affect the choice of twist parameterization.

The twist-induced proposal may not be tractable to evaluate at the final timestep, since it may be costly to evaluate the terminal potential $\phi_T(\mathbf{s}_{1:T})$ for all $s_T \in \mathcal{V}$ given a context $\mathbf{s}_{1:T-1}$ (as described in Sec. 3.2). Thus, we learn an approximate $\psi_T(\mathbf{s}_{1:T}) \approx \phi_T(\mathbf{s}_{1:T})$ for proposal sampling, which can be easily evaluated over $|\mathcal{V}|$ next tokens. The final $\pi_T(\mathbf{s}_{1:T}) = \sigma(\mathbf{s}_{1:T})$ is defined using $\phi(\mathbf{s}_{1:T})$ in order to preserve unbiased estimation. However, after sampling the proposal according to ψ_T, we only need to evaluate $\phi(\mathbf{s}_{1:T})$ over K full sequences to calculate the importance weights at the final step (Eq. (16)). See *Intermediate Potential Tractable over K Sequences Only* paragraph below.

Intermediate Potentials Tractable over $|\mathcal{V}|$ Sequences However, in settings where the intermediate potentials $\phi_t(\mathbf{s}_{1:t})$ *are* tractable to calculate for all $s_t \in \mathcal{V}$ given $\mathbf{s}_{1:t-1}$ (e.g. using an indicator function or forward pass in a transformer architecture, as in Table 4), it may be useful to use a Φ_t parameterization of the twist targets and twist-induced proposal. This allows us to use the *exact* immediate potentials $\phi_t(\mathbf{s}_{1:t})$ alongside an estimated $\Phi_t^{\boldsymbol{\theta}}$, instead of an approximate $\psi_t^{\boldsymbol{\theta}} \approx \phi_t\Phi_t^*$ which estimates both the immediate ϕ_t and future expected value of potentials Φ_t^*. Using notation established in Eq. (32) and Prop. B.1, the twisting targets in Eq. (Twist Targets (ψ)) can be rewritten using a $\Phi_t^{\boldsymbol{\theta}}$ parameterization

$$\pi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}) = \frac{1}{\mathcal{Z}_t^\psi}\, p_0(\mathbf{s}_{1:t})\left(\prod_{\tau=1}^{t-1}\phi_\tau(\mathbf{s}_{1:\tau})\right)\phi_t(\mathbf{s}_{1:t})\Phi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}) \qquad (t < T) \tag{Twist Targets (Φ))}$$

with $\pi_T(\mathbf{s}_{1:T}) = \sigma(\mathbf{s}_{1:T})$ as before. The twist-induced proposal $q_t^\pi(s_t|\mathbf{s}_{1:t-1}) \propto p_0(s_t|\mathbf{s}_{1:t-1})\phi_t(\mathbf{s}_{1:t})\Phi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})$ and its normalization constant are tractable in this case, by evaluating both the given $\phi_t(\mathbf{s}_{1:t})$ and parameterized $\Phi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})$ in a single forward pass and normalizing over the discrete vocabulary of next tokens.

Intermediate Potentials Tractable over K Sequences Only In cases where the intermediate potentials are difficult to evaluate, we would like to limit evaluation of $\phi_t(\mathbf{s}_{1:t})$ to only K partial sequences. In this case, parameterizing the twisted targets π_t using $\psi_t^{\boldsymbol{\theta}}$ or $Q_t^{\boldsymbol{\theta}}$ (Eq. (Twist Targets (ψ)), Eq. (Twist Targets (Soft RL Q))) instead of $\Phi_t^{\boldsymbol{\theta}}$ or $V_t^{\boldsymbol{\theta}}$ may be preferable to ensure a tractable twist-induced proposal. Separate parameterizations of the proposal (using $\psi_t^{\boldsymbol{\xi}}$) and targets ($\phi_t \Phi_t^{\boldsymbol{\theta}}$) might also be considered.

In the case of the final timestep described above or in Sec. 3.2, note that we use a learned $\psi_T^{\boldsymbol{\xi}}$ to parameterize a tractable variational proposal $q_T(s_T|\mathbf{s}_{1:T-1})$. In this case, we have no future value $\Phi_T(\mathbf{s}_{1:T}) = 1$ and only need to evaluate the terminal potential $\phi(\mathbf{s}_{1:T})$ for calculating importance weights over K sequences.

C. Twist Learning Losses

In this section, we describe various methods for twist learning beyond our proposed contrastive twist learning (CTL) procedure from Sec. 4. In App. C.1, we first describe several losses from the soft RL literature from a probabilistic perspective, building closely on our developments in App. B.1. We then proceed to describe SIXO (Lawson et al., 2022) and FUDGE (Yang & Klein, 2021) in App. C.3-C.4.

We emphasize losses found in related work or used as experimental baselines using equation tags (e.g. Eq. (SIXO)), where equations Eq. (RL Baseline), Eq. (SIXO), Eq. (FUDGE) are used in our experiments. We consider settings with intermediate potentials in App. C.1, but focus on the ($\phi_t = 1$ for $t < T$) setting in the remainder of the section, as in the main text.

C.1. Soft Q-Learning (RL) and Path Consistency Losses from Log Importance Weights

From the probabilistic perspective of the SMC log importance weights, we can derive several losses for twist learning, including soft Q-learning and path consistency learning (PCL) (Nachum et al., 2017) losses from the soft RL literature.

A general principle for deriving loss functions would be to minimize the variance of the (log) importance weights under some sampling distribution π_s, which leads to constant importance weights at optimality. To draw connections with previous work, we also consider minimizing the square of the log weights (as in, for example, Scharth & Kohn (2016)), which at optimality, ensures that $\log w = 0$ and $w = 1$ are equal to a *particular* constant. We will proceed to parameterize the twist functions using parameters $\boldsymbol{\theta}$ and consider loss terms which minimize the variance or square of c-step log weights at time t,

$$\mathcal{L}_{\text{log Var}}^{(t,c)}(\boldsymbol{\theta}) \coloneqq \text{Var}_{\pi_s}\left[\sum_{\tau=t}^{t+c-1} \log w_\tau(\mathbf{s}_{1:\tau})\right] \qquad \mathcal{L}_{\text{log Cons}}^{(t,c)}(\boldsymbol{\theta}) \coloneqq \mathbb{E}_{\pi_s}\left[\left(\sum_{\tau=t}^{t+c-1} \log w_\tau(\mathbf{s}_{1:\tau})\right)^2\right]. \tag{47}$$

$\mathcal{L}_{\text{log Cons}}^{(t,c)}(\boldsymbol{\theta})$ indicates 'consistency' in log-weight space for c-step-ahead weights at time t (see Eq. (c-Step SMC Weights)).

We will consider various choices of parameterization and proposal in the following subsections. For example, let $\mathcal{L}_{\text{log Cons}}^{(t,c)}(\boldsymbol{\theta}; \{\psi_t, q_t^\pi\})$ denote the log-consistency loss corresponding to twisting targets parameterized by $\psi_t^{\boldsymbol{\theta}}$ and the twist induced proposal q_t^π (note, our notation for the one-step weights $w_t(\mathbf{s}_{1:t})$ does not make these choices explicit).

For reference, we derive the log importance weights with intermediate potentials and arbitrary q as

$$\log w_t(\mathbf{s}_{1:t}) = \log \frac{\tilde{\pi}_t(\mathbf{s}_{1:t})}{\tilde{\pi}_{t-1}(\mathbf{s}_{1:t-1}) q(s_t|\mathbf{s}_{1:t-1})} = \log \frac{p_0(\mathbf{s}_{1:t})\left(\prod_{\tau=1}^{t-1} \phi_\tau(\mathbf{s}_{1:\tau})\right)\psi_t(\mathbf{s}_{1:t})}{p_0(\mathbf{s}_{1:t-1})\left(\prod_{\tau=1}^{t-2} \phi_\tau(\mathbf{s}_{1:\tau})\right)\psi_{t-1}(\mathbf{s}_{1:t-1}) q(s_t|\mathbf{s}_{1:t-1})}$$

$$\implies \quad \log w_t(\mathbf{s}_{1:t}) = \log \phi_{t-1}(\mathbf{s}_{1:t-1}) + \log \psi_t(\mathbf{s}_{1:t}) - \log \psi_{t-1}(\mathbf{s}_{1:t-1}) - \log \frac{q(s_t|\mathbf{s}_{1:t-1})}{p_0(s_t|\mathbf{s}_{1:t-1})} \tag{48}$$

Various special cases arise from choices of twist parameterizations and proposals in the following subsections.

C.1.1. Soft Q-Learning and RL Baseline

For single-step log-weights, the *ψ-parameterization* of the targets (Eq. (Twist Targets (ψ)), Eq. (Twist Targets (Soft RL Q))), and the *twist-induced proposal* (Eq. (Twist-Induced Proposal (ψ)), Eq. (Twist-Induced Proposal (soft RL))), we have

$$\log w_t(\mathbf{s}_{1:t}) = \log \phi_{t-1}(\mathbf{s}_{1:t-1}) + \log \psi_t(\mathbf{s}_{1:t}) - \log \psi_{t-1}(\mathbf{s}_{1:t-1}) - \left(\cancel{\log \frac{p_0(s_t|\mathbf{s}_{1:t-1})}{p_0(s_t|\mathbf{s}_{1:t-1})}} + \log \psi_t(\mathbf{s}_{1:t}) - \log \sum_{s_t} p_0(s_t|\mathbf{s}_{1:t-1})\psi_t(\mathbf{s}_{1:t})\right)$$

$$= \log \phi_{t-1}(\mathbf{s}_{1:t-1}) + \log \sum_{s_t} p_0(s_t|\mathbf{s}_{1:t-1})\psi_t(\mathbf{s}_{1:t}) - \log \psi_{t-1}(\mathbf{s}_{1:t-1}) \tag{49}$$

where the second term $\log Z_t^\pi(\mathbf{s}_{1:t-1}) = \log \sum_{s_t} p_0(s_t|\mathbf{s}_{1:t-1})\psi_t(\mathbf{s}_{1:t})$ normalizes the twist-induced proposal (Eq. (14)).

Minimizing the sum of *one-step log consistency losses* (i.e. squared log weights in Eq. (48)) will yield the familiar soft Q-learning loss (e.g. Lioutas et al. (2022) Eq. (4)-(5)). Adjusting indexing from Eq. (48) and introducing a stop-gradient within $\log Z_t^\pi(\mathbf{s}_{1:t-1})$, we have

$$\begin{aligned}
\min_{\boldsymbol{\theta}} \mathcal{L}_{\text{SOFTQ}}(\boldsymbol{\theta}) &\coloneqq \min_{\boldsymbol{\theta}} \sum_{t=1}^{T} \mathcal{L}^{(t+1,1)}_{\log \text{Cons}}(\boldsymbol{\theta}; \{\psi_t, q_t^\pi\}) && \text{(Soft Q Learning)} \\
&= \min_{\boldsymbol{\theta}} \sum_{t=1}^{T} \mathbb{E}_{\pi_s(\cdot)}\left[\left(\log \phi_t(\mathbf{s}_{1:t}) + \log \sum_{s_{t+1}} p_0(s_{t+1}|\mathbf{s}_{1:t}) \text{sg}\big(\psi^{\boldsymbol{\theta}}_{t+1}(\mathbf{s}_{1:t+1})\big) - \log \psi^{\boldsymbol{\theta}}_t(\mathbf{s}_{1:t})\right)^2\right] \\
&\overset{\text{(sRL)}}{=} \min_{\boldsymbol{\theta}} \sum_{t=1}^{T} \mathbb{E}_{\pi_s(\cdot)}\left[\left(\beta r_t(\mathbf{s}_{1:t}) + \log \sum_{s_{t+1}} p_0(s_{t+1}|\mathbf{s}_{1:t}) e^{\beta \text{sg}\left(Q^{\boldsymbol{\theta}}_t(s_{t+1}, \mathbf{s}_{1:t})\right)} - \beta Q^{\boldsymbol{\theta}}_t(s_t, \mathbf{s}_{1:t-1})\right)^2\right]
\end{aligned}$$

In the final line, we rewrite the loss for the soft RL special case, $\phi_t(\mathbf{s}_{1:t}) = e^{\beta r_t(\mathbf{s}_{1:t})}$ using the substitutions in Eq. (Twist to Soft RL). Note that the log-normalization term is analogous to an induced soft value $V_{Q_t^{\boldsymbol{\theta}}}(\mathbf{s}_{1:t-1}) = \frac{1}{\beta}\log \sum_{s_t} p_0(s_t|\mathbf{s}_{1:t-1}) e^{\beta Q^{\boldsymbol{\theta}}_t(s_t, \mathbf{s}_{1:t-1})}$, so that each squared error loss has the form $\mathbb{E}[\beta^2(r_t + V_t - Q_t)^2]$. Hence, we refer to this loss as *Soft Q-learning* loss.

The log-normalization term, which arises from normalizing the twist-induced proposal, is analogous to the 'target' value in deep Q-learning. Lioutas et al. (2022) consider the soft-Q learning loss to SMC sampling in self-driving applications where interaction with the environment is expensive. Lawson et al. (2018) adopt a similar loss function (using a parameterization of the value $V_t^{\boldsymbol{\theta}}$) in the setting of state-space models with tractable intermediate rewards.

RL Baseline with no Intermediate Reward The soft Q-learning loss in Eq. (Soft Q Learning) simplifies nicely in the case of no intermediate rewards, as in the main text ($\phi_t(\mathbf{s}_{1:t}) = 1$ for $t < T$ and $\Phi_T = 1$).

Written in terms of twist functions, we separate the terms at $t < T$ and $t = T$ for purposes of exposition

$$\begin{aligned}
&\min_{\boldsymbol{\theta}} \mathcal{L}_{\text{RL}}(\boldsymbol{\theta}) \coloneqq \min_{\boldsymbol{\theta}} \sum_{t=1}^{T} \mathcal{L}^{(t+1,1)}_{\log \text{Cons}}(\boldsymbol{\theta}; \{\psi_t, q_t^\pi, \phi_t = 1\}) && \text{(RL Baseline)} \\
&= \min_{\boldsymbol{\theta}} \sum_{t=1}^{T-1} \mathbb{E}_{\pi_s(\cdot)}\left[\left(\log \sum_{s_{t+1}} p_0(s_{t+1}|\mathbf{s}_{1:t}) \text{sg}\big(\psi^{\boldsymbol{\theta}}_{t+1}(\mathbf{s}_{1:t+1})\big) - \log \psi^{\boldsymbol{\theta}}_t(\mathbf{s}_{1:t})\right)^2\right] + \mathbb{E}_{\pi_s(\cdot)}\left[\left(\log \phi(\mathbf{s}_{1:T}) - \log \psi^{\boldsymbol{\theta}}_T(\mathbf{s}_{1:T})\right)^2\right]
\end{aligned}$$

For intermediate timesteps, note that Eq. (RL Baseline) enforces the recursion $\psi^{\boldsymbol{\theta}}_{t-1}(\mathbf{s}_{1:t-1}) = \sum_{s_t} p_0(s_t|\mathbf{s}_{1:t-1})\psi^{\boldsymbol{\theta}}_t(\mathbf{s}_{1:t})$ in Eq. (13) of the main text, albeit in log space. In App. C.2 below, we consider the one-step squared error loss enforcing this recursion directly (without logarithms), i.e. $\mathbb{E}_{\pi_s}[(\psi^{\boldsymbol{\theta}}_{t-1}(\mathbf{s}_{1:t-1}) - \sum_{s_t} p_0(s_t|\mathbf{s}_{1:t-1})\psi^{\boldsymbol{\theta}}_t(\mathbf{s}_{1:t}))^2]$,

C.1.2. Path Consistency Learning (for Twist Learning)

Using the *value parameterization* of the targets (Φ_t or V_t, see Eq. (Twist Targets (Φ)), Eq. (Twist Targets (Soft RL V))), the one-step log consistency loss with arbitrary proposal q recovers the path-consistency loss (PCL) from Nachum et al. (2017).

Switching to a $\Phi_t^{\boldsymbol{\theta}}$ parameterization of the twisting targets, we substitute $\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}) = \phi_t(\mathbf{s}_{1:t})\Phi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})$ into the log importance weights in Eq. (48). The log-consistency loss becomes,

$$\begin{aligned}
\min_{\boldsymbol{\theta}} \mathcal{L}_{\text{PCL}}(\boldsymbol{\theta}) &\coloneqq \min_{\boldsymbol{\theta}} \sum_{t=1}^{T} \mathcal{L}^{(t,1)}_{\log \text{Cons}}(\boldsymbol{\theta}; \{\Phi_t, \text{any } q\}) && \text{(PCL)} \\
&= \min_{\boldsymbol{\theta}} \sum_{t=1}^{T} \mathbb{E}_{\pi_s}\left[\left(\log \phi_t(\mathbf{s}_{1:t}) + \log \Phi^{\boldsymbol{\theta}}_t(\mathbf{s}_{1:t}) - \log \Phi^{\boldsymbol{\theta}}_{t-1}(\mathbf{s}_{1:t-1}) - \log \frac{q(s_t|\mathbf{s}_{1:t-1})}{p_0(s_t|\mathbf{s}_{1:t-1})}\right)^2\right] \\
&\overset{\text{(sRL)}}{=} \min_{\boldsymbol{\theta}} \sum_{t=1}^{T} \mathbb{E}_{\pi_s}\left[\left(\beta\Big(r_t(\mathbf{s}_{1:t}) + V^{\boldsymbol{\theta}}_t(\mathbf{s}_{1:t}) - V^{\boldsymbol{\theta}}_{t-1}(\mathbf{s}_{1:t-1})\Big) - \log \frac{q(s_t|\mathbf{s}_{1:t-1})}{p_0(s_t|\mathbf{s}_{1:t-1})}\right)^2\right]
\end{aligned}$$

In particular, substituting the soft RL potential terms from Eq. (Twist to Soft RL), Eq. (PCL) recovers the path consistency loss from Nachum et al. (2017). Note that we derived PCL from an importance sampling perspective, whereas PCL was originally derived by enforcing KKT conditions of the soft RL problem.

We might also consider multi-step losses for various c. Minimizing the square of the multi-step log weights with arbitrary q recovers the multi-step PCL loss (Nachum et al., 2017),

$$\begin{aligned}
\min_{\boldsymbol{\theta}} \mathcal{L}_{\text{PCL}}^{(t,c)}(\boldsymbol{\theta}) &:= \min_{\boldsymbol{\theta}} \mathcal{L}_{\log \text{Cons}}^{(t,c)}(\boldsymbol{\theta}; \{\Phi_t, \text{any } q\}) && \text{(multi-step PCL)} \\
&= \min_{\boldsymbol{\theta}} \mathbb{E}_{\pi_s}\left[\left(\sum_{\tau=t}^{t+c} \log \phi_\tau(\mathbf{s}_{1:\tau}) + \log \Phi^{\boldsymbol{\theta}}_{t+c}(\mathbf{s}_{1:t+c}) - \log \Phi^{\boldsymbol{\theta}}_{t-1}(\mathbf{s}_{1:t-1}) - \sum_{\tau=t}^{t+c} \log \frac{q(s_\tau|\mathbf{s}_{1:\tau-1})}{p_0(s_\tau|\mathbf{s}_{1:\tau-1})}\right)^2\right] \\
&= \min_{\boldsymbol{\theta}} \mathbb{E}_{\pi_s}\left[\left(\sum_{\tau=t-1}^{t+c-1} \log \phi_\tau(\mathbf{s}_{1:\tau}) + \log \psi^{\boldsymbol{\theta}}_{t+c}(\mathbf{s}_{1:t+c}) - \log \psi^{\boldsymbol{\theta}}_{t-1}(\mathbf{s}_{1:t-1}) - \sum_{\tau=t}^{t+c} \log \frac{q(s_\tau|\mathbf{s}_{1:\tau-1})}{p_0(s_\tau|\mathbf{s}_{1:\tau-1})}\right)^2\right] && (50) \\
&\overset{\text{(sRL)}}{=} \min_{\boldsymbol{\theta}} \mathbb{E}_{\pi_s}\left[\left(\beta \sum_{\tau=t}^{t+c} r_\tau(\mathbf{s}_{1:\tau}) + \beta\, V^{\boldsymbol{\theta}}_{t+c}(\mathbf{s}_{1:t+c}) - \beta\, V^{\boldsymbol{\theta}}_{t-1}(\mathbf{s}_{1:t-1}) - \sum_{\tau=t}^{t+c} \log \frac{q(s_\tau|\mathbf{s}_{1:\tau-1})}{p_0(s_\tau|\mathbf{s}_{1:\tau-1})}\right)^2\right]
\end{aligned}$$

where we write the $\psi_t^{\boldsymbol{\theta}}$ parameterization in Eq. (50) explicitly for use in App. D.1. While PCL considers learned a proposal or policy q with the goal of approximating the solution of a regularized MDP, we leave joint learning of proposals $\{q^{\boldsymbol{\xi}}(s_t|\mathbf{s}_{1:t-1})\}_{t=1}^T$ and SMC target twists $\{\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})\}_{t=1}^T$ or $\{V_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})\}_{t=1}^T$ to future work.

In App. E, we describe using PCL to learn the proposal *only* (Guo et al., 2021), with the values $V_{Q_t}(\mathbf{s}_{1:t})$ induced from learned proposal twists $Q_t^{\boldsymbol{\xi}}(s_{t+1}, \mathbf{s}_{1:t})$ which define $\{q^{\boldsymbol{\xi}}_{Q_t}(s_{t+1}|\mathbf{s}_{1:t})\}_{t=0}^{T-1}$ (in similar fashion to Eq. (Twist-Induced Proposal (soft RL)), but without reference to twisting targets).

C.2. Controlled Decoding Losses via Optimal Twist Identities (Mudgal et al., 2023)

In Prop. B.1 (or Prop. 3.2 and Eq. (13) in the main text), we noted that the optimal twists satisfy the following relationships

$$\begin{aligned}
\psi_t^*(\mathbf{s}_{1:t}) &= c_t\, \phi_t(\mathbf{s}_{1:t}) \sum_{\mathbf{s}_{t+1:T}} p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t}) \prod_{\tau=t+1}^{T} \phi_\tau(\mathbf{s}_{1:\tau}) && = \frac{c_t}{c_{t+1}} \phi_t(\mathbf{s}_{1:t}) \sum_{s_{t+1}} p_0(s_{t+1}|\mathbf{s}_{1:t}) \psi^*_{t+1}(\mathbf{s}_{1:t+1}) \\
&\overset{(\phi_t=1)}{=} c_t \sum_{\mathbf{s}_{t+1:T}} p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t}) \phi(\mathbf{s}_{1:T}) && \overset{(\phi_t=1)}{=} \frac{c_t}{c_{t+1}} \sum_{s_{t+1}} p_0(s_{t+1}|\mathbf{s}_{1:t}) \psi^*_{t+1}(\mathbf{s}_{1:t+1})
\end{aligned} \qquad (51)$$

We proceed to describe two 'controlled decoding' (CD) losses from Mudgal et al. (2023) as using a *squared error* loss to enforce the optimality conditions in Eq. (51), for settings with no intermediate potentials ($\phi_t(\mathbf{s}_{1:t}) = 1$ for $t < T$). Mudgal et al. (2023) also propose two ways to use the learned 'twists' at inference time, which we discuss in relation to our proposed SMC framework in App. D.1.

CD-Q The CD-Q loss from Mudgal et al. (2023) corresponds to minimizing the one-step recursion in Eq. (51) using the expected squared error under a (possibly off-policy) sampling distribution π_s. Assuming *no intermediate reward* and an additional squared error loss to approximate the terminal potential $\psi_T^{\boldsymbol{\theta}}(\mathbf{s}_{1:T}) \approx \phi(\mathbf{s}_{1:T})$, we have

$$\min_{\boldsymbol{\theta}} \mathcal{L}_{\text{CD-Q}}(\boldsymbol{\theta}) := \min_{\boldsymbol{\theta}} \sum_{t=1}^{T-1} \mathbb{E}_{\pi_s(\cdot)}\left[\Big(\sum_{s_{t+1}} p_0(s_{t+1}|\mathbf{s}_{1:t}) \psi^{\boldsymbol{\theta}}_{t+1}(\mathbf{s}_{1:t+1}) - \psi^{\boldsymbol{\theta}}_t(\mathbf{s}_{1:t})\Big)^2\right] + \mathbb{E}_{\pi_s(\cdot)}\left[\big(\phi(\mathbf{s}_{1:T}) - \psi^{\boldsymbol{\theta}}_T(\mathbf{s}_{1:T})\big)^2\right] \qquad \text{(CD-Q)}$$

Eq. (CD-Q) enforces the same optimality condition as the Eq. (RL Baseline) loss (i.e. $\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}) = \sum_{s_{t+1}} p_0(s_{t+1}|\mathbf{s}_{1:t}) \psi^{\boldsymbol{\theta}}_{t+1}(\mathbf{s}_{1:t+1})$), without log scaling of each term inside the squared error. At optimality, we have zero-variance one-step importance weights ($w(\mathbf{s}_{1:t}) = 1$ in Eq. (10)) for the twist-induced proposal.

At optimality, we in fact also have $\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}) = \sum_{\mathbf{s}_{t+1:T}} p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t}) \phi_T(\mathbf{s}_{1:T})$ (as in Eq. (51) and the proof of Prop. B.1).

CD-FUDGE While we might naively like to consider the loss $\mathbb{E}_{\pi_s(\cdot)}\big[\big(\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}) - \sum_{\mathbf{s}_{t+1:T}} p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t}) \phi(\mathbf{s}_{1:T})\big)^2\big]$ to enforce Prop. 3.2 or Eq. (51), note that marginalization over multiple steps is not tractable in general.

Instead, the CD-FUDGE loss[4] defined as

$$\min_{\boldsymbol{\theta}} \mathcal{L}_{\text{CD-FUDGE}}(\boldsymbol{\theta}) \coloneqq \min_{\boldsymbol{\theta}} \sum_{t=1}^{T} \mathbb{E}_{\pi_s(\mathbf{s}_{1:t})}\left[\mathbb{E}_{p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t})}\left[\left(\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}) - \phi(\mathbf{s}_{1:T})\right)^2\right]\right] \tag{CD-FUDGE}$$

can be shown to have the same gradient as the desired (but intractable) squared error loss above (Mudgal et al., 2023).

Since the minimizer of the expected squared error (under $p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t})$) to a single function $\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})$ (which is independent of $\mathbf{s}_{t+1:T}$) is given by the conditional expectation (Banerjee et al., 2005), we can also see that Eq. (CD-FUDGE) has the desired minimum $\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}) = \sum_{\mathbf{s}_{t+1:T}} p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t})\phi(\mathbf{s}_{1:T})$. Note, it is crucial that the inner expectation samples rollouts under the base model $p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t})$ to obtain the desired conditional expectation as the minimizer. While it appears that any prefix sampling distribution can be used, $\pi_s = p_0$ allows for losses to be calculated at all t in a single sampling run.

Mudgal et al. (2023) also propose two decoding-time usages for the learned twist functions $\psi_t^{\boldsymbol{\theta}}$: stochastic token-by-token sampling and argmax decoding of partial sequences. We discuss their inconsistencies with our SMC framework in App. D.

CD-FUDGE for $\log \psi_t^{\boldsymbol{\theta}}$ We can also compare Eq. (CD-FUDGE) with the multi-step PCL loss in Eq. (50), choosing $\phi_t = 1$ for $t < T$ and the proposal equal to the base model $q = p_0$ so that the proposal terms cancel. Noting that $\psi_T(\mathbf{s}_{1:T}) = \phi(\mathbf{s}_{1:T})$ is fixed to the exact terminal potential and choosing the $c = T - t + 1$-step PCL loss for each t, note that Eq. (50) would reduce to $\sum_t \mathbb{E}[\left(\log \phi(\mathbf{s}_{1:T}) + 0 - \log \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}) - 0\right)^2]$. Deng & Raffel (2023) optimize this loss with reweighting of terms based on timestep (higher weight for $t \approx T$). Eq. (CD-FUDGE) optimizes the squared error of the difference *without log scaling of each term*, under appropriate sampling of rollouts. [5]

C.3. SIXO: Smoothing Inference with Twisted Objectives (Lawson et al., 2022)

Lawson et al. (2022) adopt a noise-contrastive estimation loss (Gutmann & Hyvärinen, 2010) to learn the target twist functions using binary classification. For state space models, Lawson et al. (2022) adopt our setting in App. B.2 with observation variables o_t emitted based on the sampling state $\mathbf{s}_{1:t}$ (or simply s_t) and a known likelihood $\phi_t(o_t, s_t) = \sigma(o_t|s_t)$. As discussed in App. B.4, in these settings with easily evaluable intermediate potentials, it may be preferable to parameterize $\Phi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}, \boldsymbol{o}_{t+1:T})$ as in Eq. (Twist Targets (Φ)). Lawson et al. (2022) indeed use this parameterization (see their Eq. 5).

Recall from Eq. (38) that the optimal twists or future values amount to conditional likelihoods,

$$\Phi_t^*(\mathbf{s}_{1:t}, \boldsymbol{o}_{t+1:T}) \overset{\boldsymbol{o}_{t+1:T}}{\propto} \sigma(\boldsymbol{o}_{t+1:T}|\mathbf{s}_{1:t}), \qquad \psi_t^*(\mathbf{s}_{1:t}, \boldsymbol{o}_{t:T}) \overset{\boldsymbol{o}_{t:T}}{\propto} \sigma(\boldsymbol{o}_{t:T}|\mathbf{s}_{1:t}), \tag{52}$$

where $\overset{\boldsymbol{o}}{\propto}$ denotes proportionality up to a constant which depends on $\boldsymbol{o}$ only. Using Bayes rule, we have

$$\sigma(\boldsymbol{o}_{t+1:T}|\mathbf{s}_{1:t}) = \frac{\sigma(\mathbf{s}_{1:t}|\boldsymbol{o}_{t+1:T})\sigma(\boldsymbol{o}_{t+1:T})}{p_0(\mathbf{s}_{1:t})} \overset{\boldsymbol{o}_{t+1:T}}{\propto} \frac{\sigma(\mathbf{s}_{1:t}|\boldsymbol{o}_{t+1:T})}{p_0(\mathbf{s}_{1:t})}, \qquad \sigma(\boldsymbol{o}_{t:T}|\mathbf{s}_{1:t}) \overset{\boldsymbol{o}_{t:T}}{\propto} \frac{\sigma(\mathbf{s}_{1:t}|\boldsymbol{o}_{t:T})}{p_0(\mathbf{s}_{1:t})}, \tag{53}$$

noting that $\sigma(\boldsymbol{o}_{t+1:T})$ and $p_0(\mathbf{s}_{1:t})$ are marginals of $\sigma(\mathbf{s}_{1:t}, \boldsymbol{o}_{t+1:T})$ by definition. The above reasoning suggests that we may learn the twists, or likelihood ratio $\Phi_t^*(\mathbf{s}_{1:t}, \boldsymbol{o}_{t+1:T}) \propto \sigma(\boldsymbol{o}_{t+1:T}|\mathbf{s}_{1:t}) \propto \sigma(\mathbf{s}_{1:t}|\boldsymbol{o}_{t+1:T})/p_0(\mathbf{s}_{1:t})$, using a classifier which seeks to distinguish samples from $\sigma(\mathbf{s}_{1:t}|\boldsymbol{o}_{t+1:T})$ and $p_0(\mathbf{s}_{1:t})$ (Gutmann & Hyvärinen, 2010; Lawson et al., 2022). In particular, at each t, we classify the event $y = 1$, indicating that $\mathbf{s}_{1:t} \sim \sigma(\mathbf{s}_{1:t}|\boldsymbol{o}_{t+1:T})$, or $y = 0$, indicating that $\mathbf{s}_{1:t} \sim p_0(\mathbf{s}_{1:t})$.

Consider a given $\boldsymbol{o}_{t+1:T}$, which can be either $\boldsymbol{o}_{t+1:T} = \mathbf{1}$ in the unconditional case or $\boldsymbol{o}_{t+1:T} \sim \pi_s(\boldsymbol{o}_{t+1:T})$ drawn from a behavioral policy as discussed below. The SIXO loss becomes

$$\begin{aligned}
\mathcal{L}_{\text{SIXO}}(\boldsymbol{o}_{1:T}; \boldsymbol{\theta}) &= -\sum_{t=1}^{T-1} \mathbb{E}_{\sigma(\mathbf{s}_{1:t}|\boldsymbol{o}_{t+1:T})}\left[\log \texttt{sigmoid}\left(\log \Phi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}, \boldsymbol{o}_{t+1:T})\right)\right] + \mathbb{E}_{p_0(\mathbf{s}_{1:t})}\left[\log\left(1 - \texttt{sigmoid}\left(\log \Phi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}, \boldsymbol{o}_{t+1:T})\right)\right)\right] \\
&= -\sum_{t=1}^{T} \mathbb{E}_{\sigma(\mathbf{s}_{1:t}|\boldsymbol{o}_{t:T})}\left[\log \texttt{sigmoid}\left(\log \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}, \boldsymbol{o}_{t:T})\right)\right] + \mathbb{E}_{p_0(\mathbf{s}_{1:t})}\left[\log\left(1 - \texttt{sigmoid}\left(\log \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}, \boldsymbol{o}_{t:T})\right)\right)\right] \\
&= -\sum_{t=1}^{T} \mathbb{E}_{\sigma(\mathbf{s}_{1:t}|\boldsymbol{o}_{t:T})}\left[\log \frac{\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}, \boldsymbol{o}_{t:T})}{1 + \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}, \boldsymbol{o}_{t:T})}\right] + \mathbb{E}_{p_0(\mathbf{s}_{1:t})}\left[\log \frac{1}{1 + \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}, \boldsymbol{o}_{t:T})}\right]
\end{aligned} \tag{SIXO}$$

[4] Note, we reserve the naming convention FUDGE (Yang & Klein, 2021) for a binary cross entropy loss described in App. C.4, as opposed to the CD-FUDGE squared error loss from Mudgal et al. (2023).

[5] Note the difference in choice of proposal between Eq. (CD-Q) (twist-induced $q = q_t^{\pi}$) and Eq. (CD-FUDGE) (base $q = p_0$).

Note that we can perform approximate positive sampling as in Sec. 4 to estimate expectations in the first term.

Exact Conditional Sampling However, we can also use the BDMC trick in Sec. 3.3 to obtain exact target samples for general observation variables. In order to facilitate tractable sampling, we optimize the Eq. (SIXO) loss over a sampling distribution $\pi_s(\boldsymbol{o}_{1:T}) = \sigma(\boldsymbol{o}_{1:T})$ for all t, such that the objective becomes

$$\mathbb{E}_{\sigma(\boldsymbol{o}_{1:T})}[\mathcal{L}_{\text{SIXO}}(\boldsymbol{o}_{1:T};\boldsymbol{\theta})] = -\sum_{t=1}^{T}\mathbb{E}_{\sigma(\mathbf{s}_{1:t},\boldsymbol{o}_{t+1:T})}\left[\log\frac{\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t},\boldsymbol{o}_{t:T})}{1+\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t},\boldsymbol{o}_{t:T})}\right] + \mathbb{E}_{p_0(\mathbf{s}_{1:t})\sigma(\boldsymbol{o}_{t+1:T})}\left[\log\frac{1}{1+\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t},\boldsymbol{o}_{t:T})}\right]$$

With this choice, note that we may sample once from $\sigma(\mathbf{s}_{1:T},\boldsymbol{o}_{1:T}) = \prod_{t=1}^{T} p_0(s_t|\mathbf{s}_{1:t-1})\sigma(o_t|\mathbf{s}_{1:t})$ using ancestral sampling and use the appropriate truncations for positive sampling terms involving $\sigma(\mathbf{s}_{1:t},\boldsymbol{o}_{t+1:T})$. By shuffling observation variables across a batch of K samples, we may obtain samples from the product of marginals $p_0(\mathbf{s}_{1:T})\sigma(\boldsymbol{o}_{1:T})$ or $p_0(\mathbf{s}_{1:t})\sigma(\boldsymbol{o}_{t+1:T})$ in the negative sampling term.

In the main text, note that we condition on $o_T = 1$ or $o_T = \mathbf{s}_{T+1:T+c}$ (for infilling).

Gradient and Comparison with CTL Proceeding with the $\psi_t^{\boldsymbol{\theta}}$ parameterization for the target $\sigma(\mathbf{s}_{1:T}|o_T) = \sigma(\mathbf{s}_{1:T})$ with fixed o_T and unconditional twists $\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})$, the gradient of Eq. (SIXO) with respect to $\boldsymbol{\theta}$ is

$$\begin{aligned}
-\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{SIXO}}(\boldsymbol{\theta}) &= \sum_{t=1}^{T}\mathbb{E}_{\sigma(\mathbf{s}_{1:t})}\left[\nabla_{\boldsymbol{\theta}}\log\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}) - \frac{\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})}{1+\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})}\nabla_{\boldsymbol{\theta}}\log\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})\right] - \mathbb{E}_{p_0(\mathbf{s}_{1:t})}\left[\frac{\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})}{1+\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})}\nabla_{\boldsymbol{\theta}}\log\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})\right] \\
&= \sum_{t=1}^{T}\mathbb{E}_{\sigma(\mathbf{s}_{1:t})}\left[\frac{1}{1+\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})}\nabla_{\boldsymbol{\theta}}\log\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})\right] - \mathbb{E}_{p_0(\mathbf{s}_{1:t})}\left[\frac{\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})}{1+\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})}\nabla_{\boldsymbol{\theta}}\log\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})\right]
\end{aligned} \tag{SIXO Gradient}$$

The SIXO gradient is superficially similar to our CTL gradient in Sec. 4.1, in that it involves $\nabla_{\boldsymbol{\theta}}\log\psi_t^{\boldsymbol{\theta}}$ under positive and negatives samples. However, viewing $\tilde{\pi}_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}) = p_0(\mathbf{s}_{1:t})\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})$ as the unnormalized density of our intermediate twisting target, we can see that the second term in the SIXO update includes $\tilde{\pi}_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})$. Rewriting to highlight differences with our CTL gradient, we have

$$\begin{aligned}
-\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{SIXO}} &= \sum_{t=1}^{T}\left(\sum_{\mathbf{s}_{1:t}}\sigma(\mathbf{s}_{1:t})\frac{1}{1+\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})}\nabla_{\boldsymbol{\theta}}\log\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}) - \sum_{\mathbf{s}_{1:t}}\tilde{\pi}_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})\frac{1}{1+\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})}\nabla_{\boldsymbol{\theta}}\log\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})\right) \\
-\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{CTL}} &= \sum_{t=1}^{T}\left(\sum_{\mathbf{s}_{1:t}}\sigma(\mathbf{s}_{1:t})\quad\nabla_{\boldsymbol{\theta}}\log\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}) - \sum_{\mathbf{s}_{1:t}}\tilde{\pi}_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})\quad\frac{1}{\mathcal{Z}_t^{\psi}}\quad\nabla_{\boldsymbol{\theta}}\log\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})\right)
\end{aligned} \tag{SIXO vs. CTL}$$

To compare the two, first note that the positive sampling gradient in SIXO is scaled by a factor of $\frac{1}{1+\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})}$ factor (which reflects the misclassification probability under $\psi_t^{\boldsymbol{\theta}}$). For the negative sampling terms, note that $\tilde{\pi}_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})$ is divided by a factor of $\frac{1}{1+\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})}$ in the SIXO gradient, instead of the true normalization constant $\mathcal{Z}_t^{\psi}$ for the gradient of our CTL loss Eq. (21).

C.4. FUDGE: Future Discriminators (Yang & Klein, 2021)

In contrast to SIXO, the FUDGE method from Yang & Klein (2021) seeks to directly learn a discriminative classifier to match the conditional likelihood $\psi_t^*(\mathbf{s}_{1:t}, o_T) \propto \sigma(o_T|\mathbf{s}_{1:t})$ or $\psi_t^*(\mathbf{s}_{1:t},\boldsymbol{o}_{t:T}) \propto \sigma(\boldsymbol{o}_{t:T}|\mathbf{s}_{1:t})$ (see App. B.2).

As before, we define the joint distribution $\sigma(\mathbf{s}_{1:T}, o_T) = p_0(\mathbf{s}_{1:T})\sigma(o_T|\mathbf{s}_{1:T})$ with $\phi(\mathbf{s}_{1:T}, o_T) = \sigma(o_T|\mathbf{s}_{1:T})$. From Eq. (52) above or App. B.2 Eq. (39), we have

$$\psi_t^*(\mathbf{s}_{1:t}, o_T) \propto \sigma(o_T|\mathbf{s}_{1:t}) \coloneqq \sum_{\mathbf{s}_{t+1:T}} p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t})\sigma(o_T|\mathbf{s}_{1:T}) \tag{54}$$

Yang & Klein (2021) consider training a 'future discriminator' $\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}, o_T) \approx \sigma(o_T|\mathbf{s}_{1:t})$ which, as in Eq. (54) marginalizes over future tokens to predict the expected probability that a full sequence with prefix $\mathbf{s}_{1:t}$ emits o_T (e.g., let $o_T = a$ be the probability of a classifier for class a, or the probability that $\mathbf{s}_{1:T}$ satisfies a desired attribute indicated by a boolean $o_T = 1$).

In similar fashion to SIXO in the previous section, we define a binary random variable y such that

$$\sigma(y|\mathbf{s}_{1:t}, o_T) = \begin{cases} \sigma(o_T|\mathbf{s}_{1:t}) & y = 1 \\ 1 - \sigma(o_T|\mathbf{s}_{1:t}) & y = 0 \end{cases} \qquad p_{\psi_t^{\boldsymbol{\theta}}}(y|\mathbf{s}_{1:t}, o_T) = \begin{cases} \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}, o_T) & y = 1 \\ 1 - \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}, o_T) & y = 0 \end{cases} \tag{55}$$

where we directly parameterize $p_{\psi_t^{\boldsymbol{\theta}}}(y|\mathbf{s}_{1:t}, o_T) = \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}, o_T)$ to be a probability distribution (e.g. using a sigmoid or softmax activation). For a given observation random variable o_T and partial sequence $\mathbf{s}_{1:t}$, we can define the FUDGE loss

$$\begin{aligned} \sum_{t=1}^{T} \mathcal{L}_{\text{FUDGE}}(\mathbf{s}_{1:t}, o_T; \boldsymbol{\theta}) &\coloneqq \sum_{t=1}^{T} D_{\text{KL}}\Big(\sigma(y|\mathbf{s}_{1:t}, o_T) \,\Big\|\, p_{\psi_t^{\boldsymbol{\theta}}}(y|\mathbf{s}_{1:t}, o_T)\Big) \qquad \text{(FUDGE)} \\ &= \sum_{t=1}^{T} -\Big[\sigma(y = 1|\mathbf{s}_{1:t}, o_T) \log p_{\psi_t^{\boldsymbol{\theta}}}(y = 1|\mathbf{s}_{1:t}, o_T) + \sigma(y = 0|\mathbf{s}_{1:t}, o_T) \log p_{\psi_t^{\boldsymbol{\theta}}}(y = 0|\mathbf{s}_{1:t}, o_T)\Big] + \text{const.} \\ &= \sum_{t=1}^{T} -\mathbb{E}_{p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t})}\Big[\sigma(o_T|\mathbf{s}_{1:T}) \log \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}, o_T) + \Big(1 - \sigma(o_T|\mathbf{s}_{1:T})\Big) \log\Big(1 - \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}, o_T)\Big)\Big)\Big] + \text{const.} \end{aligned}$$

where, in moving from the second to the third line, we have used the fact that $\sigma(y = 1|\mathbf{s}_{1:t}, o_T) = \sigma(o_T|\mathbf{s}_{1:t}) = \sum_{\mathbf{s}_{t+1:T}} p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t})\sigma(o_T|\mathbf{s}_{1:T})$ from Eq. (54) and Eq. (55). At the optimum, $p_{\psi_t^{\boldsymbol{\theta}}}(y = 1|\mathbf{s}_{1:t}, o_T) = \sigma(y = 1|\mathbf{s}_{1:t}, o_T)$ implies $\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}, o_T) = \sigma(o_T|\mathbf{s}_{1:t})$, as desired.

While sampling may be done using an arbitrary distribution over prefixes $\mathbf{s}_{1:t}$ and observation o_T, Eq. (FUDGE) *requires* that rollouts be sampled under the base model $p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t})$ in order to ensure sampling from the appropriate distribution $\sigma(y = 1|\mathbf{s}_{1:t}, o_T)$. This restriction is similar to what we required in Eq. (CD-FUDGE), although the loss in Eq. (FUDGE) is based on cross entropy classification rather than a squared error. We discuss the choices made in our experiments below.

Yang & Klein (2021) Setting In the original FUDGE paper, Yang & Klein (2021) consider learning from a dataset of labelled examples $(\mathbf{s}_{1:T}, o_T)$ or $(\mathbf{s}_{1:t}, o_T)$ for a binary observation variable $o_T = 1$ which defines the target distribution.

Unconditional Twist Setting For the unconditional twist experiments in Sec. 7.2.1-7.2.2, we sample under the base model proposal $\pi_s(\mathbf{s}_{1:t}) = p_0(\mathbf{s}_{1:t})$ where the target distribution conditions on $o_T = 1$ and $\sigma(o_T = 1|\mathbf{s}_{1:T}) = \phi(\mathbf{s}_{1:T}) = \sigma(y = 1|\mathbf{s}_{1:T}, o_T = 1)$. In particular, we optimize

$$\min_{\theta} \sum_{t=1}^{T} \mathbb{E}_{p_0(\mathbf{s}_{1:t})}[\mathcal{L}_{\text{FUDGE}}(\mathbf{s}_{1:t}, o_T = 1; \boldsymbol{\theta})]$$

Conditional Twist Setting For conditional twist learning, we can consider amortizing learning the twists $\psi_t(\mathbf{s}_{1:t}, o_T)$ over some distribution of observation variables $\pi_s(\mathbf{s}_{1:t}, o_T)$. In particular, in our infilling experiments in Sec. 7.2.3, we consider sampling under the following joint distribution,

$$\pi_s(\mathbf{s}_{1:t}, o_T) = p_0(\mathbf{s}_{1:t})\sigma(o_T \,|\, \mathbf{s}_{1:t}) \,,$$

which we can sample from by first sampling from $p_0(\mathbf{s}_{1:T})\sigma(o_T \,|\, \mathbf{s}_{1:T})$ and then dropping $\mathbf{s}_{t+1:T}$ subsequence. Therefore, the overall objective becomes

$$\begin{aligned} &\min_{\boldsymbol{\theta}} \mathbb{E}_{\pi_s(\mathbf{s}_{1:t}, o_T)}[\mathcal{L}_{\text{FUDGE}}(\mathbf{s}_{1:t}, o_T; \boldsymbol{\theta})] \qquad (56) \\ &= \min_{\boldsymbol{\theta}} \sum_{t=1}^{T} -\mathbb{E}_{p_0(\mathbf{s}_{1:T})\sigma(o_T \,|\, \mathbf{s}_{1:t})}\Big[\sigma(o_T|\mathbf{s}_{1:T}) \log \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}, o_T) + \Big(1 - \sigma(o_T|\mathbf{s}_{1:T})\Big) \log\Big(1 - \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}, o_T)\Big)\Big)\Big] \,, \end{aligned}$$

where the expectation $p_0(\mathbf{s}_{1:T})$ includes the expectation under $p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t})$ from Eq. (FUDGE). Note that rollout of $\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t}$ used to sample from $p_0(\mathbf{s}_{1:T})$ should be independent of the rollout used to sample from $\sigma(o_T|\mathbf{s}_{1:t})$.

D. Decoding Strategies using Learned Twists from Mudgal et al. (2023)

D.1. Proposal Sampling in Mudgal et al. (2023)

As noted in App. C.2 (and in $\mathcal{L}^*(\boldsymbol{\theta})$ in Mudgal et al. (2023)), the CD losses can be seen as enforcing the optimality conditions

$$\psi_t^{\text{cd}*}(\mathbf{s}_{1:t}) = \sum_{\mathbf{s}_{t+1:T}} p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t})\phi(\mathbf{s}_{1:T}), \quad \forall t \qquad \text{and} \qquad \psi_t^{\text{cd}*}(\mathbf{s}_{1:t}) = \sum_{s_{t+1}} p_0(s_{t+1}|\mathbf{s}_{1:t})\psi_{t+1}^{\text{cd}*}(\mathbf{s}_{1:t+1}) \tag{57}$$

In RL terms, the twists $\psi_t^{\text{cd}*}$ perform *policy evaluation* for the expected 'reward' $\phi(\mathbf{s}_{1:T})$ of a fixed policy $p_0(\mathbf{s}_{1:T})$ in an *unregularized* MDP. The notation of Mudgal et al. (2023) (their Eq. (1), (5), our Eq. (57)) indeed corresponds to

$$\phi(\mathbf{s}_{1:T}) =: r_{\text{cd}}(\mathbf{s}_{1:T}) \qquad \sigma(\mathbf{s}_{1:T}) = p_0(\mathbf{s}_{1:T})r_{\text{cd}}(\mathbf{s}_{1:T}). \tag{CD reward}$$

However, Mudgal et al. (2023) propose to use the learned twist functions $\psi_t^{\boldsymbol{\theta}}$ to perform one-step sampling as

$$q_t^{\text{cd}}(s_t|\mathbf{s}_{1:t-1}) \propto p_0(s_t|\mathbf{s}_{1:t-1})e^{\beta\, \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})} \tag{CD proposal}$$

We proceed to explain that this scheme *does not correspond to sampling from the twist-induced proposal* under either of two different definitions of the target $\sigma(\mathbf{s}_{1:T})$ and potential $\phi(\mathbf{s}_{1:T})$ in our SMC framework.

Comparison with Our $\phi(\mathbf{s}_{1:T}) = r_{\mathbf{cd}}(\mathbf{s}_{1:T})$ Case: As we have argued above, the CD-Q and CD-FUDGE may be viewed as learning twist values $\psi_t^{\boldsymbol{\theta}}$ for a terminal potential $\phi(\mathbf{s}_{1:T}) = r_{\text{cd}}(\mathbf{s}_{1:T})$. However, our twist-induced proposal which minimizes the variance of the one-step importance weights with these SMC targets $\{\pi_t^{\boldsymbol{\theta}}\}$ would yield

$$q_t^{\pi}(s_t|\mathbf{s}_{1:t-1}) \propto p_0(s_t|\mathbf{s}_{1:t-1})\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}), \tag{Twist-Ind. proposal ($\phi = r_{\text{cd}}$)}$$

which, compared to Eq. (CD proposal) does not exponentiate or scale $\psi_t^{\boldsymbol{\theta}}$ and is directly proportional to the expected r_{cd}.

Comparison with Our $\phi(\mathbf{s}_{1:T}) = e^{\beta r_{\mathbf{cd}}(\mathbf{s}_{1:T})}$ Case (Soft RL): The stochastic sampling in Eq. (CD proposal) is also reminiscent of the twist-induced proposal in the soft RL case of our framework where, in contrast to Eq. (CD reward), the target is defined via $\phi(\mathbf{s}_{1:T}) = e^{\beta r_{\text{cd}}(\mathbf{s}_{1:T})}$. As in App. B.3,

$$q_t^{\pi}(s_t|\mathbf{s}_{1:t-1}) \propto p_0(s_t|\mathbf{s}_{1:t-1})e^{\beta\, V_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})} \tag{Twist-Ind. proposal ($\phi = e^{\beta r_{\text{cd}}}$)}$$

We proceed to write both q_t^{cd} and q_t^{π} as the solution to a variational optimization, highlighting similarities in blue, but noting the *different definitions of ϕ in terms of r_{cd}*. We assume no intermediate potential or reward, and consider the optimal twists to emphasize the role of r_{cd}. Using Mudgal et al. (2023) Eq. 2 and Thm 2.1 (for CD) and Eq. (Optimal Intermediate Soft Value) (for soft RL), we have

$$q_t^{\text{cd}^*}(s_t|\mathbf{s}_{1:t-1}) = \underset{q(s_t|\mathbf{s}_{1:t-1})}{\arg\max}\, \mathbb{E}_{q(s_t|\mathbf{s}_{1:t-1})}\Big[\underbrace{\mathbb{E}_{p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t})}\big[r_{\text{cd}}(\mathbf{s}_{1:T})\big]}_{\psi_t^{\text{cd}*}(\mathbf{s}_{1:t}) \quad (\text{for } \phi = r_{\text{cd}})}\Big] - \frac{1}{\beta} D_{\text{KL}}(q(s_t|\mathbf{s}_{1:t-1}) \,\|\, p_0(s_t|\mathbf{s}_{1:t-1}))$$

(CD proposal optimization)

$$q_t^{\pi^*}(s_t|\mathbf{s}_{1:t-1}) = \underset{q(s_t|\mathbf{s}_{1:t-1})}{\arg\max}\, \mathbb{E}_{q(s_t|\mathbf{s}_{1:t-1})}\Big[\underbrace{\frac{1}{\beta}\log \mathbb{E}_{p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t})}\big[e^{\beta r_{\text{cd}}(\mathbf{s}_{1:T})}\big]}_{V_t^*(\mathbf{s}_{1:t}) \quad (\text{for } \phi = e^{\beta r_{\text{cd}}})}\Big] - \frac{1}{\beta} D_{\text{KL}}(q(s_t|\mathbf{s}_{1:t-1}) \,\|\, p_0(s_t|\mathbf{s}_{1:t-1}))$$

(Soft RL proposal optimization)

The second terms of Eq. (CD proposal optimization) and Eq. (Soft RL proposal optimization) match and correspond to one-step KL divergence regularization of the policy $q_t(s_t|\mathbf{s}_{1:t-1})$. However, the expectation terms differ as we now discuss.

Soft Values Account for Future Regularization Using Eq. (Optimal Intermediate Soft Value) to expand the definition of the soft value function, we see that Eq. (Soft RL proposal optimization) also implicitly contains an expected terminal reward,

$$V_t^*(\mathbf{s}_{1:t}) = \tfrac{1}{\beta}\log \mathbb{E}_{p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t})} e^{\beta r_{\text{cd}}(\mathbf{s}_{1:T})} = \max_{q(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t})} \mathbb{E}_{q(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t})}\big[r_{\text{cd}}(\mathbf{s}_{1:T})\big] - \tfrac{1}{\beta} D_{\text{KL}}(q(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t}) \,\|\, p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t})) \tag{58}$$

As $\beta \to 0$ in Eq. (58), this optimization strictly enforces $q(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t}) = p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t})$, and the soft value function recovers the expected reward under the base model $\mathbb{E}_{p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t})}[r_{\text{cd}}(\mathbf{s}_{1:T})]$, which appears in first term Eq. (CD proposal optimization). On the other hand, the second term in Eq. (CD proposal optimization) uses $\beta > 0$ for optimization of the proposal $q(s_t|\mathbf{s}_{1:t-1})$ at the current step. This inconsistency in Eq. (CD proposal optimization) (using $\beta = 0$ in the first term and $\beta > 0$ in the second term) arises from the fact that Eq. (CD proposal optimization) does not consider the effect of *future* regularization, while the MDP formulation in Eq. (Soft RL proposal optimization) does so via the optimization in Eq. (58) and the log-mean-exp form of the soft value function V_t^*.

On Mudgal et al. (2023)'s One-Step Proposal and SMC Interpretation As noted in Eq. (57), the twists learned by Mudgal et al. (2023) correspond to policy evaluation for the reward r_{cd} under the base model p_0. However, we have argued that the one-step proposal in Eq. (CD proposal) (which considers one-step KL regularization of q_t^{cd} to p_0) does not immediately fit within our SMC framework.

In particular, it is not clear that the composition of one-step proposals $q^{\text{cd}}(\mathbf{s}_{1:t}) = \prod_{\tau=1}^t q_\tau^{\text{cd}}(s_\tau|\mathbf{s}_{1:\tau-1}) = p_0(\mathbf{s}_{1:t}) e^{\beta \sum_{\tau=1}^t \psi^{\text{cd}^*}(\mathbf{s}_{1:\tau}) - \beta \sum_{\tau=1}^t \frac{1}{\beta} \log \mathbb{E}_{p_0(s_\tau|\mathbf{s}_{1:\tau-1})} e^{\beta \psi^{\text{cd}^*}(\mathbf{s}_{1:\tau})}}$ samples from the marginals $\sigma(\mathbf{s}_{1:t})$ of a meaningful target distribution $\sigma(\mathbf{s}_{1:T})$ at optimality. On the other hand, the soft RL value functions satisfy the optimality condition $V_t^*(\mathbf{s}_{1:t-1}) = {}^1\!/\!_\beta \log \sum_{s_t} p_0(s_t|\mathbf{s}_{1:t-1})\, e^{\beta V_t^*(\mathbf{s}_{1:t})}$ in Eq. (43), which is the analogue of Eq. (57) for $\phi = e^{\beta r}$ (soft RL) but is not satisfied by substituting ψ^{cd^*} for V_t^*. This condition facilitates a telescoping cancellation, $\prod_{\tau=1}^t q_t^{\pi^*}(s_\tau|\mathbf{s}_{1:\tau-1}) = p_0(\mathbf{s}_{1:t}) e^{\beta \sum_{\tau=1}^t V^*(\mathbf{s}_{1:\tau}) - \beta \sum_{\tau=1}^t \frac{1}{\beta} \log \mathbb{E}_{p_0(s_\tau|\mathbf{s}_{1:\tau-1})} e^{\beta V^*(\mathbf{s}_{1:\tau})}} \propto p_0(\mathbf{s}_{1:t}) e^{\beta V^*(\mathbf{s}_{1:t})}$ and yields the marginals of $\sigma(\mathbf{s}_{1:T})$.

Flexible Inference-Time β Scaling The experiments in Mudgal et al. (2023) evaluate tradeoff curves between expected reward and $D_{\text{KL}}\big(q^{\text{cd}}(\mathbf{s}_{1:T}) \,\big\|\, p_0(\mathbf{s}_{1:T})\big)$ for various values of regularization strength β. Since the twists learned by Mudgal et al. (2023) in Eq. (57) do not depend on β, sampling according to Eq. (CD proposal) or Eq. (CD proposal optimization) has the benefit of allowing flexible tempering or β-scaling at inference time without additional learning.

Such tradeoff curves are also natural from the perspective of soft-RL (c.f. Eq. (41) and Eq. (46)). While Eq. (58) appears to require separate twist-learning procedures for each β, flexible inference-time β scaling could be achieved with a single training run in our framework by learning a conditional twist network $\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}, \beta)$ which considers β in its input and training loss, or adapting the methods of (Bae et al., 2022) proposed in the context of rate-distortion optimization.

Comparison with Khanov et al. (2024) Khanov et al. (2024) consider softmax decoding similar to Eq. (Twist-Ind. proposal ($\phi = r_{\text{cd}}$)). However, instead of $V_t^\theta(\mathbf{s}_{1:t})$ as the logit, they use a reward model $r_T(\mathbf{s}_{1:T})$ which is trained from full sequences ($\phi(\mathbf{s}_{1:T}) = e^{\beta r_T(\mathbf{s}_{1:T})}$), but applied to partial sequences without modification, $r_T(\mathbf{s}_{1:t})$. This clearly does not correspond to a twist or soft value function $V_t^*(\mathbf{s}_{1:t}) = \frac{1}{\beta} \log \sum_{\mathbf{s}_{t+1:T}} p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t}) e^{\beta r_T(\mathbf{s}_{1:T})} \neq r_T(\mathbf{s}_{1:t})$.

D.2. Blockwise Greedy Decoding in Mudgal et al. (2023)

As an alternative use of the twist functions at inference time and a generalization of best-of-K decoding to partial sequences, Mudgal et al. (2023) consider a 'blockwise' decoding scheme using the learned twist functions $\psi_t^{\boldsymbol{\theta}}$. For K partial completions of length M (from a prefix $\mathbf{s}_{1:t}$), sampled from the base model, $\mathbf{s}_{t+1:t+M}^{(k)} \sim p_0(\mathbf{s}_{t+1:t+M}|\mathbf{s}_{1:t})$, Mudgal et al. (2023) select

$$\mathbf{s}_{t+1:t+M}^{\omega} = \arg\max_k \psi_{t+M}^{\boldsymbol{\theta}}(\mathbf{s}_{1:t+M}^k) \tag{59}$$

and proceed with sampling K further continuations with prefix $\mathbf{s}_{1:t+M}^{\omega}$ until the next resampling step or an end-of-string token is reached. The $\arg\max$ selection strategy may seem natural from the unregularized RL (as $\beta \to \infty$) or expected future reward perspective in App. D.1, but does not yield samples from $\sigma(\mathbf{s}_{1:T})$ with the corresponding optimal twists. Finally, Khanov et al. (2024) also consider $\arg\max$ decoding of next tokens using the unmodified $r_T(\mathbf{s}_{1:t})$ described above.

Our SMC framework instead would advocate *probabilistic* resampling based on the approximate twist functions using the (c- or M-step) importance weights in Sec. 3 in order to match the desired target distribution.

E. Proposal Learning Methods

We next describe methods for learning variational policies or proposals $q^{\boldsymbol{\xi}}(\mathbf{s}_{1:T}) = \prod_{t=1}^T q_t^{\boldsymbol{\xi}}(s_t|\mathbf{s}_{1:t-1})$ parameterized by $\boldsymbol{\xi}$, which can be used for SMC sampling with intermediate targets $\pi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})$ and learned twists $\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})$ or $V_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})$ parameterized by $\boldsymbol{\theta}$. Alternatively, such proposals may be used directly in the IWAE bounds on $\log \mathcal{Z}_\sigma$, which rely on simple importance sampling over full sequences as in Sec. 2.1 and *do not require the definition of intermediate targets* π_t.

In App. E.3, we provide a detailed description of the DPG policy gradient method, which can be interpreted as a maximum likelihood objective for a sequential energy-based model. To distinguish this EBM approach from our CTL method for twist learning, we emphasize issues which can arise from naive use of a proposal-learning objective to define intermediate twisting targets for SMC in App. E.3.1.

E.1. Path Consistency Learning for Controlled Generation

Guo et al. (2021) consider learning Q-values to obtain a fine-tuned variational policy which can be directly used as a sampling distribution for controlled generation. Building on the path consistency learning (PCL) loss in Nachum et al. (2017) and App. C.1.2, Guo et al. (2021) consider parameterizing the proposal using $Q_t^{\boldsymbol{\xi}}(s_t, \mathbf{s}_{1:t-1})$,

$$q_t^{\boldsymbol{\xi}}(s_t|\mathbf{s}_{1:t-1}) = p_0(s_t|\mathbf{s}_{1:t-1})e^{\beta Q_t^{\boldsymbol{\xi}}(s_t,\mathbf{s}_{1:t-1})-\beta V_{Q^{\boldsymbol{\xi}}}(\mathbf{s}_{1:t-1})} \tag{60}$$

where $V_{Q^{\boldsymbol{\xi}}}(\mathbf{s}_{1:t-1}) = \frac{1}{\beta}\log\sum_{s_t} p_0(s_t|\mathbf{s}_{1:t-1})e^{\beta Q_t^{\boldsymbol{\xi}}}$ enforces normalization.

Guo et al. (2021) define the targets using $\bar{Q}_t^{\boldsymbol{\xi}}(s_t, \mathbf{s}_{1:t-1})$, a slowly-updated target network based on $Q_t^{\boldsymbol{\xi}}$. Using the implied form of the soft value $\bar{V}(\mathbf{s}_{1:t-1}) \coloneqq \frac{1}{\beta}\log\sum_{s_t} p_0(s_t|\mathbf{s}_{1:t-1})e^{\beta\bar{Q}_t^{\boldsymbol{\xi}}(s_t,\mathbf{s}_{1:t-1})}$, the single-step PCL loss becomes

$$\mathcal{L}_{\text{PCL}-\text{Q}}(\boldsymbol{\xi}) = \min_{\boldsymbol{\xi}} \sum_{t=1}^{T} \mathbb{E}_{\pi_s(\mathbf{s}_{1:t})}\left[\left(r_t(\mathbf{s}_{1:t}) + \mathtt{sg}(\bar{V}_t(\mathbf{s}_{1:t})) - \mathtt{sg}(\bar{V}_{t-1}(\mathbf{s}_{1:t-1})) - Q_t^{\boldsymbol{\xi}}(s_t, \mathbf{s}_{1:t-1}) + V_{Q^{\boldsymbol{\xi}}}(\mathbf{s}_{1:t-1})\right)^2\right] \tag{61}$$

where $\mathtt{sg}(\cdot)$ indicates stop gradient. Building on the interpretation in App. C.1, we view $\bar{V}_t(\mathbf{s}_{1:t})$ and $\bar{V}_{t-1}(\mathbf{s}_{1:t-1})$ as the twisting targets, with a learned proposal parameterized by $Q_t^{\boldsymbol{\xi}}$ as in Eq. (60) (or App. B.4). While the loss in Eq. (61) is similar in practice to the soft Q-learning loss in App. C.1.1, we emphasize that the latter is motivated from the SMC perspective with the twisting targets as the primary object of interest and flexibility in the choice of proposal. By contrast, Guo et al. (2021) are interested in learning a proposal policy and do not consider, for example, resampling according to $\bar{V}_t$.

Guo et al. (2021); Nachum et al. (2017) also consider 'multi-step' PCL losses (Eq. (multi-step PCL)) which use observed reward during rollouts of length λ to limit reliance on estimated intermediate values $\bar{V}_t(\mathbf{s}_{1:t})$. The objective in Hu et al. (2023) also corresponds to a PCL objective.

E.2. Policy Gradient Methods

Traditional RLHF pipelines use a policy gradient method such as PPO to optimize the objective in Eq. (41),

$$\mathcal{L}_{\text{ELBO}}(\boldsymbol{\xi}) = \max_{\boldsymbol{\xi}} \mathbb{E}_{q^{\boldsymbol{\xi}}(\mathbf{s}_{1:T})}[r_T(\mathbf{s}_{1:T})] - \frac{1}{\beta}D_{\text{KL}}\big(q^{\boldsymbol{\xi}}(\mathbf{s}_{1:T}) \,\big\|\, p_0(\mathbf{s}_{1:T})\big) = \min_{\boldsymbol{\xi}} D_{\text{KL}}\big(q^{\boldsymbol{\xi}}(\mathbf{s}_{1:T}) \,\big\|\, \sigma(\mathbf{s}_{1:T})\big) \tag{62}$$

where $r_T(\mathbf{s}_{1:T}) = \frac{1}{\beta}\log\phi(\mathbf{s}_{1:T})$ corresponds to our final twist. As in Eq. (46), the gap in this optimization is the mode-seeking KL divergence $D_{\text{KL}}\big(q^{\boldsymbol{\xi}}(\mathbf{s}_{1:T}) \,\big\|\, \sigma(\mathbf{s}_{1:T})\big)$.

Notably, this objective does not make use of exact target samples from $\sigma(\mathbf{s}_{1:T})$ when they are available. Further, the mode-seeking behavior has been shown to reduce diversity of fine-tuned models (Stiennon et al., 2020; Go et al., 2023). To combat this, Go et al. (2023) derive policy gradient methods to optimize arbitrary f-divergences $D_f\big(q^{\boldsymbol{\xi}}(\mathbf{s}_{1:T}) \,\big\|\, \sigma(\mathbf{s}_{1:T})\big)$ between the learned variational policy $q^{\boldsymbol{\xi}}$ and target σ.

E.3. Policy Gradient with Mass-Covering / Maximum Likelihood KL Divergence

We focus on the case of minimizing the mass-covering KL divergence $D_{\text{KL}}\big(\sigma(\mathbf{s}_{1:T}) \,\big\|\, q^{\boldsymbol{\xi}}(\mathbf{s}_{1:T})\big)$ to train $q_{\boldsymbol{\xi}}$, which constitutes the distributional policy gradients (DPG) method for language model finetuning (Parshakova et al., 2019; Khalifa et al., 2020; Korbak et al., 2022a; Go et al., 2023) and has been used to learn SMC proposals in state-space models in (Gu et al., 2015).

In particular, the gradient of $D_{\text{KL}}\big(\sigma(\mathbf{s}_{1:T}) \,\big\|\, q^{\boldsymbol{\xi}}(\mathbf{s}_{1:T})\big) = \mathbb{E}_{\sigma(\mathbf{s}_{1:T})}[\log\sigma(\mathbf{s}_{1:T}) - \log q^{\boldsymbol{\xi}}(\mathbf{s}_{1:T})]$ is

$$\begin{aligned}\nabla_{\boldsymbol{\xi}} D_{\text{KL}}\big(\sigma(\mathbf{s}_{1:T}) \,\big\|\, q^{\boldsymbol{\xi}}(\mathbf{s}_{1:T})\big) = -\mathbb{E}_{\sigma(\mathbf{s}_{1:T})}\big[\nabla_{\boldsymbol{\xi}}\log q^{\boldsymbol{\xi}}(\mathbf{s}_{1:T})\big] &= -\mathbb{E}_{q^{\boldsymbol{\xi}}(\mathbf{s}_{1:T})}\left[\frac{\sigma(\mathbf{s}_{1:T})}{q^{\boldsymbol{\xi}}(\mathbf{s}_{1:T})}\nabla_{\boldsymbol{\xi}}\log q^{\boldsymbol{\xi}}(\mathbf{s}_{1:T})\right] \\ &= -\mathbb{E}_{q^{\boldsymbol{\xi}}(\mathbf{s}_{1:T})}\left[\frac{1}{\mathcal{Z}_\sigma}\frac{\tilde{\sigma}(\mathbf{s}_{1:T})}{q^{\boldsymbol{\xi}}(\mathbf{s}_{1:T})}\nabla_{\boldsymbol{\xi}}\log q^{\boldsymbol{\xi}}(\mathbf{s}_{1:T})\right]\end{aligned} \tag{63}$$

We recognize the importance weights $w(\mathbf{s}_{1:T}) = \frac{\tilde{\sigma}(\mathbf{s}_{1:T})}{q^{\boldsymbol{\xi}}(\mathbf{s}_{1:T})}$ from Eq. (3). Go et al. (2023) consider estimating Eq. (63) using a moving average estimate of the partition function $\hat{Z}_\sigma$

$$\nabla_{\boldsymbol{\xi}} D_{\mathrm{KL}}\big(\sigma(\mathbf{s}_{1:T}) \,\big\|\, q^{\boldsymbol{\xi}}(\mathbf{s}_{1:T})\big) \;\approx\; \sum_{k=1}^{K} \frac{1}{\hat{Z}_\sigma} w(\mathbf{s}_{1:T}^{(k)}) \nabla_{\boldsymbol{\xi}} \log q^{\boldsymbol{\xi}}(\mathbf{s}_{1:T}^{(k)}), \qquad \text{(DPG (general } \hat{Z}_\sigma\text{))}$$

Alternatively, the expectation may thus be estimated using SIS with the variational policy $q^{\boldsymbol{\xi}}(\mathbf{s}_{1:T})$. Using self-normalized importance sampling (SNIS) to estimate Eq. (63) as in Eq. (5) corresponds to $\hat{Z}_\sigma = \sum_{j=1}^{K} w(\mathbf{s}_{1:T}^{(k)})$, with

$$\nabla_{\boldsymbol{\xi}} D_{\mathrm{KL}}\big(\sigma(\mathbf{s}_{1:T}) \,\big\|\, q^{\boldsymbol{\xi}}(\mathbf{s}_{1:T})\big) \;\approx\; \sum_{k=1}^{K} \frac{w(\mathbf{s}_{1:T}^{(k)})}{\sum_{j=1}^{K} w(\mathbf{s}_{1:T}^{(j)})} \nabla_{\boldsymbol{\xi}} \log q^{\boldsymbol{\xi}}(\mathbf{s}_{1:T}^{(k)}). \tag{64}$$

We use this gradient for DPG proposal learning in the main text experiments, although we use the parameterization described in Eq. (DPG) below.

DPG as Sequential Maximum Likelihood Objective We now show Eq. (64) is equivalent to a sequential maximum likelihood EBM objective. Consider minimizing the KL divergence,

$$D_{\mathrm{KL}}\big(\sigma(\mathbf{s}_{1:T}) \,\big\|\, q^{\boldsymbol{\xi}}(\mathbf{s}_{1:T})\big) = \sum_{t=1}^{T} \mathbb{E}_{\sigma(\mathbf{s}_{1:t-1})} D_{\mathrm{KL}}\Big(\sigma(s_t|\mathbf{s}_{1:t-1}) \,\Big\|\, q_t^{\boldsymbol{\xi}}(s_t|\mathbf{s}_{1:t-1})\Big) \qquad \text{(EBM proposal learning)}$$

$$\text{where} \qquad q_t^{\boldsymbol{\xi}}(s_t|\mathbf{s}_{1:t-1}) = \frac{p_0(s_t|\mathbf{s}_{1:t-1})\psi_t^{\boldsymbol{\xi}}(\mathbf{s}_{1:t})}{\sum_{s_t} p_0(s_t|\mathbf{s}_{1:t-1})\psi_t^{\boldsymbol{\xi}}(\mathbf{s}_{1:t})}. \tag{65}$$

While this is reminscent of the twist-induced proposal in Prop. 3.3, we emphasize distinctions between energy-based learning of the proposal (DPG) versus energy-based learning of twist functions (CTL) in App. E.3.1.

The gradient of Eq. (EBM proposal learning) becomes

$$\nabla_{\boldsymbol{\xi}} D_{\mathrm{KL}}\Big(\sigma(\mathbf{s}_{1:T}) \,\Big\|\, q^{\boldsymbol{\xi}}(\mathbf{s}_{1:T})\Big) = \sum_{t=1}^{T} \mathbb{E}_{\sigma(\mathbf{s}_{1:t-1})}\Big[\mathbb{E}_{\sigma(s_t|\mathbf{s}_{1:t-1})}\big[\nabla_{\boldsymbol{\xi}} \log \psi_t^{\boldsymbol{\xi}}(\mathbf{s}_{1:t})\big] - \mathbb{E}_{q_t^{\boldsymbol{\xi}}(s_t|\mathbf{s}_{1:t-1})}\big[\nabla_{\boldsymbol{\xi}} \log \psi_t^{\boldsymbol{\xi}}(\mathbf{s}_{1:t})\big]\Big]. \tag{66}$$

Starting from Eq. (64), we now seek to recover Eq. (66). Using Eq. (65), we can write

$$\log q^{\boldsymbol{\xi}}(\mathbf{s}_{1:T}^{(k)}) = \sum_{t=1}^{T} \big(\log p_0(s_t^{(k)}|\mathbf{s}_{1:t-1}^{(k)}) + \log \psi_t^{\boldsymbol{\xi}}(\mathbf{s}_{1:t}^{(k)}) - \log \sum_{s_t} p_0(s_t|\mathbf{s}_{1:t-1}^{(k)})\psi_t^{\boldsymbol{\xi}}(s_t, \mathbf{s}_{1:t-1}^{(k)})\big)$$

$$\nabla_{\boldsymbol{\xi}} \log q^{\boldsymbol{\xi}}(\mathbf{s}_{1:T}^{(k)}) = \sum_{t=1}^{T} \Big(\nabla_{\boldsymbol{\xi}} \log \psi_t^{\boldsymbol{\xi}}(\mathbf{s}_{1:t}^{(k)}) - \mathbb{E}_{q_t^{\boldsymbol{\xi}}(s_t|\mathbf{s}_{1:t-1}^{(k)})}\Big[\nabla_{\boldsymbol{\xi}} \log \psi_t^{\boldsymbol{\xi}}(s_t, \mathbf{s}_{1:t-1}^{(k)})\Big]\Big)$$

Substituting into Eq. (64), we recover

$$\nabla_{\boldsymbol{\xi}} D_{\mathrm{KL}}\big(\sigma(\mathbf{s}_{1:T}) \,\big\|\, q^{\boldsymbol{\xi}}(\mathbf{s}_{1:T})\big) \;\approx\; \sum_{k=1}^{K} \frac{w(\mathbf{s}_{1:T}^{(k)})}{\sum_{j=1}^{K} w(\mathbf{s}_{1:T}^{(k)})} \sum_{t=1}^{T} \Big(\nabla_{\boldsymbol{\xi}} \log \psi_t^{\boldsymbol{\xi}}(\mathbf{s}_{1:t}^{(k)}) - \mathbb{E}_{q_t^{\boldsymbol{\xi}}(s_t|\mathbf{s}_{1:t-1}^{(k)})}\Big[\nabla_{\boldsymbol{\xi}} \log \psi_t^{\boldsymbol{\xi}}(s_t, \mathbf{s}_{1:t-1}^{(k)})\Big]\Big) \tag{DPG}$$

which is an SNIS estimate of the maximum likelihood EBM gradient in Eq. (66), as desired. Note that the expectation over $q_t^{\boldsymbol{\xi}}(s_t|\mathbf{s}_{1:t-1}^{(k)})$ can be calculated exactly.

Comparison with CTL Objective The gradient in Eq. (DPG) above appears similar to our CTL objective and gradient in Sec. 4.1. However, the DPG loss in Eq. (EBM proposal learning) is a single (joint) KL divergence over the entire sequence, whereas CTL optimizes T separate KL divergences for each intermediate marginal.

For the DPG gradient in Eq. (66), negative sampling is performed using a 'positive' prefix $\mathbf{s}_{1:t-1}^{(k)} \sim \sigma(\mathbf{s}_{1:t-1})$ and an *exact* 'negative' sample from the one-step-ahead $q_t^{\boldsymbol{\xi}}(s_t|\mathbf{s}_{1:t-1}^{(k)})$ (Eq. (65), which we have assumed to be tractable). In practice, we obtain the prefixes using the truncation of exact samples or approximate positive sampling with the final target weights as in Eq. (DPG). By contrast, the CTL gradient in Eq. (21) involves *approximate* negative sampling under each $\pi_t(\mathbf{s}_{1:t})$.

E.3.1. Naive Use of Proposal Learning to define Twisted SMC Targets

While we have shown in Prop. 3.3 how one-step proposals $\{q_t^\pi(s_t|\mathbf{s}_{1:t-1})\}_{t=1}^T$ can be induced from a given set of twist functions $\{\psi_t(\mathbf{s}_{1:t})\}_{t=1}^T$ or target distributions $\{\pi_t(\mathbf{s}_{1:t})\}_{t=1}^T$, we now emphasize that moving the other direction (inducing intermediate twisting targets from a proposal learning scheme parameterized by $\{\psi_t^{\boldsymbol{\xi}}\}_{t=1}^T$) does not yield the correct intermediate targets for resampling (App. A.1), even at optimality in the proposal learning objective.

We focus our arguments on learning with the EBM maximum likelihood objective in Eq. (EBM proposal learning) as an example. The proposal energies $\psi_t^{\boldsymbol{\xi}}(\mathbf{s}_{1:t})$ appear to play a role analogous to the twist function $\psi_t(\mathbf{s}_{1:t})$ in the one-step proposal induced from twist targets $\{\pi_t\}_{t=1}^T$ in Sec. 3.

However, we will show in Prop. E.2 that naive use of $\psi_t^{\boldsymbol{\xi}}$ to define the following twisting targets (with no intermediate ϕ_t)

$$\pi_t^{\boldsymbol{\xi}}(\mathbf{s}_{1:t}) = \begin{cases} \frac{1}{\mathcal{Z}_t^\psi}\, p_0(\mathbf{s}_{1:t})\, \psi_t^{\boldsymbol{\xi}}(\mathbf{s}_{1:t}) & t \neq T \\ \frac{1}{\mathcal{Z}_\sigma}\, p_0(\mathbf{s}_{1:T})\, \phi(\mathbf{s}_{1:T}) & t = T \end{cases} \tag{67}$$

need not lead to an SMC procedure for which $\pi_t^{\boldsymbol{\xi}}(\mathbf{s}_{1:t}) = \sigma(\mathbf{s}_{1:t})$, even if $q_t^{\boldsymbol{\xi}}(s_t|\mathbf{s}_{1:t-1}) = \sigma(s_t|\mathbf{s}_{1:t-1})$ for all t. We thus argue that $\psi_t^{\boldsymbol{\xi}}$ learned using Eq. (EBM proposal learning) *should not be used* as target twists in Eq. (67), since they do not yield the optimal interemdiate target distributions at optimality (App. A.1).

We begin by showing a simple lemma for the one-step conditionals in Eq. (EBM proposal learning).

Lemma E.1. *Any twist induced proposal $q_t^{\boldsymbol{\xi}}(s_t|\mathbf{s}_{1:t-1})$ (induced by $\psi_t^{\boldsymbol{\xi}}(\mathbf{s}_{1:t})$) is invariant to rescaling $\psi_t^{\boldsymbol{\xi}}(\mathbf{s}_{1:t})$ by an arbitrary constant $c(\mathbf{s}_{1:t-1})$ with respect to $\mathbf{s}_{1:t-1}$,*

$$\psi_t^{\boldsymbol{\xi}c}(\mathbf{s}_{1:t}) \coloneqq c(\mathbf{s}_{1:t-1})\psi_t^{\boldsymbol{\xi}}(\mathbf{s}_{1:t}) \tag{68}$$

Proof.

$$q_t^{\boldsymbol{\xi}c}(s_t|\mathbf{s}_{1:t-1}) = \frac{p_0(s_t|\mathbf{s}_{1:t-1})\psi_t^{\boldsymbol{\xi}c}(\mathbf{s}_{1:t})}{\sum_{s_t} p_0(s_t|\mathbf{s}_{1:t-1})\psi_t^{\boldsymbol{\xi}c}(\mathbf{s}_{1:t})} = \frac{p_0(s_t|\mathbf{s}_{1:t-1})c(\mathbf{s}_{1:t-1})\psi_t^{\boldsymbol{\xi}}(\mathbf{s}_{1:t})}{\sum_{s_t} p_0(s_t|\mathbf{s}_{1:t-1})c(\mathbf{s}_{1:t-1})\psi_t^{\boldsymbol{\xi}}(\mathbf{s}_{1:t})} = \frac{p_0(s_t|\mathbf{s}_{1:t-1})\psi_t^{\boldsymbol{\xi}}(\mathbf{s}_{1:t})}{\sum_{s_t} p_0(s_t|\mathbf{s}_{1:t-1})\psi_t^{\boldsymbol{\xi}}(\mathbf{s}_{1:t})} = q_t^{\boldsymbol{\xi}}(s_t|\mathbf{s}_{1:t-1})\,.$$

□

Proposition E.2. *There exist $\{\psi_t^{\boldsymbol{\xi}^*}\}_{t=1}^T$ such that (i) $q_t^{\boldsymbol{\xi}^*}(s_t|\mathbf{s}_{1:t-1}) = \sigma(s_t|\mathbf{s}_{1:t-1})$ and (ii) the SMC targets $\{\pi_t^{\boldsymbol{\xi}^*}(\mathbf{s}_{1:t})\}_{t=1}^T$ induced by $\{\psi_t^{\boldsymbol{\xi}^*}\}_{t=1}^T$ via Eq. (67) are different from $\sigma(\mathbf{s}_{1:t})$.*

Proof. To satisfy condition (i) of the current proposition, we define

$$\psi_\tau^{\boldsymbol{\xi}^*}(\mathbf{s}_{1:\tau}) \coloneqq \begin{cases} \sum_{\mathbf{s}_{\tau+1:T}} p_0(\mathbf{s}_{\tau+1:T}|\mathbf{s}_{1:\tau})\phi(\mathbf{s}_{1:T}) & \tau \neq t \\ c(\mathbf{s}_{1:t-1})\ \sum_{\mathbf{s}_{t+1:T}} p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t})\phi(\mathbf{s}_{1:T}) & \tau = t \end{cases} \tag{69}$$

which for all τ, yields optimal proposals: (i) $q^{\boldsymbol{\xi}^*}(s_\tau|\mathbf{s}_{1:\tau-1}) = \sigma(s_\tau|\mathbf{s}_{1:\tau-1}) \propto p_0(s_\tau|\mathbf{s}_{1:\tau-1})\psi_\tau^{\boldsymbol{\xi}^*}(\mathbf{s}_{1:\tau})$ via Lemma E.1. However, it is clear that $c(\mathbf{s}_{1:t-1}) \neq 1$ can break the necessary condition for optimality of SMC sampling that $\pi_t(\mathbf{s}_{1:t}) = \sigma(\mathbf{s}_{1:t})$ (Prop. A.4). In particular, consider

$$\begin{aligned} \pi_t^{\boldsymbol{\xi}^*}(\mathbf{s}_{1:t}) = \frac{1}{\mathcal{Z}_t^\psi}\, p_0(\mathbf{s}_{1:t})\, \psi_t^{\boldsymbol{\xi}^*}(\mathbf{s}_{1:t}) &= \frac{1}{\mathcal{Z}_t^\psi}\, c(\mathbf{s}_{1:t-1})p_0(\mathbf{s}_{1:t}) \sum_{\mathbf{s}_{t+1:T}} p_0(\mathbf{s}_{t+1:T}|\mathbf{s}_{1:t})\phi(\mathbf{s}_{1:T}) \\ &= \frac{1}{\mathcal{Z}_t^\psi} c(\mathbf{s}_{1:t-1})\tilde{\sigma}(\mathbf{s}_{1:t}) \neq \sigma(\mathbf{s}_{1:t}) \end{aligned} \tag{70}$$

for $c(\mathbf{s}_{1:t-1}) \neq 1$, which introduces an additional factor which depends on $\mathbf{s}_{1:t}$. Thus, the twist target $\pi_t^{\boldsymbol{\xi}^*}(\mathbf{s}_{1:t})$ induced from $\psi_t^{\boldsymbol{\xi}^*}(\mathbf{s}_{1:t})$ in Eq. (69) is not equal to the desired marginal $\sigma(\mathbf{s}_{1:t})$, despite the fact that all proposals are optimal. □

We indeed observed experimentally that resampling based on Eq. (67) after training using Eq. (EBM proposal learning) could lead to *worse* SMC $\log \mathcal{Z}_\sigma$ bounds than simply calculating the SIS or IWAE bound with $\prod_{t=1}^T q_t^{\boldsymbol{\xi}}(s_t|\mathbf{s}_{1:t-1})$.

Optimality in CTL Objective implies Optimal Twisted SMC In contrast to Prop. E.2, note that the global optimum of our CTL objective $\min \sum_{t=1}^{T} D_{\mathrm{KL}}\left(\sigma(\mathbf{s}_{1:t}) \,\middle\|\, \pi_t^{\psi}(\mathbf{s}_{1:t})\right)$ (which occurs for the optimal twists $\{\psi_t^*\}_{t=1}^{T-1}$ in Prop. 3.2), results in both the twist-induced proposal $q_t^{\pi^*}(s_t|\mathbf{s}_{1:t-1}) = \sigma(s_t|\mathbf{s}_{1:t-1})$ and the twisting targets $\pi_t^*(\mathbf{s}_{1:t}) = \sigma(\mathbf{s}_{1:t})$ satisfying the necessary and sufficient conditions for optimality outlined in App. A.1 Prop. A.3.

E.3.2. SMC WITH NORMALIZED TARGETS INDUCED BY LEARNED PROPOSAL LEADS TO UNIFORM WEIGHTS

The issue in Prop. E.2 arises from the degree of freedom $c(\mathbf{s}_{1:t-1})$ in the normalization constant of the one-step proposal. To avoid this, we can instead define *normalized* twisted intermediate targets using

$$\tilde{\pi}_t^{\boldsymbol{\xi}}(\mathbf{s}_{1:t}) = \begin{cases} p_0(\mathbf{s}_{1:t}) \prod_{\tau=1}^{t} \frac{\psi_\tau^{\boldsymbol{\xi}}(\mathbf{s}_{1:\tau})}{Z_\tau^{\boldsymbol{\xi}}(\mathbf{s}_{1:\tau-1})} = \prod_{\tau=1}^{t} q_\tau^{\boldsymbol{\xi}}(s_\tau|\mathbf{s}_{1:\tau-1}) & t \neq T \\ p_0(\mathbf{s}_{1:T})\, \phi(\mathbf{s}_{1:T}) & t = T \end{cases} \tag{71}$$

where $Z_t^{\boldsymbol{\xi}}(\mathbf{s}_{1:t-1})$ arises from $q_t^{\boldsymbol{\xi}}(s_t|\mathbf{s}_{1:t-1}) \coloneqq \frac{1}{Z_t^{\boldsymbol{\xi}}(\mathbf{s}_{1:t-1})} p_0(s_t|\mathbf{s}_{1:t-1})\psi_t^{\boldsymbol{\xi}}(\mathbf{s}_{1:t})$ learned with Eq. (EBM proposal learning).

Crucially, $\tilde{\pi}_t^{\boldsymbol{\xi}}$ in Eq. (71) are automatically normalized for $t \neq T$, as the product of normalized proposals. In this case, SMC resampling with $q^{\boldsymbol{\xi}}$ or the twist-induced proposal yields uniform resampling weights,

$$(\text{for } t < T): \; w_t(\mathbf{s}_{1:t}) = \frac{\tilde{\pi}_t^{\boldsymbol{\xi}}(\mathbf{s}_{1:t})}{\tilde{\pi}_{t-1}^{\boldsymbol{\xi}}(\mathbf{s}_{1:t-1}) q^{\boldsymbol{\xi}}(s_t|\mathbf{s}_{1:t-1})} = \frac{p_0(\mathbf{s}_{1:t}) \prod_{\tau=1}^{t} \frac{\psi_\tau^{\boldsymbol{\xi}}(\mathbf{s}_{1:\tau})}{Z_\tau^{\boldsymbol{\xi}}(\mathbf{s}_{1:\tau-1})}}{p_0(\mathbf{s}_{1:t-1}) \left(\prod_{\tau=1}^{t-1} \frac{\psi_\tau^{\boldsymbol{\xi}}(\mathbf{s}_{1:\tau})}{Z_\tau^{\boldsymbol{\xi}}(\mathbf{s}_{1:\tau-1})}\right) \frac{1}{Z_t^{\boldsymbol{\xi}}(\mathbf{s}_{1:t-1})} p_0(s_t|\mathbf{s}_{1:t-1})\psi_t^{\boldsymbol{\xi}}(\mathbf{s}_{1:t})} = 1 \tag{72}$$

Although we were able to construct well-behaved intermediate twisting targets from a proposal-learning scheme $q_t^{\boldsymbol{\xi}}(s_t|\mathbf{s}_{1:t-1}) \propto p_0(s_t|\mathbf{s}_{1:t-1})\psi_t^{\boldsymbol{\xi}}(\mathbf{s}_{1:t})$, Eq. (72) shows that this *does not lead to meaningful intermediate SMC resampling*. In other words, for $t < T$, the marginal distributions of SMC samples $\mathbf{s}_{1:t}^k$ with this scheme are simply $q^{\boldsymbol{\xi}}(\mathbf{s}_{1:t})$, the same as we would obtain with no resampling (SIS/IWAE).

F. Bidirectional SMC

In this section, we recall the extended state-space probabilistic interpretation of SMC from (Maddison et al., 2017; Andrieu et al., 2010). The idea is to define an unnormalized target distribution $\sigma_{\text{SMC}}(\boldsymbol{S})$ and normalized proposal $q_{\text{SMC}}(\boldsymbol{S})$ over an extended state space $\boldsymbol{S}$ containing all random variables relevant to SMC sampling and importance weighting with K sequences of length T. Defining $\tilde{\sigma}_{\text{SMC}}(\boldsymbol{S})$ such that its normalization constant matches $\mathcal{Z}_\sigma$, we can use simple importance sampling (SIS) in this extended state space to show that K-sequence SMC sampling yields an unbiased estimator of $\mathcal{Z}_\sigma$, for example $\mathcal{Z}_\sigma = \mathbb{E}_{q_{\text{SMC}}(\boldsymbol{S})}[\frac{\tilde{\sigma}_{\text{SMC}}(\boldsymbol{S})}{q_{\text{SMC}}(\boldsymbol{S})}]$ (as in Eq. (8)). Our end goal is to use this probabilistic interpretation to derive the lower and upper bounds on $\log \mathcal{Z}_\sigma$ in Prop. 5.1, following Brekelmans et al. (2021) App. A.

We define the extended state space proposal and target distributions below, noting that our bounds will require sampling from normalized $\sigma_{\text{SMC}}(\boldsymbol{S})$ or $q_{\text{SMC}}(\boldsymbol{S})$, and evaluating $\tilde{\sigma}_{\text{SMC}}(\boldsymbol{S})$ and $q_{\text{SMC}}(\boldsymbol{S})$. We summarize the algorithm for sampling $\sigma_{\text{SMC}}(\boldsymbol{S})$ in Alg. 2, using concatenation notation for simplicity (instead of index histories as in the text).

Single-Sequence Target and Proposal We construct our importance sampling bounds with the goal of estimating the (log) partition function and sampling from a target distribution $\sigma(\mathbf{s}_{1:T}) = \tilde{\sigma}(\mathbf{s}_{1:T})/\mathcal{Z}_\sigma$. We leverage a sequence of intermediate target distributions, $\{\pi_t(\mathbf{s}_{1:t}) = \frac{1}{\mathcal{Z}_t}\tilde{\pi}_t(\mathbf{s}_{1:t})\}_{t=1}^{T}$ over partial sequences, with the final target $\pi_T(\mathbf{s}_{1:T}) = \sigma(\mathbf{s}_{1:T})$ and $\mathcal{Z}_T = \mathcal{Z}_\sigma$. We assume $\tilde{\pi}_0(\mathbf{s}_0) = 1$ for all prompts with $\mathcal{Z}_0 = 1$. Finally, our bounds and sampling procedures also depend on a given set of proposal distribution $\{q(s_t \,|\, \mathbf{s}_{1:t-1})\}_{t=1}^{T}$, as in Sec. 2.2.

Extended State Space Random Variables Consider an extended state space $\boldsymbol{S}$ containing KT tokens $\{s_t^k\}_{t=1,k=1}^{T,K}$ with $s_t^k \in \mathcal{V}$ and KT indexing random variables $\{\omega_t^k\}_{t=1,k=1}^{T,K}$ with $\omega_t^k \in [1, K]$, to represent the results of resampling (Eq. (7)),

$$\boldsymbol{S} \coloneqq \left\{s_t^k, \omega_t^k\right\}_{t=1,k=1}^{T,K} \tag{73}$$

For ease of notation (and similarly to Maddison et al. (2017); Andrieu et al. (2010)), we call attention to our use of recursive backtracking index operations to collect sequences $\{\mathbf{s}_{1:t}\}$ based on the results of resampling $\{\omega_t^k\}$. We use *lists* of index

histories to construct sequences of tokens, with two recursive definitions of histories. Letting $+$ indicate appending of lists,

$$\begin{aligned} \boldsymbol{h}_0^k &\coloneqq [\,] \quad \forall k, \qquad \boldsymbol{h}_t^k \coloneqq \boldsymbol{h}_{t-1}^{\omega_t^k} + [\omega_t^k] \\ \bar{\boldsymbol{h}}_0^k &\coloneqq [\,] \quad \forall k, \qquad \bar{\boldsymbol{h}}_t^k \coloneqq \boldsymbol{h}_{t-1}^k + [k] \end{aligned} \tag{Index Notation}$$

For example, the history $\boldsymbol{h}_{t-1}^k$ will be used to construct prefix sequences $\mathbf{s}_{1:t-1}^{\boldsymbol{h}_{t-1}^k}$ (i.e. lists of tokens) for sampling a next token s_t^k. We denote sequences of tokens with the index history in the superscript and also expand the definition for clarity,

$$\begin{aligned} \mathbf{s}_{1:t}^{\boldsymbol{h}_t^k} &\coloneqq \mathbf{s}_{1:t-1}^{\boldsymbol{h}_{t-1}^{\omega_t^k}} + [s_t^{\omega_t^k}] \quad = [s_1^{\boldsymbol{h}_{t-1}^{\omega_t^k}[1]}, ..., s_{t-1}^{\boldsymbol{h}_{t-1}^{\omega_t^k}[t-1]}, s_t^{\omega_t^k}] = [s_1^{\omega_1^{\cdots^{\omega_t^k}}}, ..., s_{t-2}^{\omega_{t-1}^{\omega_t^k}}, s_{t-1}^{\omega_t^k}, s_t^{\omega_t^k}] \\ \mathbf{s}_{1:t}^{\bar{\boldsymbol{h}}_t^k} &\coloneqq \mathbf{s}_{1:t-1}^{\boldsymbol{h}_{t-1}^k} + [s_t^k] \end{aligned} \tag{Sequence Notations}$$

In the second line, we define $\mathbf{s}_{1:t}^{\bar{\boldsymbol{h}}_t^k}$ as a sequence of length t which concatenates the prefix $\mathbf{s}_{1:t}^{\boldsymbol{h}_{t-1}^k}$ with next token s_t^k. The notation $\mathbf{s}_{1:t}^{\bar{\boldsymbol{h}}_t^k}$ represents partial sequences *before* resampling. By contrast, we will use the notation $\mathbf{s}_{1:t}^{\boldsymbol{h}_t^k}$ in the first line of Eq. (Sequence Notations) to refer to sequences *after* resampling.

Consider the sequence $\mathbf{s}_{1:t}^{\bar{\boldsymbol{h}}_t^i}$ in a particular index $i \in [1, K]$ *before* resampling. Resampling at time t may result in choosing $\omega_t^k = i$ for some k. Using the first line, we see that $\mathbf{s}_{1:t}^{\boldsymbol{h}_t^k} = \mathbf{s}_{1:t-1}^{\boldsymbol{h}_{t-1}^{\omega_t^k}} + [s_t^{\omega_t^k}] = \mathbf{s}_{1:t-1}^{\boldsymbol{h}_{t-1}^i} + [s_t^i]$ for those indices such that $\omega_t^k = i$. Indeed, this matches the definition of $\mathbf{s}_{1:t}^{\bar{\boldsymbol{h}}_t^i} = \mathbf{s}_{1:t-1}^{\boldsymbol{h}_{t-1}^i} + [s_t^i]$ in the second line (before resampling). Thus, the indexing notation in Eq. (Sequence Notations) reflects resampling or cloning of sequences $\mathbf{s}_{1:t}^{\bar{\boldsymbol{h}}_t^i}$ into the indices such that $\omega_t^k = i$, which yields prefixes $\mathbf{s}_{1:t}^{\boldsymbol{h}_t^k}$ for the next step of sampling $(t+1)$ in each index $k \in [1, K]$.

Extended State Space Proposal Distribution Sampling from the extended state space proposal corresponds to the procedure described in Sec. 2.2 and Alg. 1, which we write as[6]

$$q_{\text{SMC}}\left(\{s_t^k, \omega_t^k\}_{t=1,k=1}^{T,K}\right) \coloneqq \prod_{t=1}^{T}\left[\prod_{k=1}^{K} q\left(s_t^k \,\middle|\, \mathbf{s}_{1:t-1}^{\boldsymbol{h}_{t-1}^k}\right) \prod_{k=1}^{K} q\left(\omega_t^k \mid \mathbf{s}_{1:t}^{1:K}\right)\right] \tag{SMC Extended Proposal}$$

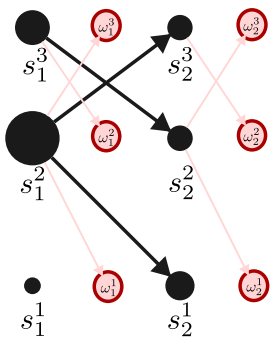
$$\text{where } \forall\, k, \quad q\left(\omega_t^k = i \mid \mathbf{s}_{1:t}^{1:K}\right) \coloneqq \frac{\dfrac{\tilde{\pi}_t\left(\mathbf{s}_{1:t}^{\bar{\boldsymbol{h}}_t^i}\right)}{\tilde{\pi}_{t-1}\left(\mathbf{s}_{1:t-1}^{\boldsymbol{h}_{t-1}^i}\right) q\left(s_t^i \,\middle|\, \mathbf{s}_{1:t-1}^{\boldsymbol{h}_{t-1}^i}\right)}}{\sum_{\kappa=1}^{K} \dfrac{\tilde{\pi}_t\left(\mathbf{s}_{1:t}^{\bar{\boldsymbol{h}}_t^\kappa}\right)}{\tilde{\pi}_{t-1}\left(\mathbf{s}_{1:t-1}^{\boldsymbol{h}_{t-1}^\kappa}\right) q\left(s_t^\kappa \,\middle|\, \mathbf{s}_{1:t-1}^{\boldsymbol{h}_{t-1}^\kappa}\right)}} = \frac{w_t\left(\mathbf{s}_{1:t}^{\bar{\boldsymbol{h}}_t^i}\right)}{\sum_{\kappa=1}^{K} w_t\left(\mathbf{s}_{1:t}^{\bar{\boldsymbol{h}}_t^\kappa}\right)} \tag{74}$$

To recount the description above, note that the next token s_t^i in index i is sampled from the proposal, conditioned on the prefix $\mathbf{s}_{1:t-1}^{\boldsymbol{h}_{t-1}^i}$. We concatenate these tokens $\mathbf{s}_{1:t}^{\bar{\boldsymbol{h}}_t^i} = \mathbf{s}_{1:t-1}^{\boldsymbol{h}_{t-1}^i} + [s_t^i]$ (Eq. (Sequence Notations)) and calculate importance weights. We perform resampling in each index k according to $q(\omega_t^k | \mathbf{s}_{1:t}^{1:K})$, or SNIS with the calculated weights (as in Eq. (7)). Finally, after resampling, we clone the sequence in the chosen index ω_t^k into index k and proceed to sample s_{t+1}^k with an prefix defined by the indices $\boldsymbol{h}_t^k = \boldsymbol{h}_{t-1}^{\omega_t^k} + [\omega_t^k]$.

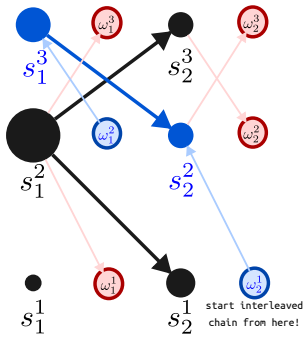
Worked Example: To make this more concrete, we provide a worked example of the procedure in Fig. 4 (a). At step $t = 1$, we resample the token $s_{t=1}^{k=2}$ twice (for indices $k = 1, 3$), with $\omega_1^1 = \omega_1^3 = 2$ (and in index 2, set $\omega_1^2 = 3$ to sample s_1^3). We record the prefix history as, for example, $\boldsymbol{h}_1^1 = \boldsymbol{h}_1^3 = [\omega_1^1] = [2]$, which corresponds to $\mathbf{s}_1^{\boldsymbol{h}_1^1} = s_1^2$.

At step 2 in (a), we proceed to sample $s_2^1 \sim q(s_2 | \mathbf{s}_1^{\boldsymbol{h}_1^1} = [s_1^2])$ (and similarly $s_2^3 \sim q(s_2 | \mathbf{s}_1^{\boldsymbol{h}_1^3} = [s_1^2])$), whereas $s_2^2 \sim q(s_2 | \mathbf{s}_1^{\boldsymbol{h}_1^1} = [s_1^3])$. We next evaluate the importance weights over three concatenated sequences: $\mathbf{s}_1^{\bar{\boldsymbol{h}}_1^1} = [s_1^2] + [s_2^1]$,

[6] Note that $\boldsymbol{h}_t^k$, $\mathbf{s}_{1:t}^{\bar{\boldsymbol{h}}_t^k}$, and $\mathbf{s}_{1:t}^{\boldsymbol{h}_t^k}$ are deterministically constructed from $\{s_t^k, \omega_t^k\}_{t=1,k=1}^{T,K}$ during sampling, and simply track the quantities to be calculated when evaluating densities.



(a) SMC Extended State-Space Proposal ($T = 2$)



(b) SMC Extended State-Space Target ($T = 2$)

Figure 4: Graphical Models for extended state-space proposal and target distributions which result in the bidirectional SMC bounds. We show density evaluation in the proposal and target for a fixed set of $\{s_t^k, \omega_t^k\}_{k=1,t=1}^{3,2}$. We let the size of the circles reflect the (hypothetical) importance weights of sequences $\mathbf{s}_{1:t}^{\bar{\boldsymbol{h}}_t^k}$ and ω_t^k reflect the (hypothetical) results of resampling with these weights. In (b), we assume fixed $j_{T+1} = j_3 = 1$ as in the text, with $\omega_2^1 = 2$.

Algorithm 2 (Twisted) SMC Target Sampling (σ_{SMC})
(blue indicates changes from SMC proposal algorithm; $\mathfrak{s}_{1:T}$ is an exact posterior sample)

SMC-TARGET$\left(p_0, q, \{\psi_t\}_{t=1}^{T-1}, \phi, K, \{t_r\}_{t=1}^{R-1}, t_0 = 0, t_R = T, \mathfrak{s}_{1:T}\right)$:
Initialize $j \sim \texttt{uniform}(\{1, \ldots, K\})$
for $t = 1, ..., T$ **do**
  for $k = 1, ..., K$ **do**
    if $k = j$ **then**
      $s_t^k \leftarrow \mathfrak{s}_t$
    else
      Sample $s_t^k \sim q\left(s_t \,\middle|\, \mathbf{s}_{1:t-1}^k\right)$
    end if
    $\mathbf{s}_{1:t}^k \leftarrow \texttt{concat}\left(\mathbf{s}_{1:t-1}^k, s_t^k\right)$
    if $t < T$ **then**
      $w_t^k \leftarrow \frac{p_0\left(s_t^k \,\middle|\, \mathbf{s}_{1:t-1}^k\right)}{q\left(s_t^k \,\middle|\, \mathbf{s}_{1:t-1}^k\right)} \frac{\psi_t\left(\mathbf{s}_{1:t}^k\right)}{\psi_{t-1}\left(\mathbf{s}_{1:t-1}^k\right)}$
    else
      $w_t^k \leftarrow \frac{p_0\left(s_t^k \,\middle|\, \mathbf{s}_{1:t-1}^k\right)}{q\left(s_t^k \,\middle|\, \mathbf{s}_{1:t-1}^k\right)} \frac{\phi\left(\mathbf{s}_{1:t}^k\right)}{\psi_{t-1}\left(\mathbf{s}_{1:t-1}^k\right)}$
    end if
  end for
  if $t \in \{t_r\}_{r=1}^{R-1}$ **then**
    $\bar{\mathbf{s}}_{1:t}^{1:K} \leftarrow \mathbf{s}_{1:t}^{1:K}.\texttt{copy}()$
    $\bar{j} \leftarrow j.\texttt{copy}()$
    Sample $j \sim \texttt{uniform}(\{1, \ldots, K\})$
    for $k = 1, ..., K$ **do**
      if $k = j$ **then**
        $\omega_t^k \leftarrow \bar{j}$
      else
        $\omega_t^k \sim \texttt{cat}\left(\left\{\frac{\prod_{t=t_{r-1}+1}^{t_r} w_t^i}{\sum_{j=1}^K \prod_{t=t_{r-1}+1}^{t_r} w_t^j}\right\}_{i=1}^K\right)$
      end if
      $\mathbf{s}_{1:t}^k \leftarrow \bar{\mathbf{s}}_{1:t}^{\omega_t^k}$
    end for
  end if
end for
return $\left\{\mathbf{s}_{1:T}^k, \prod_{t=t_{R-1}+1}^T w_t^k\right\}_{k=1}^K$
  $\check{\mathcal{Z}}_\sigma^{\text{SMC}} = \prod_{r=1}^R \frac{1}{K} \sum_{k=1}^K \prod_{t=t_{r-1}+1}^{t_r} w_t^k$

$\mathbf{s}_1^{\bar{\boldsymbol{h}}_1^2} = [s_1^3] + [s_2^2]$, and $\mathbf{s}_1^{\bar{\boldsymbol{h}}_1^3} = [s_1^2] + [s_2^3]$, emphasizing that s_2^k is the final token in each index. Shown in the red circles, we proceed to resample $\omega_2^1 = 2, \omega_2^2 = 3$, and $\omega_2^3 = 2$ at step $t = 2$.

Finally, we need to backtrack to obtain the history of the indices for the sequence to be cloned in resampling. Namely, for index 1 where $\omega_{t=2}^{k=1} = 2$, we concatenate $\boldsymbol{h}_1^{\omega_2^1} + [\omega_2^1] = \boldsymbol{h}_1^2 + [2] = [3, 2] =: \boldsymbol{h}_2^1$ (i.e. the history for time 2, index 1). This list of indices specifies the prefix $\mathbf{s}_{1:2}^{\boldsymbol{h}_2^1} = [s_1^3, s_2^2]$ at step $t = 3$, index $k = 1$. Similar reasoning applies for other indices.

Extended State Space Target We are finally ready to specify the extended state space target distribution. The crucial difference is to identify a single sequence $\mathbf{s}_{1:T}^{\boldsymbol{h}_T^1}$ of length T (the choice of index 1 is arbitrary). This sequence $\mathbf{s}_{1:T}^{\boldsymbol{h}_T^1}$ will be evaluated under the unnormalized target distribution $\tilde{\pi}_T(\mathbf{s}_{1:T}) = \tilde{\sigma}(\mathbf{s}_{1:T})$ or exactly sampled from the target $\mathbf{s}_{1:T}^{\boldsymbol{h}_T^1} \sim \sigma(\mathbf{s}_{1:T})$ in the extended state space target distribution.

In particular, we begin by sampling a full sequence of indices $\{j_t\}_{t=1}^T$ uniformly at random $\Pr(j_1, j_2, ...j_T) = (1/K)^T$. Setting $\omega_T^1 = j_T$, we let $\omega_{t-1}^{j_t} = j_{t-1}$ for all t. This implies the following,

$$\omega_T^1 = j_T,\ \omega_{t-1}^{j_t} = j_{t-1} \quad \Longrightarrow \quad \boldsymbol{h}_T^1 = [j_1, j_2, ...j_T], \qquad \boldsymbol{h}_{t-1}^{j_t} = [j_1, j_2, ...j_{t-1}], \tag{75}$$

$$\text{and} \qquad \bar{\boldsymbol{h}}_t^{j_t} = \boldsymbol{h}_t^{j_{t+1}} \tag{76}$$

To show these identities, note that $\omega_{t-1}^{j_t} = j_{t-1}$ and Eq. (Index Notation) imply $\boldsymbol{h}_{t-1}^{j_t} = \boldsymbol{h}_{t-2}^{\omega_{t-1}^{j_t}} + [\omega_{t-1}^{j_t}] = \boldsymbol{h}_{t-2}^{j_{t-1}} + [j_{t-1}] = \bar{\boldsymbol{h}}_{t-1}^{j_{t-1}}$, which matches Eq. (76). Applying this recursion again yields $\boldsymbol{h}_{t-1}^{j_t} = \boldsymbol{h}_{t-3}^{j_{t-2}} + [j_{t-2}, j_{t-1}]... = [j_1, j_2, ...j_{t-1}]$. Taken together, these notations allow us to interleave a true target sample in particular indices $\{j_t\}$, guaranteeing that at least one target samples appears at each step.

The extended state space target distribution differs from q_{SMC} in its handling of this sequence, which identified as $\mathbf{s}_{1:T}^{\boldsymbol{h}_T^1}$ with prefixes $\mathbf{s}_{1:t-1}^{\boldsymbol{h}_{t-1}^{j_t}}$ using Eq. (75). Noting that sampling $\{j_t\}_{t=1}^T$ amounts to specifying a particular set of ω_t^k as in Eq. (75)-(76),

$$\tilde{\sigma}_{\text{SMC}}\left(\{s_t^k, \omega_t^k\}_{t=1,k=1}^{T,K}\right) = \underbrace{\Pr(j_1, j_2, ...j_T)}_{\left(\frac{1}{K}\right)^T}\ \tilde{\pi}_T\left(\mathbf{s}_{1:T}^{\boldsymbol{h}_T^1}\right) \prod_{t=1}^T \left[\prod_{\substack{k=1\\k\neq j_t}}^K q\left(s_t^k \,\middle|\, \mathbf{s}_{1:t-1}^{\boldsymbol{h}_{t-1}^k}\right) \prod_{\substack{k=1\\k\neq j_{t+1}}}^K q\left(\omega_t^k \mid \mathbf{s}_{1:t}^{1:K}\right)\right]. \tag{SMC Extended Target}$$

Note, the normalization constant of $\tilde{\sigma}_{\text{SMC}}(\boldsymbol{S})$ is equal to $\mathcal{Z}_\sigma$ since only $\tilde{\pi}_T(\mathbf{s}_{1:T}) = \tilde{\sigma}(\mathbf{s}_{1:T})$ is unnormalized.

To describe ancestral sampling from Eq. (SMC Extended Target), we first sample $\{j_t\}_{t=1}^T$ uniformly as above, and place an exact target sequence in indices $\mathbf{s}_{1:T}^{\boldsymbol{h}_T^1}$ (or, equivalently, sequentially sample $s_t^{j_t} \sim \pi_t(s_t|\mathbf{s}_{1:t-1}^{\boldsymbol{h}_{t-1}^{j_t}})$. At each step, the remaining $K-1$ indices $k \neq j_t$ are sampled from the proposal. For resampling, we fix index j_t to hold the exact sample and resample the remaining $K-1$ indices. Note that the resampling weights $q\left(\omega_t^k \mid \mathbf{s}_{1:t}^{1:K}\right)$ in Eq. (74) *include* the exact sample, which may be cloned additional times into indices other than j_t if its importance weights are high. The procedure above simply ensures that *at least* one exact sequence is sampled. See Alg. 2 for the pseudocode of the algorithm.

Note that Maddison et al. (2017, Alg. 2) presents a different SMC extended state space target distribution than ours. In their work, $j_1 = 1$ and they sample $\mathbf{j}_{2:T+1}$, while in ours $j_{T+1} = 1$ and we sample $\mathbf{j}_{1:T}$. However, both targets result in the same log partition function bounds.

Worked Example: In Fig. 2 (c), we use blue circles and arrows to highlight the exact-sample indices $\boldsymbol{h}_T^1 = [j_1, j_2] = [3, 2]$ and the target sequence $\mathbf{s}_{1:T}^{\boldsymbol{h}_T^1} = [s_1^3, s_2^2]$. Using the recursion $\omega_{t-1}^{j_t} = j_{t-1}$ with $j_{T+1} = j_3 = 1$ fixed, we may also express $\boldsymbol{h}_T^1 = [j_1, j_2] = [3, 2] = [\omega_1^2, \omega_2^1]$. At step 2, note the target sequence is sampled/evaluated an additional time in index 3.

Importance Weights in the Extended State Space Assume we are given a fixed set of $\{s_t^k, \omega_t^k\}_{t=1,k=1}^{T,K}$, which may be sampled from either $\sigma_{\text{SMC}}(\boldsymbol{S})$ or $q_{\text{SMC}}(\boldsymbol{S})$. We proceed to show that the unnormalized importance weights in the extended state space simplify as follows.

Lemma F.1. *For the extended state space target $\tilde{\sigma}_{\text{SMC}}$ and proposal q_{SMC} above, the simple importance weights in the extended state space become*

$$\frac{\tilde{\sigma}_{\text{SMC}}}{q_{\text{SMC}}}\left(\{s_t^k, \omega_t^k\}_{t=1,k=1}^{T,K}\right) = \prod_{t=1}^T \frac{1}{K} \sum_{k=1}^K \frac{\tilde{\pi}_t\left(\mathbf{s}_{1:t}^{\bar{\boldsymbol{h}}_t^k}\right)}{\tilde{\pi}_{t-1}\left(\mathbf{s}_{1:t-1}^{\boldsymbol{h}_{t-1}^k}\right) q\left(s_t^k \,\middle|\, \mathbf{s}_{1:t-1}^{\boldsymbol{h}_{t-1}^k}\right)} = \prod_{t=1}^T \frac{1}{K} \sum_{k=1}^K w_t\left(\mathbf{s}_{1:t}^{\bar{\boldsymbol{h}}_t^k}\right) =: \prod_{t=1}^T \frac{1}{K} \sum_{k=1}^K w_t(\mathbf{s}_{1:t}^k) \tag{77}$$

which can be used to obtain unbiased $\mathcal{Z}_\sigma$ estimators (Eq. (8)) or bounds on $\log \mathcal{Z}_\sigma$ (Prop. 5.1, with proof below).

Proof. To evaluate the importance weights (with the goal of estimating $\mathcal{Z}_\sigma$), we consider

$$\frac{\tilde{\sigma}_{\text{SMC}}}{q_{\text{SMC}}}\left(\{s_t^k, \omega_t^k\}_{t=1,k=1}^{T,K}\right) = \frac{\left(\frac{1}{K}\right)^T \tilde{\pi}_T\left(\mathbf{s}_{1:T}^{\boldsymbol{h}_T^1}\right) \prod_{t=1}^T \left[\prod_{\substack{k=1 \\ k\neq j_t}}^K q\left(s_t^k \,\middle|\, \mathbf{s}_{1:t-1}^{\boldsymbol{h}_{t-1}^k}\right) \prod_{\substack{k=1 \\ k\neq j_{t+1}}}^K q\left(\omega_t^k \,|\, \mathbf{s}_{1:t}^{1:K}\right)\right]}{\prod_{t=1}^T \left[\prod_{k=1}^K q\left(s_t^k \,\middle|\, \mathbf{s}_{1:t-1}^{\boldsymbol{h}_{t-1}^k}\right) \prod_{k=1}^K q\left(\omega_t^k \,|\, \mathbf{s}_{1:t}^{1:K}\right)\right]} \tag{78}$$

$$\overset{(1)}{=} \left(\frac{1}{K}\right)^T \tilde{\pi}_T\left(\mathbf{s}_{1:T}^{\boldsymbol{h}_T^1}\right) \prod_{t=1}^T \frac{1}{q\left(s_t^{j_t} \,\middle|\, \mathbf{s}_{1:t-1}^{\boldsymbol{h}_{t-1}^{j_t}}\right) q\left(\omega_t^{j_{t+1}} \,\middle|\, \mathbf{s}_{1:t}^{1:K}\right)} \tag{79}$$

where in (1), note that terms in the denominator cancel except for the indices $[0, j_1, \ldots j_T] = \boldsymbol{h}_T^1$. Recalling that $\omega_t^{j_{t+1}} = j_t$ from Eq. (76), we expand the resampling weights $q(j_t|\mathbf{s}_{1:t}^{1:K})$ for the sequence indexed by $s_t^{j_t}$, $\mathbf{s}_{1:t-1}^{\boldsymbol{h}_{t-1}^{j_t}}$, and $\mathbf{s}_{1:t-1}^{\bar{\boldsymbol{h}}_t^{j_t}}$,

$$\overset{(2)}{=} \left(\frac{1}{K}\right)^T \tilde{\pi}_T\left(\mathbf{s}_{1:T}^{\boldsymbol{h}_T^1}\right) \prod_{t=1}^T \frac{\sum_{k=1}^K \frac{\tilde{\pi}_t\left(\mathbf{s}_{1:t}^{\bar{\boldsymbol{h}}_t^k}\right)}{\tilde{\pi}_{t-1}\left(\mathbf{s}_{1:t-1}^{\boldsymbol{h}_{t-1}^k}\right) q\left(s_t^k \,\middle|\, \mathbf{s}_{1:t-1}^{\boldsymbol{h}_{t-1}^k}\right)}}{\cancel{q\left(s_t^{j_t} \,\middle|\, \mathbf{s}_{1:t-1}^{\boldsymbol{h}_{t-1}^{j_t}}\right)} \frac{\tilde{\pi}_t\left(\mathbf{s}_{1:t}^{\bar{\boldsymbol{h}}_t^{j_t}}\right)}{\cancel{\tilde{\pi}_{t-1}\left(\mathbf{s}_{1:t-1}^{\boldsymbol{h}_{t-1}^{j_t}}\right) q\left(s_t^{j_t} \,\middle|\, \mathbf{s}_{1:t-1}^{\boldsymbol{h}_{t-1}^{j_t}}\right)}}} \tag{80}$$

Finally, we obtain a telescoping cancellation of $\tilde{\pi}_t$ terms using the indexing identities in Eq. (75)-(76). In particular, since $\bar{\boldsymbol{h}}_t^{j_t} = \boldsymbol{h}_t^{j_{t+1}}$ and $\bar{\boldsymbol{h}}_{t-1}^{j_{t-1}} = \boldsymbol{h}_{t-1}^{j_t}$ with $\bar{\boldsymbol{h}}_T^{j_T} = \boldsymbol{h}_T^1$, we can simplify the terms in Eq. (80) as

$$\tilde{\pi}_T\left(\mathbf{s}_{1:T}^{\boldsymbol{h}_T^1}\right) \prod_{t=1}^T \frac{\tilde{\pi}_{t-1}\left(\mathbf{s}_{1:t-1}^{\boldsymbol{h}_{t-1}^{j_t}}\right)}{\tilde{\pi}_t\left(\mathbf{s}_{1:t}^{\bar{\boldsymbol{h}}_t^{j_t}}\right)} = \tilde{\pi}_T\left(\mathbf{s}_{1:T}^{\bar{\boldsymbol{h}}_T^{j_T}}\right) \prod_{t=1}^T \frac{\tilde{\pi}_{t-1}\left(\mathbf{s}_{1:t-1}^{\bar{\boldsymbol{h}}_{t-1}^{j_{t-1}}}\right)}{\tilde{\pi}_t\left(\mathbf{s}_{1:t}^{\bar{\boldsymbol{h}}_t^{j_t}}\right)} = \cancel{\tilde{\pi}_T\left(\mathbf{s}_{1:T}^{\bar{\boldsymbol{h}}_T^{j_T}}\right)} \cancel{\frac{\tilde{\pi}_{T-1}\left(\mathbf{s}_{1:T-1}^{\bar{\boldsymbol{h}}_{T-1}^{j_{T-1}}}\right)}{\tilde{\pi}_T\left(\mathbf{s}_{1:T}^{\bar{\boldsymbol{h}}_T^{j_T}}\right)}} \cancel{\frac{\tilde{\pi}_{T-2}\left(\mathbf{s}_{1:T-2}^{\bar{\boldsymbol{h}}_{T-2}^{j_{T-2}}}\right)}{\tilde{\pi}_{T-1}\left(\mathbf{s}_{1:T-1}^{\bar{\boldsymbol{h}}_{T-1}^{j_{T-1}}}\right)}} \cdots \frac{1}{\cancel{\tilde{\pi}_1\left(\mathbf{s}_{1:1}^{\bar{\boldsymbol{h}}_1^{j_1}}\right)}} = 1$$

using the assumption that $\tilde{\pi}_0(\cdot) = 1$. Simplifying from Eq. (80), the final unnormalized importance weights become

$$\frac{\tilde{\sigma}_{\text{SMC}}}{q_{\text{SMC}}}\left(\{s_t^k, \omega_t^k\}_{t=1,k=1}^{T,K}\right) = \prod_{t=1}^T \frac{1}{K} \sum_{k=1}^K \frac{\tilde{\pi}_t\left(\mathbf{s}_{1:t}^{\bar{\boldsymbol{h}}_t^k}\right)}{\tilde{\pi}_{t-1}\left(\mathbf{s}_{1:t-1}^{\boldsymbol{h}_{t-1}^k}\right) q\left(s_t^k \,\middle|\, \mathbf{s}_{1:t-1}^{\boldsymbol{h}_{t-1}^k}\right)} = \prod_{t=1}^T \frac{1}{K} \sum_{k=1}^K w_t\left(\mathbf{s}_{1:t}^{\bar{\boldsymbol{h}}_t^k}\right) =: \prod_{t=1}^T \frac{1}{K} \sum_{k=1}^K w_t(\mathbf{s}_{1:t}^k) \tag{81}$$

as desired, where we abbreviate the importance weights as $w_t(\mathbf{s}_{1:t}^k)$ for simplicity of notation. Note that we also obtain an unbiased estimate of the partition function via

$$\mathcal{Z}_\sigma = \mathbb{E}_{q_{\text{SMC}}(\boldsymbol{S})}\left[\frac{\tilde{\sigma}_{\text{SMC}}(\boldsymbol{S})}{q_{\text{SMC}}(\boldsymbol{S})}\right] = \mathbb{E}_{q_{\text{SMC}}(\boldsymbol{S})}\left[\prod_{t=1}^T \frac{1}{K} \sum_{k=1}^K w_t\left(\mathbf{s}_{1:t}^k\right)\right]$$

□

Proposition 5.1. (Bidirectional SMC Bounds) *The log partition function* $\log \mathcal{Z}_\sigma$ *of a target distribution* $\sigma(\mathbf{s}_{1:T})$ *can be lower and upper bounded by*

$$\begin{aligned} \mathbb{E}_{q_{\text{SMC}}(\boldsymbol{S})}\left[\log \prod_{t=1}^T \frac{1}{K} \sum_{i=1}^K w_t\left(\mathbf{s}_{1:t}^i\right)\right] &\leq \log \mathcal{Z}_\sigma \\ \log \mathcal{Z}_\sigma &\leq \mathbb{E}_{\sigma_{\text{SMC}}(\boldsymbol{S})}\left[\log \prod_{t=1}^T \frac{1}{K} \sum_{i=1}^K w_t\left(\mathbf{s}_{1:t}^i\right)\right]. \end{aligned} \tag{23}$$

The gap in the lower bound is $D_{\text{KL}}(q_{\text{SMC}}(\boldsymbol{S}) \,\|\, \sigma_{\text{SMC}}(\boldsymbol{S}))$*, and the gap in the upper bound is* $D_{\text{KL}}(\sigma_{\text{SMC}}(\boldsymbol{S}) \,\|\, q_{\text{SMC}}(\boldsymbol{S}))$.

Proof. The proof follows directly from Brekelmans et al. (2021) App. A, where it is shown that for $\sigma_{\text{ext}}(\boldsymbol{S}), q_{\text{ext}}(\boldsymbol{S})$ such that $\mathcal{Z}_\sigma = \mathbb{E}_{q_{\text{ext}}(\boldsymbol{S})}[\frac{\tilde{\sigma}_{\text{ext}}(\boldsymbol{S})}{q_{\text{ext}}(\boldsymbol{S})}]$, we can construct lower and upper bounds on $\log \mathcal{Z}_\sigma$

$$D_{\text{KL}}(q_{\text{ext}}(\boldsymbol{S}) \,\|\, \sigma_{\text{ext}}(\boldsymbol{S})) + \mathbb{E}_{q_{\text{ext}}(\boldsymbol{S})}\left[\log \frac{\tilde{\sigma}_{\text{ext}}(\boldsymbol{S})}{q_{\text{ext}}(\boldsymbol{S})}\right] = \log \mathcal{Z}_\sigma = \mathbb{E}_{\sigma_{\text{ext}}(\boldsymbol{S})}\left[\log \frac{\tilde{\sigma}_{\text{ext}}(\boldsymbol{S})}{q_{\text{ext}}(\boldsymbol{S})}\right] - D_{\text{KL}}(\sigma_{\text{ext}}(\boldsymbol{S}) \,\|\, q_{\text{ext}}(\boldsymbol{S})) \tag{82}$$

$$\mathbb{E}_{q_{\text{ext}}(\boldsymbol{S})}\left[\log \frac{\tilde{\sigma}_{\text{ext}}(\boldsymbol{S})}{q_{\text{ext}}(\boldsymbol{S})}\right] \leq \log \mathcal{Z}_\sigma \leq \mathbb{E}_{\sigma_{\text{ext}}(\boldsymbol{S})}\left[\log \frac{\tilde{\sigma}_{\text{ext}}(\boldsymbol{S})}{q_{\text{ext}}(\boldsymbol{S})}\right] \tag{83}$$

where the gap in the lower and upper bounds are $D_{\text{KL}}(q_{\text{ext}}(\boldsymbol{S}) \,\|\, \sigma_{\text{ext}}(\boldsymbol{S}))$ and $D_{\text{KL}}(\sigma_{\text{ext}}(\boldsymbol{S}) \,\|\, q_{\text{ext}}(\boldsymbol{S}))$, respectively.

Substituting our SMC probabilistic interpretation in Eq. (SMC Extended Proposal) and Eq. (SMC Extended Target), along with the importance weights in Lemma F.1, into Eq. (83) yields the desired bounds in Eq. (23). □

IWAE as a Special Case of our SMC Probabilistic Interpretation Note that we recover IWAE (or SIS over K samples) from SMC with no intermediate resampling. In particular, this corresponds to $\omega_t^k = k$ for all $t < T$, with importance weighting from resampling occurring at the final step $\prod_{k=1}^K q(\omega_T^k | \mathbf{s}_{1:T}^{1:K})$. This yields the $1/K$ average inside the log in the IWAE bounds (i.e., SMC with only one resampling step at $t = T$). While the importance weights are crucial to construct the bound, note that 'resampling' is not necessary at the final step and we may return all K samples along with their weights.

Viewing IWAE as a special case of our SMC probabilistic interpretation is complementary to the interpretations in Domke & Sheldon (2018); Brekelmans et al. (2021) and also provides upper bounds (Sobolev & Vetrov, 2019).

G. Additional Experiment Details

G.1. Common Details Across Experiments

For all experiments, we use the Adam optimizer with $\beta_1, \beta_2 = \{0.9, 0.999\}$. We use custom implementations of SMC. For PPO, we use the HuggingFace TRL PPO Trainer (`https://github.com/huggingface/trl/blob/main/trl/trainer/ppo_trainer.py`), modified slightly to accomodate our custom twist parameterizations, as described below. For other methods, we use Optax (Flax) and custom loss functions. We use HuggingFace models (`https://huggingface.co/models`) for the base p_0 models and build custom layers on top of those.

For the twist $\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})$, we always parameterize $\log \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})$ for numerical stability. We choose random normal initializations centered at mean 0, with low variance,[7] such that $\log \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}) \approx 0, \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}) \approx 1$ at the beginning of training, which means the initial sequences generated by the twist-induced proposal approximately come from the base model p_0. All methods are initialized using the same random seeds, and thus start from the same parameter values. See App. G.2 for additional discussion of choices for the twist parameterization.

For methods that directly learn a proposal (DPG and PPO), we could directly finetune a language model that outputs $q(\mathbf{s}_{1:t})$. However, in order to ensure consistency in terms of model capacity and ease of learning compared to our twisted proposals, we instead have these proposal learning methods output a modifier $\log \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})$ which is added to the base model log probability $\log p_0(\mathbf{s}_{1:t})$. Note that using random normal initializations centered at mean 0 with low variance, this scheme results in initial q samples coming approximately from p_0.

For methods that can make use of exact posterior samples, when we have access to them (Sec. 7.2.3, App. H.3), we use them. This is straightforward for methods like DPG, SIXO, and our CTL (unless we have only a single sample, as we discuss for infilling in App. G.4). For our RL twist learning, we found the best empirical performance training on a combination of q and exact σ samples when they were available (as opposed to just q otherwise), and use those results. Similarly, for FUDGE, when exact σ samples are available, we use them together with p_0 samples.

It is not straightforward to compare PPO versus other methods, because of the inner loop in PPO that repeats several clipped gradient steps on a given set of samples. This means that, for a constant number of samples, PPO makes more gradient updates than other methods, while for a constant number of gradient updates, PPO sees fewer samples. Ultimately we decided to normalize based on the number of samples seen; we consider each outer step (including a full PPO inner loop, in our experiments, 4 gradient steps) as a single "gradient update." We make this choice since sampling is the main bottleneck in terms of computational cost, and the number of inner PPO steps is a hyperparameter which we did not tune.

[7] We specifically use a form of Xavier initialization, taking the variance as $\frac{2}{n_{\text{inputs}}+n_{\text{outputs}}}$.

All of our experiments were run on a single GPU, usually on an NVIDIA A40 with 48G memory. All experiments took no longer than 9 wall-clock hours to run for a single learning method, with infilling (Sec. 7.2.3) experiments taking longest; most other experiments took no longer than 4 hours.

G.2. Choices of Twist Parameterization

The choice of parameterization for the twist $\log \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})$ is a design decision, independent of our overall framework. While one could keep an entirely separate model for each $\log \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})$, this is likely to be memory-inefficient and learn slowly. Instead, we use a shared parameterization across $\mathbf{s}_{1:t}$, in the same way that the base language model uses a single architecture to output probability distributions over tokens at each time step t. We lay out parameterization choices we considered below.

G.2.1. Linear Head

The simplest choice is to replace the linear head of the base language model with a new linear head, keep the base model fixed, and only train the linear head. This parameterization incurs very little additional computation cost compared to just using the base language model. However, we found this to be capacity constrained in our experiments, achieving worse KL divergences than other parameterizations.

G.2.2. MLP Head

Instead of a linear head, we consider a 3-layer fully connected neural network (MLP) with ReLU non-linearities as a head on top of the base language model. The base model is still kept fixed; only the MLP head is trained. This incurs more computational cost than a linear head (App. G.2.1), but the additional cost is still small relative to the cost of a forward pass through the base transformer model. We found this to generally perform well in our experiments, so we use it for the toxicity threshold experiment in Sec. 7.1 and sentiment in Sec. 7.2.2.

G.2.3. Separate Transformer for the Twist

We can also consider an entirely separate transformer that outputs only the twist value. That is, we copy the base model, and repurpose it to output a twist value $\log \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})$ instead of logits for next-token probabilities. We then train the entire network end-to-end. This is significantly more computationally costly than the former approaches, and does not always do better than just an MLP head (App. G.2.2), so we generally do not recommend using this. Still, we found it to perform well in toxicity classification in Sec. 7.2.1, so we use it there.

G.2.4. Separate Transformer for the Twist, with MLP Head

This is similar to App. G.2.3, except we also replace the final linear head with a MLP head as in App. G.2.2. The model outputs $\log \psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t})$ and is trained end-to-end. This is the most computationally costly approach outlined here, and is unnecessary for most of our settings. However, in infilling with 15 generated tokens (Sec. 7.2.3) we found this parameterization to perform materially better than all others, particularly with DPG (App. E.3), so we use it for all infilling experiments.

With both this parameterization and App. G.2.3, we increase computation time by a factor of around 2 on the forward pass, and significantly increase memory and time usage on the backwards pass during training (though sampling is still the main time bottleneck). Whether this parameterization is worth the potential gain in performance depends on the desired use case. We emphasize that our overall framework is independent of the choice of parameterization.

G.3. Comments on Our Choices of Experiment Settings

Our settings and evaluation metrics in Sec. 7 are chosen to highlight our scientific findings. In particular, the toxicity threshold experiment in Sec. 7.1 demonstrates the improvement of SMC over SIS with the base model with CTL learned twists. In order to highlight this distinction, we have chosen a setting where it is *extremely* difficult to draw samples satisfying the threshold using the base model p_0 (see SIS/IWAE LB line in Fig. 3).

However, twist-learning in the toxicity threshold setting presents challenges. For approximate positive sampling and a thresholded target, all importance weights will be 0 if none of our K samples meet the threshold. As noted above, sampling from p_0, or the SMC/twisted proposal for $\psi_t^{\boldsymbol{\theta}}(\mathbf{s}_{1:t}) \approx 1$ at initialization, is extremely unlikely to draw samples meeting the threshold (i.e., within the support of the target) in the setting of Sec. 7.1. As a result, initial iterations of twist learning receive no learning signal until a thresholded positive sample is drawn from the base model.

To avoid this difficulty for baselines comparisons in Sec. 7.2, we instead focused on settings with $\phi(\mathbf{s}_{1:T})$ given by probabilities. Nevertheless, we note that there are no fundamental differences between the settings considered in Sec. 7.1

and Sec. 7.2. Thus, we may also evaluate single-sample $D_{\text{KL}}(\sigma \,\|\, q)$ and $D_{\text{KL}}(q \,\|\, \sigma)$ in the setting of Sec. 7.1, or plot $\log \mathcal{Z}_\sigma$ bounds as a function of K in for the settings in Sec. 7.2.

G.4. Experiment-Specific Details

Details for SIS and SMC Comparison (Sec. 7.1) We generate 10 output tokens, and train twists using Sec. 4.1 with approximate positive sampling as discussed in Sec. 4.1.2.

Note that using $\sigma(\mathbf{s}_{1:T}) \propto p_0(\mathbf{s}_{1:T})\mathbb{I}[\mathbf{s}_{1:T} \in \mathcal{C}]$ where $\mathcal{C} \coloneqq \{\mathbf{s}_{1:T} \,|r(\mathbf{s}_{1:T}) \leq \eta\}$ directly runs into numerical issues for calculating $\log \mathcal{Z}_\sigma$ when $\mathbf{s}_{1:T} \notin \mathcal{C}$ and $\mathbb{I}[\mathbf{s}_{1:T} \in \mathcal{C}] = 0$. We instead use $\epsilon + \mathbb{I}[\mathbf{s}_{1:T} \in \mathcal{C}]$ everywhere instead of $\mathbb{I}[\mathbf{s}_{1:T} \in \mathcal{C}]$, where $\epsilon = 10^{-16}$. In Fig. 3, this yields a SIS/IWAE $\log \mathcal{Z}_\sigma$ LB ≈ -36 when no samples are drawn that fall in the set $\mathcal{C}$.

We use an MLP head to parameterize the twist, as in App. G.2.2, with 768 hidden units per layer, matching the TinyStories model's embedding dimension. We use a batch size (number of SMC particles/samples) of 1000, with a learning rate of 0.0001, and train using CTL for a total of 5000 gradient updates. We did not tune hyperparameters because we found this setting to work well, and we are not comparing across different learning methods.

For each point on each line on Fig. 3, we run SIS or SMC 20 times, each with a different randomly selected true posterior sample for the upper bounds. The line shows the average value across these 20 runs, while the shaded area shows 95% confidence intervals. See also App. G.1 for details common across experiments.

Details for Toxicity (Sec. 7.2.1) We generate 20 output tokens. We parameterize the twist with a separate network as in App. G.2.3. We use a batch size (number of SMC particles/samples) of 100, and train for a total of 2048 gradient updates. For each learning method, we used a coarse grid search over learning rates between 0.000001 and 0.001, using the best one found, which was usually 0.00003 or 0.0001. We run each learning method over 5 different random seeds, reporting the average KL divergence and 95% confidence intervals over these 5 seeds.

For each KL divergence evaluation, we first get sandwich bounds on $\log \mathcal{Z}_\sigma$ as laid out in Sec. 5, using the learned twists for the twisted proposal with $K = 500$ samples. We find SIS/IWAE and SMC bounds to be similarly tight, so use SIS/IWAE for simplicity. We do this 4 times, providing 4 upper bound estimates and 4 lower bound estimates, and take the average midpoint as the $\log \mathcal{Z}_\sigma$ estimate for each experiment. We then take the median (across all learning methods and seeds) of these estimates, and use that as our estimate of $\log \mathcal{Z}_\sigma$. This is then used as a common value for the KL divergence across all methods and seeds, which controls for possible noise in $\log \mathcal{Z}_\sigma$ bounds and ensures a fair comparison across methods. We generally have tight bounds (upper bound $\approx$ lower bound), which suggest our $\log \mathcal{Z}_\sigma$ estimates are generally accurate, but note that any inaccuracies in estimating $\log \mathcal{Z}_\sigma$ would only affect the absolute values of the KL divergences, not the relative differences among different learning methods.

We estimate expectations in Eq. (22) with 2000 samples from q and 2000 exact posterior samples for σ. With 2000 samples, our estimates have 95% confidence intervals generally between 0.05 and 0.10, suggesting that our estimates of expectations are unlikely to be off by more than 0.10. The exact posterior samples were collected offline; such a large number of samples takes several hours to collect, and in practical settings, we would likely only be able to collect a much smaller number of samples. All our methods still apply with fewer exact posterior samples, but the variance in estimates will be higher. See also App. G.1 for details common across experiments.

Details for Sentiment (Sec. 7.2.2) We generate 10 output tokens. We parameterize the twist using an MLP head (App. G.2.2), with 1024 hidden units per layer, matching the GPT2Medium model's embedding dimension. Other details are the same as for toxicity above. Collecting exact posterior samples is less time consuming in this case (less than an hour). See App. G.1 for common experimental details.

Details for Infilling (Sec. 7.2.3) We parameterize the twist using a separate transformer with an MLP head (App. G.2.4), with 768 hidden units per layer (matching the TinyStories model's embedding dimension). We make the following adjustments to the forward pass of the language model for the conditional twist setting. Instead of taking in only $\mathbf{s}_{1:T}$, the model takes in both $\mathbf{s}_{1:T}$ and $\mathbf{s}_{T+1:T+c}$ and passes each separately through the body (everything except the head). Thus, $\mathbf{s}_{T+1:T+c}$ can be seen as a second prompt. For $\mathbf{s}_{T+1:T+c}$, we take the embeddings produced after the last conditioning token s_{T+c} has been processed, broadcast it across time steps $1:T$, and pass that as additional input to the MLP head (concatenated with embeddings for $\mathbf{s}_{1:T}$ at each $t \in 1...T$). This allows the MLP head to produce different output depending on the conditioning tokens.

Since we are in the conditional target distribution setting (Sec. 3.3), with $o_T = \mathbf{s}_{T+1:T+c}$, to compare across learning methods using a single quantity, we estimate $\mathbb{E}_{o_T}[D_{\mathrm{KL}}(q_{o_T} \,\|\, \sigma_{o_T})] := \mathbb{E}_{o_T}[D_{\mathrm{KL}}(q(\mathbf{s}_{1:T}|o_T) \,\|\, \sigma(\mathbf{s}_{1:T}|o_T))]$ and $\mathbb{E}_{o_T}[D_{\mathrm{KL}}(\sigma_{o_T} \,\|\, q_{o_T})] := \mathbb{E}_{o_T}[D_{\mathrm{KL}}(\sigma(\mathbf{s}_{1:T}|o_T) \,\|\, q(\mathbf{s}_{1:T}|o_T))]$ where $\mathbb{E}_{o_T}[\cdot] := \mathbb{E}_{p_0(\mathbf{s}_{T+1:T+c})}[\cdot]$ for infilling. Note that,

$$\mathbb{E}_{o_T}[D_{\mathrm{KL}}(q(\mathbf{s}_{1:T}|o_T) \,\|\, \sigma(\mathbf{s}_{1:T}|o_T))] = \mathbb{E}_{o_T}\left[\mathbb{E}_{q(\mathbf{s}_{1:T}|o_T)}\left[\log \frac{q(\mathbf{s}_{1:T}|o_T)}{p_0(\mathbf{s}_{1:T})\phi(\mathbf{s}_{1:T}, o_T)}\right]\right] + \mathbb{E}_{o_T}[\log \mathcal{Z}_\sigma(o_T)]$$

$$\mathbb{E}_{o_T}[D_{\mathrm{KL}}(\sigma(\mathbf{s}_{1:T}|o_T) \,\|\, q(\mathbf{s}_{1:T}|o_T))] = \mathbb{E}_{o_T}\left[\mathbb{E}_{\sigma(\mathbf{s}_{1:T}|o_T)}\left[\log \frac{p_0(\mathbf{s}_{1:T})\phi(\mathbf{s}_{1:T}, o_T)}{q(\mathbf{s}_{1:T}|o_T)}\right]\right] - \mathbb{E}_{o_T}[\log \mathcal{Z}_\sigma(o_T)]$$

where for a fixed o_T, $\mathbb{E}_{q(\mathbf{s}_{1:T}|o_T)}\left[\log \frac{q(\mathbf{s}_{1:T}|o_T)}{p_0(\mathbf{s}_{1:T})\phi(\mathbf{s}_{1:T}, o_T)}\right]$ and $\mathbb{E}_{\sigma(\mathbf{s}_{1:T}|o_T)}\left[\log \frac{p_0(\mathbf{s}_{1:T})\phi(\mathbf{s}_{1:T}, o_T)}{q(\mathbf{s}_{1:T}|o_T)}\right]$ may be evaluated as before, similar to the unconditional setting. In particular, for our experiments, we use 1-sample estimates of these expectations, as we have a single exact sample from $\sigma(\mathbf{s}_{1:T}|o_T)$ by the BDMC trick (Sec. 3.3), and we choose to draw a single sample from the conditional proposal $q(\mathbf{s}_{1:T}|o_T)$. We average this over 2000 $o_T \sim p_0(\mathbf{s}_{T+1:T+c})$, approximating the outer expectation, giving us a 2000-sample estimate of 1-sample estimates for the first term in the right hand side of both equations above. With 2000 samples, our estimates have 95% confidence intervals generally between 0.20 and 0.30.

Note that $\mathbb{E}_{o_T}[\log \mathcal{Z}_\sigma(o_T)]$ is independent of the learning method or proposal q, unlike the first term we discussed above. Thus, in order to save computation and provide us with a more accurate estimate of $\mathbb{E}_{o_T}[\log \mathcal{Z}_\sigma(o_T)]$, we estimate this term only once. Specifically, we consider only the learning method with the lowest KL divergence (DPG), and use SIS/IWAE bounds. For each o_T, we estimate $\log \mathcal{Z}_\sigma(o_T)$ with $K = 500$ samples, which gives us relatively tight sandwich bounds, again taking the midpoint as our estimate. We average this over 1000 $o_T \sim p_0(\mathbf{s}_{T+1:T+c})$, giving us a 1000-sample estimate of $\mathbb{E}_{o_T}[\log \mathcal{Z}_\sigma(o_T)]$, where each $\log \mathcal{Z}_\sigma(o_T)$ is itself estimated via 500 samples.

For negative sampling with contrastive twist learning (CTL) in this setting, we need at least 2 negative samples per set of conditioning tokens $o_T = \mathbf{s}_{T+1:T+c}$ to perform SIS reweighting; this is in contrast with other twist learning methods which can generate a single negative sample per o_T. For the positive sample, we can use our single exact sample directly, or we can run the SMC upper bound sampling procedure ("Sampling from σ_{SMC} for SMC Upper Bounds" section in Sec. 5.2) generate more approximate σ samples using the given exact sample. We find the latter to generally perform slightly better than the former, so adopt that for our infilling experiments.

We use a fixed batch size of 100 across all methods for training twists. To clarify the meaning of this batch size, for methods other than CTL, we have 100 draws of exact σ samples, each for a different set of conditioning tokens $o_T = \mathbf{s}_{T+1:T+c}$, so we train over 100 different o_T at a time using 1 negative sample per o_T. For CTL, since we need at least 2 negative samples per o_T, we split the batch size of 100 across the number of different o_T and the number of negative samples per o_T, as an additional hyperparameter. We use 25 o_T with 4 negative samples per o_T for the experiments in Sec. 7.2.3 and 10 o_T with 10 negative samples per o_T for the experiments in App. H.2. Controlling for batch size in this way is arguably disadvantageous for CTL compared to other learning methods, as it learns on a smaller number of o_T, but this controls for memory requirements, and we feel is more fair than controlling for the number of o_T seen but allowing more negative samples for CTL relative to other methods. We train for a total of 5500 gradient updates. For each method, we used a coarse grid search over learning rates between 0.000001 and 0.001, using the best one found, which was usually 0.0001 or 0.00003. We run each learning method over 5 different random seeds, reporting the average KL divergence and 95% confidence intervals over these 5 seeds. See also App. G.1 for details common across experiments.

H. Additional Experimental Results

H.1. Qualitative Results

Toxicity Controlled Generation when No Exact Posterior Samples are Available In Sec. 7.2.1 we targeted $\sigma(\mathbf{s}_{1:T}) \propto p_0(\mathbf{s}_{1:T})e^{\beta \log p(a|\mathbf{s}_{1:T})}$ with $\beta = 1$. We can also target $\beta > 1$; higher β produces a more peaked distribution of text that is more likely to be of class a. However, for $\beta \neq 1$ we can no longer generate exact posterior samples and thus cannot upper bound $\log \mathcal{Z}_\sigma$. Our twist learning (Sec. 4.1) with approximate positive sampling (Sec. 4.1.2) can learn meaningful twists in this setting, which we illustrate with a qualitative example of a story (200 tokens upper limit) and $\beta = 10$:

"Once upon a time, there was a little girl named Lily. She had a big thumb that she liked to suck on. One day, Lily went to the park to play with her friends. She was having so much fun until her thumb got stuck in her shoe. She tried to pull it out, but it hurt too much.
Lily started to cry and her friends tried to help her, but they couldn't get her thumb out either. She was scared and didn't know what to do. Her friends tried to help her, but they couldn't get it out either. Sadly, Lily had to go to the hospital and get a big bandage on her thumb. She couldn't play with her friends anymore. From that day on, Lily never went to the park again."

Table 6: Qualitative Results - Reviews Very Likely to be of a Particular Rating

Class (Rating)	Text Generated Using Twisted SMC
1-star	"I bought this sucker for my wife to use on her python that she sent me last year. It was terrible!"
2-star	"I bought this throat raiser for combating dental caries. I didn't really like it. I didn't like"
3-star	"I bought this a few months back, and I enjoyed it every time I held it. I'm giving 3 stars"
4-star	"I bought this product a few months ago and have really enjoyed it. Only reason I gave it 4 stars is because"
5-star	"I bought this phone recently, and I've been loving it! Gorgeous design, outstanding battery life, fantastic camera"

Table 7: Qualitative Results - Infilling Examples

Proposal	Prompt ($\mathbf{s}_0$)	Generated Tokens ($\mathbf{s}_{1:T}$)	Conditioning Tokens ($\mathbf{s}_{T+1:T+c}$)
DPG	Once upon a time, there was a	little girl named Mia. She had a big heart. Mia loved to help	others and make them feel safe. Mia liked to
SIXO	Once upon a time, there was a	girl named Mia. Mia was very kind and compassionate. She always helped her	others and make them feel safe. Mia liked to
CTL	Once upon a time, there was a	girl named Mia. She had a thin, pink dress. Mia liked to	others and make them feel safe. Mia liked to

The story is coherent and follows the general style of the TinyStories base model, while having a high probability ($\approx 88\%$) of being toxic according to the toxicity classifier, likely due to the presence of negative words such as 'suck', 'hurt', 'cry', and 'scared'. This supports the ability of our methods to control outputs based on the chosen posterior distribution.

Sentiment Controlled Generation when No Exact Posterior Samples are Available As above, we also consider $\sigma(\mathbf{s}_{1:T}) \propto p_0(\mathbf{s}_{1:T})e^{\beta \log p(a|\mathbf{s}_{1:T})}$, where $\beta > 1$, except now $p(a|\mathbf{s}_{1:T})$ is based on the sentiment classifier in Sec. 7.2.2. In Table 6 we provide qualitative examples showing 20 tokens produced with twisted SMC with 500 particles, for $\beta = 100$, using twists trained with Sec. 4.1. These illustrate our framework's ability to learn reviews that embody each rating class.[8]

Infilling In Table 7 we compare qualitative results on an example set of conditioning tokens for DPG, SIXO, and CTL (in that order, to reflect increasing KL divergence). The qualitative results correlate with the quantitative measures of KL divergence; the lowest KL divergence (DPG) corresponds to infilled tokens that respect grammar and the topic. SIXO, which has higher KL divergence, fails to respect grammar. CTL generates incorrect grammar and is less on-topic, corresponding to the highest KL divergence among these methods.

H.2. Infilling with Fewer Tokens

We consider the same setting as Sec. 7.2.3 but only generating 2 tokens, conditioned on 1 token. We show KL divergence evaluations in Table 8. Our evaluation reveals interesting differences among learning methods, even in this easier setting where most methods achieve low KL divergence in both directions. DPG and RL learns best, while FUDGE learns notably slower. PPO suffers on $D_{\mathrm{KL}}(\sigma \,\|\, q)$, though this may be unsurprising since PPO does not make use of exact σ samples.

H.3. Approximate vs. Exact Posterior Sampling

In our toxicity and sentiment experiments, we train using approximate σ samples to reflect the more common real-world setting where the amount of exact samples needed for training are not available. However, here we run an additional ablation experiment for insight into the effect of positive versus approximate sampling. We use rejection sampling (Sec. 4.1.2) to generate exact posterior samples for training. This is much slower than generating approximate samples, so is not a practical strategy for training; we investigate this solely for understanding.

We provide a comparison of KL divergences (evaluated the same way as in the main paper) when training using exact versus approximate σ samples for a selection of methods that performed well in our previous experiments and are able to make use of σ samples. Toxicity (Sec. 7.2.1) results are in Table 9 and sentiment (Sec. 7.2.2) results are in Table 10. The first two columns of KL divergences are for exact σ samples. The next two are for training on the same number of samples, but using approximate positive sampling (Sec. 4.1.2). Overall, for a constant number of samples, having exact σ samples improves performance for most methods. Note however that there is an additional time cost required for rejection sampling to generate exact samples, so the exact σ training requires significantly more wall-clock time for any given number of samples.

We also plot the single-sample KL divergence in both directions as a function of training time for exact vs. approximate sampling, on toxicity and sentiment experiments, in Fig. 5. The approximate sampling results match those in the main paper (with different colors). The exact σ sample results cut off earlier because the time cost required for rejection sampling reduces the number of gradient updates that can be made for a given amount of wall-clock time.

[8]The results are slightly incoherent; this is a result of the base GPT2-Medium model often being incoherent. Qualitatively, we find that these generations are more coherent than the uncontrolled ones from p_0.

Table 8: KL Divergences (averaged over conditioning tokens drawn from the base model) for Infilling Experiments (Sec. 7.2.3) with 2 Output Tokens and 1 Conditioning Token

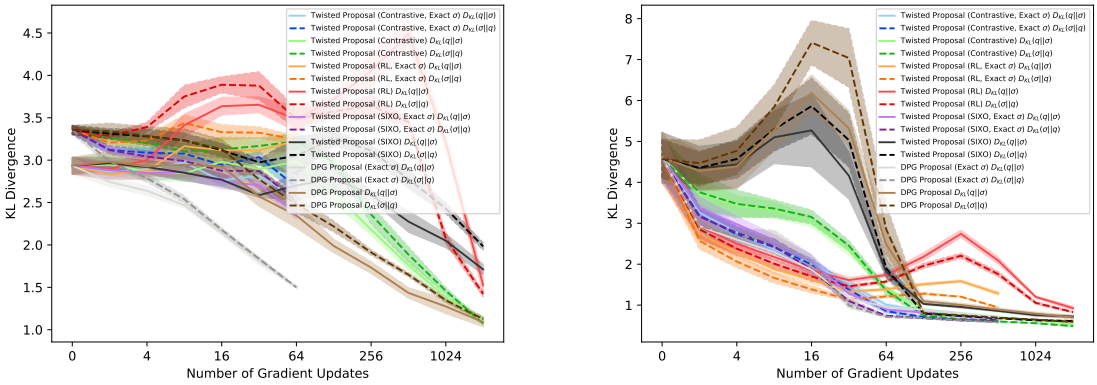
Proposal q_{o_T}	Twist Learning	$\mathbb{E}_{o_T}[D_{\mathrm{KL}}(q_{o_T} \,\|\, \sigma_{o_T})]$	$\mathbb{E}_{o_T}[D_{\mathrm{KL}}(\sigma_{o_T} \,\|\, q_{o_T})]$
Twisted	Contrastive	0.47 ± 0.10	0.25 ± 0.01
Twisted	RL	0.42 ± 0.10	0.15 ± 0.01
Twisted	SIXO	0.47 ± 0.11	0.25 ± 0.02
Twisted	FUDGE	2.62 ± 0.33	0.90 ± 0.02
DPG	–	$\mathbf{0.16 \pm 0.07}$	$\mathbf{0.14 \pm 0.01}$
PPO	–	0.52 ± 0.04	1.09 ± 0.34

Table 9: KL Div. for Toxicity Experiments (Sec. 7.2.1), comparing exact σ samples versus approximate positive sampling.

Proposal q	Type of Twist Learning	Exact σ Samples $D_{\mathrm{KL}}(q \,\|\, \sigma)$	Exact σ Samples $D_{\mathrm{KL}}(\sigma \,\|\, q)$	Same # of Approx. σ Samples $D_{\mathrm{KL}}(q \,\|\, \sigma)$	Same # of Approx. σ Samples $D_{\mathrm{KL}}(\sigma \,\|\, q)$
Twisted	Contrastive	2.54 ± 0.02	2.68 ± 0.09	2.99 ± 0.18	3.22 ± 0.09
Twisted	RL	3.23 ± 0.10	3.24 ± 0.04	3.48 ± 0.15	3.49 ± 0.13
Twisted	SIXO	2.37 ± 0.06	2.52 ± 0.05	2.70 ± 0.17	3.05 ± 0.22
DPG	–	1.51 ± 0.01	1.50 ± 0.01	2.35 ± 0.15	2.48 ± 0.10

Table 10: KL Div. for Sentiment Experiments (Sec. 7.2.2), comparing exact σ samples versus approximate positive sampling.

Proposal $q(\mathbf{s})$	Type of Twist Learning	Exact σ Samples $D_{\mathrm{KL}}(q \,\|\, \sigma)$	Exact σ Samples $D_{\mathrm{KL}}(\sigma \,\|\, q)$	Same # of Approx. σ Samples $D_{\mathrm{KL}}(q \,\|\, \sigma)$	Same # of Approx. σ Samples $D_{\mathrm{KL}}(\sigma \,\|\, q)$
Twisted	Contrastive	0.71 ± 0.02	0.64 ± 0.02	0.70 ± 0.02	0.60 ± 0.01
Twisted	RL	1.28 ± 0.05	0.94 ± 0.02	2.09 ± 0.08	1.76 ± 0.07
Twisted	SIXO	0.68 ± 0.02	0.60 ± 0.01	0.86 ± 0.02	0.68 ± 0.01
DPG	–	0.70 ± 0.02	0.58 ± 0.01	0.89 ± 0.03	0.69 ± 0.00



(a) Toxicity (Sec. 7.2.1) (b) Sentiment (Sec. 7.2.2)

Figure 5: Training comparison for Exact versus Approximate σ (positive) sampling, as described in App. H.3. Having access to exact target samples makes learning lead to lower KL divergences in a more reliable manner.