COMPOSE ONCE, SAMPLE MANY: AMORTIZED COMPOSITION OF DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Composing pretrained diffusion models enables flexible inference-time generation without retraining, but existing composition methods are expensive when many samples are required from the same target distribution: each new sample repeatedly queries all constituent models throughout a long sampling trajectory. We introduce **AmCo (Amortized Composition)**, a two-stage framework that pays this composition cost only for a small set of initial samples and reuses the resulting information for subsequent generation. In Stage 1, AmCo applies an existing composition method to obtain a small set of anchor samples together with their constituent density values. In Stage 2, these anchors are used to approximate the clean prediction of the target composed distribution, while a single constituent model provides a finite-sample correction. Consequently, Stage 2 requires only one constituent-model evaluation per sampling step, and supports substantially shorter deterministic sampling trajectories. Experiments on classifier-free-guided image generation and product-of-experts ligand generation show that AmCo preserves or improves generation quality and diversity while substantially reducing inference cost.

1 INTRODUCTION

Composing pretrained diffusion models provides a flexible way to construct new generative distributions at inference time, without retraining the underlying models. Given diffusion models associated with distributions $q_0^{(1)}, \dots, q_0^{(I)}$, one may wish to sample from a target distribution of the form

$$\pi_0(x) \propto F\left[q_0^{(1)}(x), \dots, q_0^{(I)}(x)\right], \tag{1}$$

where F specifies a desired composition. Such formulations encompass product-of-experts generation, classifier-free guidance, logical composition, and more general inference-time control (Du et al., 2023; Skreta et al., 2025a;b; Blohm & Garg, 2026; Zhang et al., 2026). A central difficulty is that directly combining the constituent scores generally does not produce the diffusion process associated with the desired composed data distribution. Recent work therefore develops increasingly accurate inference-time procedures based on MCMC, sequential Monte Carlo, density estimation, or corrected diffusion dynamics (Wu et al., 2023; Skreta et al., 2025a;b; He et al., 2026a;b; Ren et al., 2026; Lee et al., 2026).

These methods, however, can be computationally expensive when many samples are required from the same composed distribution. A composition method typically queries multiple constituent diffusion models at every sampling step, so generating one sample using T steps and I constituent models requires on the order of TI model evaluations. More importantly, essentially the same composition computation is repeated for every additional sample.

In this work, we ask a simple question: must the full cost of composition be paid for every generated sample? We introduce AmCo (Amortized Composition), a two-stage framework that amortizes expensive compositional inference over multiple samples. In Stage 1, AmCo applies an existing composition method to obtain a small set of M anchor samples and stores the constituent density information accumulated along these trajectories. In Stage 2, AmCo reuses these anchors to approximate the clean prediction of the target composed distribution through an importance-weighted estimator. Because the constituent density values needed by this estimator have already been computed

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

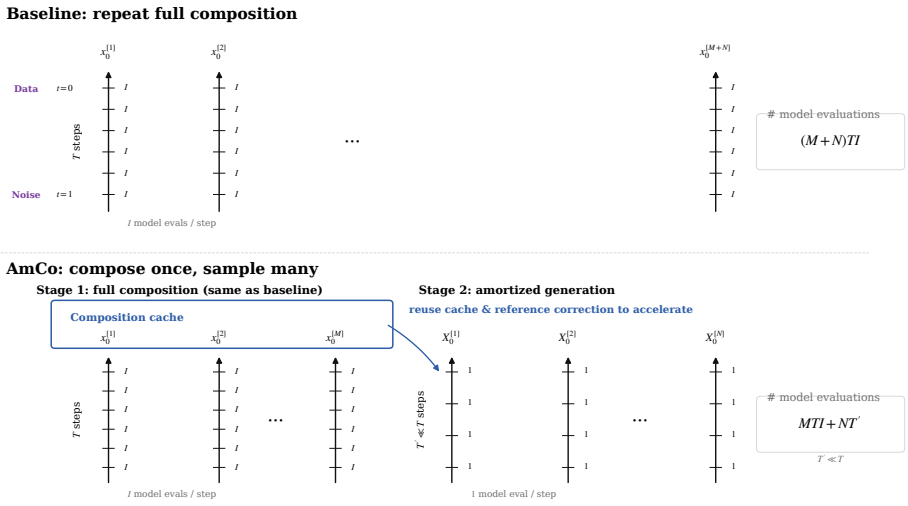


Figure 1: **Overview of AmCo.** While a standard composition method repeats the full T-step composition procedure for every sample, AmCo performs it only for M Stage-1 samples and stores the resulting composition cache. The cache and reference correction are then used to generate N additional samples with only one model evaluation per step and $T' \ll T$ steps, reducing the model-evaluation cost from $(M+N)TI$ to $MTI + NT'$.

in Stage 1, evaluating the amortized prediction itself requires no additional calls to all constituent models.

A finite set of anchors is insufficient near the data endpoint, where the importance weights can become concentrated and the resulting prediction may remain too close to the Stage-1 samples. We address this with a lightweight residual correction based on a single constituent diffusion model. Consequently, each Stage-2 step requires only one constituent-model evaluation rather than I, while retaining information about the full composition through the cached anchors. The resulting clean prediction also supports deterministic ODE-style sampling, allowing Stage 2 to use a substantially shorter trajectory of $T' \ll T$ steps. Thus, for M initial samples followed by N amortized samples, the dominant model-evaluation cost changes from approximately

$$(M+N)TI \qquad \text{to} \qquad MTI + NT'. \tag{2}$$

See 1 for visual illustration.

Unlike approaches that amortize diffusion inference by training an additional model (Venkatraman et al., 2024; Thornton et al., 2025), AmCo requires no additional learned sampler: amortization is achieved directly through a small reusable set of composition trajectories.

We evaluate AmCo in two distinct compositional settings. First, we apply it to classifier-free-guided image generation using CREPE (He et al., 2026b) as the base composition method. Second, we consider product-of-experts ligand generation using Feynman–Kac Correctors (Skreta et al., 2025a). Across both settings, AmCo substantially reduces the cost of producing additional samples while maintaining competitive generation quality and diversity.

Our contributions are:

- We introduce AmCo, a general framework for amortizing inference-time composition of pretrained diffusion models over multiple generated samples.
- We derive an amortized clean-prediction estimator together with a finite-sample correction that requires only one constituent-model evaluation per Stage-2 step and enables short deterministic sampling trajectories.

2 Background and Problem Setup

2.1 Stochastic Interpolants

Let q_0 be a data distribution on $\mathbb{R}^d$. We consider the stochastic interpolant

$$X_t = \alpha_t Z + \beta_t X_0, \qquad X_0 \sim q_0, \quad Z \sim \mathcal{N}(0, I), \tag{3}$$

where $t \in [0, 1]$, $\alpha_0 = 0$, $\beta_0 = 1$, $\alpha_1 \gg \beta_1 \geq 0$, with α_t monotonically increasing and β_t monotonically decreasing. This induces a probability path $\{q_t\}_{t\in[0,1]}$ from the data distribution toward noise.

We use three equivalent representations of this probability path. The *clean prediction*, the *score function*, and the *velocity field* are

$$m_t(x) := \mathbb{E}[X_0 \mid X_t = x], \tag{4}$$
$$s_t(x) := \nabla_x \log q_t(x), \tag{5}$$
$$v_t(x) := \mathbb{E}[\dot{\alpha}_t Z + \dot{\beta}_t X_0 \mid X_t = x]. \tag{6}$$

For the stochastic interpolant in equation 3, these quantities are analytically related through α_t and β_t. Therefore, given any one of m_t, s_t, or v_t, the other two can be recovered. We primarily work with the clean-prediction parameterization throughout the paper.

2.2 Compositional Sampling

Suppose we are given I pretrained diffusion models with data distributions $q_0^{(1)}, q_0^{(2)}, \ldots, q_0^{(I)}$.

Given a tractable function $F : (\mathbb{R}_{>0})^I \to \mathbb{R}_{>0}$, our goal is to sample from

$$\pi_0(x) \propto \widetilde{\pi}_0(x) := F[q_0^{(1)}(x), \ldots, q_0^{(I)}(x)], \tag{7}$$

assuming that $\int \widetilde{\pi}_0(x)\, dx < \infty$.

We define the target probability path by applying a stochastic interpolant to π_0:

$$\pi_t(x_t) = \int \pi_0(x_0)\mathcal{N}(x_t; \beta_t x_0, \alpha_t^2 I)\, dx_0. \tag{8}$$

Its clean prediction is

$$m_t^\pi(x_t) = \frac{\int \widetilde{\pi}_0(x_0)\mathcal{N}(x_t; \beta_t x_0, \alpha_t^2 I) x_0\, dx_0}{\int \widetilde{\pi}_0(x_0)\mathcal{N}(x_t; \beta_t x_0, \alpha_t^2 I)\, dx_0}. \tag{9}$$

If m_t^π were available, we could directly sample along the target probability path. The difficulty is that this conditional expectation is generally intractable.

3 Amortized Composition

3.1 Stage 1: Composition Cache

We first apply an existing composition method to the target distribution $\pi_0(x)$. Stage 1 produces M samples

$$x_0^{[m]} \sim \pi_0, \qquad m = 1, \ldots, M, \tag{10}$$

together with the constituent density values $q_0^{(1)}(x_0^{[m]}), \ldots, q_0^{(I)}(x_0^{[m]})$. (These density estimates can be tracked using the estimator of He et al. (2026a), without additional diffusion-model evaluations, and therefore incur negligible additional computational cost.)

3.2 Stage 2: Amortized Clean Prediction

We now reuse the Stage-1 samples to approximate the clean prediction of the target path. Rewriting the expectation defining m_t^π under the target distribution π_0 gives

$$m_t^\pi(X_t) = \frac{\mathbb{E}_{x_0\sim\pi_0}[w_t^\pi(x_0, X_t)x_0]}{\mathbb{E}_{x_0\sim\pi_0}[w_t^\pi(x_0, X_t)]}, \tag{11}$$

where, up to a multiplicative constant independent of x_0,

$$w_t^\pi(x_0, X_t) = \mathcal{N}(X_t; \beta_t x_0, \alpha_t^2 I). \tag{12}$$

Using the M samples obtained in Stage 1, we approximate this expectation by

$$\hat{m}_t^\pi(X_t) = \frac{\sum_{m=1}^M w_t^\pi(x_0^{[m]}, X_t)x_0^{[m]}}{\sum_{m=1}^M w_t^\pi(x_0^{[m]}, X_t)}. \tag{13}$$

Crucially, evaluating $\hat{m}_t^\pi$ requires only the cached Stage-1 anchors and Gaussian transition weights, with no additional evaluations of the constituent diffusion models.

3.3 Finite-Sample Correction

With a finite number of Stage-1 samples, the importance weights in the amortized estimator become increasingly concentrated as t approaches the data endpoint. Consequently, $\hat{m}_t^\pi(X_t)$ may become dominated by a nearby Stage-1 sample, causing newly generated samples to remain overly close to the Stage-1 set Scarvelis et al. (2025).

To mitigate this finite-sample effect while retaining the computational advantage of Stage 2, we use one constituent diffusion model as a reference at each sampling step. For model i, let

$$m_t^{(i)}(x) := \mathbb{E}_{q_{0|t}^{(i)}}[X_0 \mid X_t = x] \tag{14}$$

denote its exact clean prediction. Since

$$m_t^{(i)}(X_t) = \frac{\mathbb{E}_{x_0\sim\pi_0}[w_t^{(i)}(x_0, X_t)x_0]}{\mathbb{E}_{x_0\sim\pi_0}[w_t^{(i)}(x_0, X_t)]}, \tag{15}$$

where

$$w_t^{(i)}(x_0, X_t) = \frac{q_0^{(i)}(x_0)}{F[q_0^{(1)}(x_0), \ldots, q_0^{(I)}(x_0)]}\mathcal{N}\Big(X_t; \beta_t^{(i)}x_0, (\alpha_t^{(i)})^2 I\Big), \tag{16}$$

we construct the corresponding importance-weighted estimate using the same Stage-1 samples

$$\hat{m}_t^{(i)}(X_t) = \frac{\sum_{m=1}^M w_t^{(i)}(x_0^{[m]}, X_t)x_0^{[m]}}{\sum_{m=1}^M w_t^{(i)}(x_0^{[m]}, X_t)}. \tag{17}$$

We then correct the target clean-prediction estimate using the discrepancy between the exact and finite-sample predictions of the reference model:

$$\hat{m}_t(X_t) = \hat{m}_t^\pi(X_t) + \lambda_t(X_t)\left(m_t^{(i)}(X_t) - \hat{m}_t^{(i)}(X_t)\right). \tag{18}$$

Natural objective is to find $\lambda_t(X_t)$ which minimizes the sum of error and regularization term

$$\mathbb{E}\big[\|m_t^\pi(X_t) - \hat{m}_t(X_t)\|^2\big] + \eta(\lambda_t(X_t) - \lambda_0)^2 \tag{19}$$

$$= \mathbb{E}\big[\|\big(m_t^\pi(X_t) - \hat{m}_t^\pi(X_t)\big) - \lambda_t(X_t)\big(m_t^{(i)}(X_t) - \hat{m}_t^{(i)}(X_t)\big)\|^2\big] + \eta(\lambda_t(X_t) - \lambda_0)^2. \tag{20}$$

We estimate the solution using the method described in Appendix A.1. Description of our method is in Alg 1.

Algorithm 1 AmCo

// Stage 1
Use any base composition method to obtain $x_0^{[m]} \sim \pi_0(x_0)$ and $q_0^{(1)}(x_0^{[m]}), q_0^{(2)}(x_0^{[m]}), \cdots, q_0^{(I)}(x_0^{[m]})$ $(m = 1, 2, \cdots, M)$.
for $m \in \{1, 2, \cdots, M\}$ **do**
    $\tilde{\pi}_0(x_0^{[m]}) \leftarrow F[q_0^{(1)}(x_0^{[m]}), q_0^{(2)}(x_0^{[m]}), \cdots, q_0^{(I)}(x_0^{[m]})]$
end for
// Stage 2
for $n \in \{1, 2, \cdots, N\}$ **do**
    $X_1^{[n]} \overset{iid}{\sim} \mathcal{N}(0, \alpha_1^2 I)$
    for $t \in \{1, 1 - \Delta t, \cdots, \Delta t\}$ **do**
        $m_t^{(i)}(X_t^{[n]}) \leftarrow \frac{X_t^{[n]} + (\alpha_t^{(i)})^2 s_t^{(i)}(X_t^{[n]})}{\beta_t^{(i)}}$
        $\{w_t^{\pi}(x_0^{[m]}, X_t^{[n]})\}_{m=1}^M \leftarrow \{\mathcal{N}(X_t^{[n]}; \beta_t x_0^{[m]}, (\alpha_t)^2 I)\}_{m=1}^M$
        $\hat{m}_t^{\pi}(X_t^{[n]}) \leftarrow \frac{\sum_{m=1}^M w_t^{\pi}(x_0^{[m]}; X_t^{[n]}) x_0^{[m]}}{\sum_{m=1}^M w_t^{\pi}(x_0^{[m]}; X_t^{[n]})}$
        $\{w_t^{(i)}(x_0^{[m]}, X_t^{[n]})\}_{m=1}^M \leftarrow \{\frac{q_0^{(i)}(x_0^{[m]}) \mathcal{N}(X_t^{[n]}; \beta_t^{(i)} x_0^{[m]}, (\alpha_t^{(i)})^2 I)}{\tilde{\pi}_0(x_0^{[m]})}\}_{m=1}^M$
        $\hat{m}_t^{(i)}(X_t^{[n]}) \leftarrow \frac{\sum_{m=1}^M w_t^{(i)}(x_0^{[m]}; X_t^{[n]}) x_0^{[m]}}{\sum_{m=1}^M w_t^{(i)}(x_0^{[m]}; X_t^{[n]})}$
        Compute $\hat{\lambda}_t^{\mathrm{JK}}(X_t^{[n]})$ by Appendix A.1.
        $\hat{m}_t(X_t^{[n]}) \leftarrow \hat{m}_t^{\pi}(X_t) + \hat{\lambda}_t^{\mathrm{JK}}(X_t)[m_t^{(i)}(X_t) - \hat{m}_t^{(i)}(X_t)]$
        $X_{t-\Delta t}^{[n]} \leftarrow \frac{\alpha_{t-\Delta t}}{\alpha_t} X_t^{[n]} + (\beta_{t-\Delta t} - \frac{\alpha_{t-\Delta t}}{\alpha_t} \beta_t) \hat{m}_t(X_t^{[n]})$
    end for
end for
return $\{X_0^{[n]}\}_{n=1}^N$

3.4 Finite-Cache Error Analysis

We analyze the finite-cache error of the corrected clean-prediction estimator used in Stage 2. Recall that both $\hat{m}_t^{\pi}(X_t)$ and $\hat{m}_t^{(i)}(X_t)$ are self-normalized importance-sampling estimators constructed from the same M Stage-1 samples. Consequently, their finite-sample errors are generally correlated, and the reference correction can exploit this shared error.

For fixed $t \in (0, 1)$ and X_t, define

$$\psi_t^{\pi}(x_0; X_t) = \frac{w_t^{\pi}(x_0, X_t)}{\mathbb{E}_{\pi_0}[w_t^{\pi}(x_0, X_t)]} \big(x_0 - m_t^{\pi}(X_t)\big), \tag{21}$$

$$\psi_t^{(i)}(x_0; X_t) = \frac{w_t^{(i)}(x_0, X_t)}{\mathbb{E}_{\pi_0}[w_t^{(i)}(x_0, X_t)]} \big(x_0 - m_t^{(i)}(X_t)\big). \tag{22}$$

We assume that the corresponding weight normalizers are strictly positive and that the above influence functions have finite second moments.

Theorem 1 (Asymptotic error of the corrected predictor)**.** *For fixed $t \in (0, 1)$ and X_t, suppose $\hat{\lambda}_{t,M}(X_t) \to \lambda_t(X_t)$ in probability as $M \to \infty$. Then*

$$\hat{m}_t(X_t) (:= \hat{m}_t^{\pi}(X_t) + \hat{\lambda}_{t,M}(X_t)[m_t^{(i)}(X_t) - \hat{m}_t^{(i)}(X_t)]) \xrightarrow{p} m_t^{\pi}(X_t), \tag{23}$$

and

$$\sqrt{M}\big(\hat{m}_t(X_t) - m_t^{\pi}(X_t)\big) \Rightarrow \mathcal{N}\big(0, \Sigma_t^{\lambda}(X_t)\big), \tag{24}$$

where

$$\Sigma_t^{\lambda}(X_t) = \mathrm{Var}_{x_0 \sim \pi_0}\left[\psi_t^{\pi}(x_0; X_t) - \lambda_t(X_t) \psi_t^{(i)}(x_0; X_t)\right]. \tag{25}$$

Theorem 1 shows that the reference correction preserves the $M^{-1/2}$ convergence rate of the finite-cache estimator while modifying its asymptotic covariance. In particular, since the target and refer-

ence estimators reuse the same Stage-1 cache, their asymptotic errors need not be independent. The correction can therefore reduce the leading finite-cache uncertainty by exploiting this shared error.

We next characterize the optimal scalar correction.

Theorem 2 (Optimal reference correction)**.** *Consider the unregularized finite-sample objective*

$$R_M(\lambda) = \mathbb{E}_{\{x_0^{[m]}\}_{m=1}^M \overset{iid}{\sim} \pi_0} \left[\|(\hat{m}_t^\pi(X_t) - m_t^\pi(X_t)) - \lambda(\hat{m}_t^{(i)}(X_t) - m_t^{(i)}(X_t))\|^2 \right]. \tag{26}$$

Assume $\mathbb{E}_{\{x_0^{[m]}\}_{m=1}^M \overset{iid}{\sim} \pi_0} [\|\hat{m}_t^{(i)}(X_t) - m_t^{(i)}(X_t)\|^2] > 0$*. Its minimizer is*

$$\lambda_{t,M}^*(X_t) = \frac{\mathbb{E}_{\{x_0^{[m]}\}_{m=1}^M \overset{iid}{\sim} \pi_0} [(\hat{m}_t^\pi(X_t) - m_t^\pi(X_t)) \cdot (\hat{m}_t^{(i)}(X_t) - m_t^{(i)}(X_t))]}{\mathbb{E}_{\{x_0^{[m]}\}_{m=1}^M \overset{iid}{\sim} \pi_0} [\|\hat{m}_t^{(i)}(X_t) - m_t^{(i)}(X_t)\|^2]}. \tag{27}$$

Under standard moment conditions for self-normalized importance sampling,

$$\lambda_{t,M}^*(X_t) \overset{M\to\infty}{\longrightarrow} \arg\min_\lambda \operatorname{tr} \Sigma_t^\lambda(X_t) \tag{28}$$

Theorem 2 gives a control-variate interpretation of the reference correction: the finite-M coefficient minimizes prediction MSE and asymptotically minimizes the trace of the covariance in Theorem 1. Standard jackknife consistency results further imply that the unregularized estimator satisfies $\hat{\lambda}_{t,M}^{\mathrm{JK}} - \lambda_{t,M}^\star \xrightarrow{p} 0$ under regularity conditions (Parr, 1985). In practice, we regularize $\lambda_t(X_t)$ toward λ_0 for stability with small caches. Proofs are provided in Appendix A.2.

3.5 Compositional Cost

Starting from noise, we use $\hat{m}_t$ in a DDIM-style deterministic update,

$$X_{t-\Delta t} = \frac{\alpha_{t-\Delta t}}{\alpha_t} X_t + \left(\beta_{t-\Delta t} - \frac{\alpha_{t-\Delta t}}{\alpha_t} \beta_t \right) \hat{m}_t(X_t). \tag{29}$$

Unlike the stochastic composition procedure used in Stage 1, Stage 2 therefore admits deterministic sampling and can use a substantially shorter discretization with $T' \ll T$ steps.

Counting diffusion-model function evaluations, generating $M + N$ samples by repeatedly applying the base composition method requires approximately $(M + N)TI$ calls, whereas AmCo requires $MTI + NT'$.

4 Experiment

Evaluation protocol. Following prior work on diffusion composition, we use controlled settings with tractable targets to directly assess distributional correctness, and evaluate realistic high-dimensional applications using task-specific quality and diversity metrics Skreta et al. (2025a); He et al. (2026a;b). Accordingly, our 1D experiment directly measures error against the exact composed target and its clean prediction, while the image and ligand experiments evaluate generation quality and diversity in settings where direct comparison with the exact composed distribution is not tractable.

Sampling assumption. We idealize the Stage-1 anchors as i.i.d. draws from the target distribution π_0, consistent with the theoretical analysis. This assumption is exact in the controlled experiment and is asymptotically justified for particle-based samplers such as FKC. For CREPE, this should be understood as an approximation for post-burn-in samples rather than an exact independence assumption.

4.1 1D COMPOSITION.

We first study AmCo in a one-dimensional setting where the composed target distribution and its clean prediction can be evaluated essentially exactly. This controlled setting allows us to directly quantify the approximation error introduced by the finite Stage-1 cache and to isolate the effect of the finite-sample correction.

We consider two constituent distributions

$$q_0^{(1)} = \mathrm{Unif}[-0.4, 0.4], \qquad q_0^{(2)} = \mathcal{N}(0, 0.3^2), \tag{30}$$

and define a family of product-of-experts compositions

$$F_\beta[q_0^{(1)}(x_0), q_0^{(2)}(x_0)] = \left(q_0^{(1)}(x_0) q_0^{(2)}(x_0)\right)^\beta, \tag{31}$$

with $\beta \in \{0.5, 1.0, 2.0, 3.0\}$. The resulting target distribution is a symmetric truncated Gaussian supported on $[-0.4, 0.4]$, whose concentration increases with β. Varying β therefore provides a simple way to evaluate the finite-cache approximation across target distributions with different levels of concentration.

We compare three finite-sample correction schemes: no correction, with $\lambda_t(X_t) = 0$; fixed correction, with $\lambda_t(X_t) = \lambda_0$; and the adaptive jackknife correction used by AmCo. We use $q^{(1)}$ as the reference constituent for the finite-sample correction.

For each target strength β and each independent run, we draw M Stage-1 anchors exactly from the corresponding composed target distribution $\pi_0^{(\beta)}$ and generate $N = 500$ additional samples using Stage 2. We evaluate $M \in \{4, 8, 16, 32\}$ under all three correction schemes and repeat each configuration over 100 independently sampled Stage-1 caches. We compare the resulting Stage-2 empirical distributions with the exact target distributions and report the mean and variation across independent runs.

Because the exact target clean prediction m_t^π is also available in this setting, we directly evaluate the accuracy of the amortized prediction through

$$\mathrm{MSE}(t) = \frac{1}{N} \sum_{n=1}^{N} \left\| \hat{m}_t(X_t^{[n]}) - m_t^\pi(X_t^{[n]}) \right\|^2. \tag{32}$$

We report this error across diffusion time t to assess how the finite-cache error evolves along the sampling trajectory and how effectively each correction scheme mitigates it.

Figure 2 summarizes the results. Figures 2a and 2b report the clean-prediction error for $M = 4$ at $\beta = 0.5$ and $\beta = 2.0$, respectively. These settings represent relatively broad and concentrated target distributions and illustrate the behavior of the finite-cache estimator under different target geometries. The reference correction reduces the prediction error, particularly in the low-noise regime where the finite set of anchors provides a poorer local approximation to the target clean prediction.

Figure 2c compares the endpoint distributions generated using the adaptive correction with the corresponding exact targets at $M = 32$. Figure 2d reports the Wasserstein error as the cache size increases for all three correction modes. The approximation consistently improves with M, while the reference correction substantially reduces the error associated with a small Stage-1 cache.

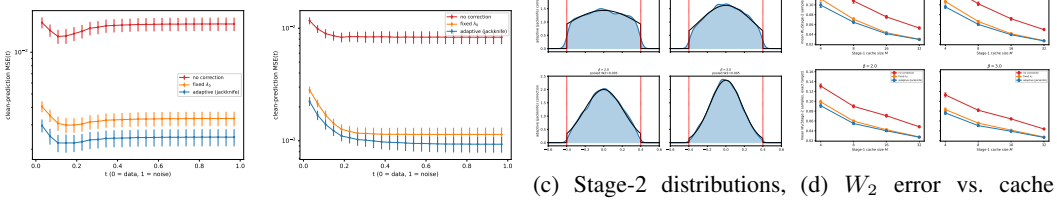
Overall, this experiment addresses two questions: (i) how the finite-cache approximation improves as the number of Stage-1 anchors increases, and (ii) whether the reference-model correction mitigates the resulting finite-sample error. Full estimator definitions, closed-form target quantities, and experimental details are provided in Appendix B.1.

4.2 IMAGE GENERATION.

We evaluate AmCo on classifier-free guidance (CFG), using CREPE He et al. (2026b) as the base composition method:

$$F_w[q_0(x_0|c), q_0(x_0|\emptyset)] = q_0(x_0|c)^w q_0(x_0|\emptyset)^{1-w}. \tag{33}$$

We consider guidance scales $w \in \{1.5, 1.75, 2.0, 2.25, 2.5\}$, Stage-1 sample sizes $M \in \{4, 8, 16\}$, and evaluate $K = 128$ samples per target composition. **AmCo** performs Stage 1 separately for each



(a) Clean-prediction error, $M = 4$, $\beta = 0.5$

(b) Clean-prediction error, $M = 4$, $\beta = 2.0$

(c) Stage-2 distributions, $M = 32$ (pooled over 100 seeds)

(d) W_2 error vs. cache size, by correction mode and target strength

Figure 2: Results of 1D composition.

w and generates $N = K - M$ additional samples with Stage 2. We use $T = 2048$ steps for CREPE and $T' = 64$ steps for Stage 2. We report generation quality together with computational cost in Table 1. FLOPs/sample denotes the marginal cost of obtaining one additional sample, while Total FLOPs denotes the total computation required for the full 5-scale, 15-seed evaluation, including both Stage-1 and Stage-2 costs for AmCo. (CREPE incurs an additional cost C due to chain initialization and the generation of burn-in samples; the same Stage-1 CREPE procedure is also used by AmCo. Thus, the total cost is $C + (M + N)TI$ for CREPE and $C + MTI + NT'$ for AmCo. We report both Total FLOPs that exclude C and include C.) Full experimental details are provided in Appendix B.2.

4.3 Ligand Generation.

We further evaluate AmCo on product-of-experts (PoE) ligand generation, using the Feynman–Kac Corrector (FKC) (Skreta et al., 2025a) as the base composition method:

$$F_\beta[q_0^{(1)}(x_0), q_0^{(2)}(x_0)] = q_0^{(1)}(x_0)^\beta q_0^{(2)}(x_0)^\beta, \tag{34}$$

where $q_0^{(1)}$ and $q_0^{(2)}$ correspond to ligand distributions associated with the two protein pockets. We consider composition strengths $\beta \in \{1.0, 1.5, 2.0\}$ across seven protein pairs. For each target composition, K denotes the total number of evaluated samples, consisting of M Stage-1 samples and $N = K - M$ additional Stage-2 samples. In the main text, we fix $M = 4$ and $K = 32$, so that $N = 28$. **AmCo** performs Stage 1 separately for each β and generates $N = K - M$ additional samples with Stage 2. We use $T = 1000$ steps for the base composition method and $T' = 64$ steps for Stage 2. For each protein pair, we select the sample with the largest $P_1 \times P_2$ among the K generated molecules and report its docking scores against the two target pockets, together with molecular diversity. We report generation quality together with computational cost in Table 2. FLOPs/sample denotes the marginal cost of obtaining one additional sample, while Total FLOPs denotes the total computation required for the full three-β, seven-protein-pair evaluation, including both Stage-1 and Stage-2 costs for AmCo. Full results across sample budgets are provided in Appendix B.3.

5 Related Works

Composition of diffusion models. A growing line of work studies inference-time composition and control of pretrained diffusion models. Existing approaches include energy-based and MCMC-based composition Du et al. (2023), density-based composition Skreta et al. (2025b), Feynman–Kac and sequential Monte Carlo methods Skreta et al. (2025a); Thornton et al. (2025), replica exchange He et al. (2026b), and corrected diffusion dynamics He et al. (2026a); Ren et al. (2026); Lee et al. (2026). Other works develop general formulations for logical composition Blohm & Garg (2026) and product-of-experts generation Zhang et al. (2026). While these methods improve the accuracy and flexibility of diffusion composition, they typically repeat the composition procedure for every generated sample. In contrast, AmCo amortizes this cost by reusing information obtained from a small number of composed samples to efficiently generate additional samples.

Monte Carlo estimation of gradients. Monte Carlo estimators have also been used to approximate otherwise intractable quantities in generative modeling and optimization. Wu et al. (2025),

Table 1: Image generation results and computational cost with $M = 4$ ($K = 128$, 15 seeds, $T = 2048$, $T' = 64$).

Image generation quality				
w	Method	CLIP (↑)	IR (↑)	Diversity (↑)
1.5	CREPE	66.12	-0.058	0.0467
	AmCo	**67.31**	**-0.049**	**0.0538**
1.75	CREPE	67.00	0.116	0.0395
	AmCo	**68.19**	**0.138**	**0.0582**
2.0	CREPE	66.31	0.125	0.0313
	AmCo	**67.62**	**0.143**	**0.0457**
2.25	CREPE	67.06	0.196	0.0264
	AmCo	**68.06**	**0.260**	**0.0396**
2.5	CREPE	66.75	0.225	0.0280
	AmCo	**67.75**	**0.247**	**0.0301**

Computational cost			
Method	FLOPs/sample	Total FLOPs	Speedup
CREPE (exclude C)	4.174×10^{14}	4.007×10^{18}	–
AmCo (exclude C)	6.522×10^{12}	1.859×10^{17}	21.6×
CREPE (include C)	4.174×10^{14}	5.089×10^{18}	–
AmCo (include C)	6.522×10^{12}	1.268×10^{18}	4.0×

Table 2: Ligand generation results and computational cost with $M = 4$, $K = 32$ across seven protein pairs ($T = 1000$, $T' = 64$).

Ligand generation quality				
β	Method	P_1 (↓)	P_2 (↓)	Diversity (↑)
1.0	FKC	-9.24	-9.05	0.18
	AmCo	**-9.83**	**-9.55**	**0.61**
1.5	FKC	-8.79	-8.79	0.18
	AmCo	**-10.00**	**-9.57**	**0.55**
2.0	FKC	-8.82	-8.99	0.32
	AmCo	**-10.30**	**-10.01**	**0.57**

Computational cost			
Method	FLOPs/sample	Total FLOPs	Speedup
FKC	1.828×10^{14}	1.685×10^{16}	–
AmCo	5.466×10^{12}	2.621×10^{15}	6.4×

Niedoba et al. (2024), and Scarvelis et al. (2025) use Monte Carlo samples to estimate diffusion scores, while Kim et al. (2026) and Wierstra et al. (2014) estimate gradients of target objectives using importance-weighted samples, closely related to the self-normalized importance-sampling estimator used in our method. Kahouli et al. (2026) further uses a reference model as a control variate to reduce finite-sample error in score estimation, which is related to our reference-model correction. However, these Monte Carlo estimators are not designed to amortize the cost of composing multiple pretrained diffusion models.

6 Conclusion

We introduced AmCo, a framework for amortizing the cost of composing pretrained diffusion models over multiple generated samples. AmCo uses a small set of samples from an existing composition method to construct an importance-weighted approximation of the target clean prediction, together with a reference-model correction that mitigates finite-sample error. This allows subsequent samples to be generated using only one constituent-model evaluation per step and a substantially shorter deterministic sampling trajectory. Experiments on classifier-free-guided image generation and product-of-experts ligand generation demonstrate that AmCo substantially reduces inference cost while maintaining competitive generation quality and diversity. We hope that amortizing repeated compositional inference provides a useful direction for making flexible inference-time composition practical at scale.

AI USE STATEMENT

In this work, we used generative AI tools to assist with manuscript writing, including language editing, rewriting, and improving the organization and presentation of the paper. We also used generative AI tools throughout the writing process to discuss and refine the structure of the manuscript, clarify technical explanations, and brainstorm how to present the methodology and experimental results.

We additionally used generative AI tools to assist with code development, including generating code snippets, debugging, and improving implementation efficiency. All AI-assisted code was reviewed by the authors and tested for correctness before being used in the experiments.

All AI-assisted text was reviewed, edited, and verified by the authors. Generative AI tools were not used as a substitute for experimental validation or scientific judgment. The authors take full responsibility for the technical content, experimental results, claims, code, and final presentation of the paper.

ETHICS STATEMENT

This work proposes a general-purpose method for reducing the computational cost of inference-time composition of pretrained diffusion models. The study does not involve human subjects, personally identifiable information, or the collection of new sensitive data.

We evaluate the method on image generation and computational ligand generation. Although molecular generative models may have potential dual-use implications, our ligand experiments are used only as a computational benchmark; we perform no experimental synthesis or biological validation. The proposed method may inherit the risks, biases, and limitations of the pretrained models and application domains to which it is applied. Users should therefore consider the relevant ethical, legal, and safety implications when deploying the method.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the proposed method, algorithms, experimental protocols, and hyperparameters throughout the main paper and appendix. Appendix A contains the derivation of the adaptive correction and the practical AmCo algorithm, while Appendix B provides the full experimental settings and additional results for the 1D, image-generation, and ligand-generation experiments. We will additionally release an anonymous GitHub repository with the implementation used in this work, including code for the proposed method, experimental configurations, and evaluation pipelines needed to reproduce the reported results.

REFERENCES

Peter Blohm and Vikas Garg. Composition of pretrained diffusion models: A logic-based calculus. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 136449–136478, 2026.

Yilun Du, Conor Durkan, Robin Strudel, Joshua B. Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Sussman Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and MCMC. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 8489–8510. PMLR, 23–29 Jul 2023.

Bradley Efron. *The jackknife, the bootstrap, and other resampling plans*, volume 38. Siam, 1982.

Jiajun He, José Miguel Hernández Lobato, Yuanqi Du, and Francisco Vargas. Rne: plug-and-play diffusion inference-time control and energy-based training. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 75732–75786, 2026a.

Jiajun He, Paul Jeha, Peter Potaptchik, Leo Zhang, José Miguel Hernández Lobato, Yuanqi Du, Saifuddin Syed, and Francisco Vargas. Crepe: Controlling diffusion with replica exchange. In

C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 99464–99494, 2026b.

Khaled Kahouli, Romuald Elie, Klaus-Robert Müller, Quentin Berthet, Oliver T. Unke, and Arnaud Doucet. Control variate score matching for diffusion models, 2026.

Yeongmin Kim, Donghyeok Shin, Byeonghu Na, Minsang Park, Richard Lee Kim, and Il chul Moon. Lookahead sample reward guidance for test-time scaling of diffusion models. In *Forty-third International Conference on Machine Learning*, 2026.

Ziseok Lee, Minyeong Hwang, Wooyeol Lee, Sanghyun Jo, Jihyung Ko, Young Bin Park, Jae-Mun Choi, Eunho Yang, and Kyungsu Kim. On the collapse of generative paths: A criterion and correction for diffusion steering. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Matthew Niedoba, Dylan Green, Saeid Naderiparizi, Vasileios Lioutas, Jonathan Wilder Lavington, Xiaoxuan Liang, Yunpeng Liu, Ke Zhang, Setareh Dabiri, Adam Scibior, Berend Zwartsenberg, and Frank Wood. Nearest neighbour score estimators for diffusion generative models. In *Forty-first International Conference on Machine Learning*, 2024.

William C. Parr. Jackknifing differentiable statistical functionals. *Journal of the Royal Statistical Society. Series B (Methodological)*, 47(1):56–66, 1985. ISSN 00359246.

Yinuo Ren, Wenhao Gao, Lexing Ying, Grant Rotskoff, and Jiequn Han. Driftlite: Lightweight drift control for inference-time scaling of diffusion models. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 76664–76702, 2026.

Christopher Scarvelis, Haitz Sáez de Ocáriz Borde, and Justin Solomon. Closed-form diffusion models. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856.

Marta Skreta, Tara Akhound-Sadegh, Viktor Ohanesian, Roberto Bondesan, Alan Aspuru-Guzik, Arnaud Doucet, Rob Brekelmans, Alexander Tong, and Kirill Neklyudov. Feynman-kac correctors in diffusion: Annealing, guidance, and product of experts. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 55906–55949. PMLR, 13–19 Jul 2025a.

Marta Skreta, Lazar Atanackovic, Joey Bose, Alexander Tong, and Kirill Neklyudov. The superposition of diffusion models using the itô density estimator. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 67004–67057, 2025b.

James Thornton, Louis Béthune, Ruixiang ZHANG, Arwen Bradley, Preetum Nakkiran, and Shuangfei Zhai. Composition and control with distilled energy diffusion models and sequential monte carlo. In Yingzhen Li, Stephan Mandt, Shipra Agrawal, and Emtiyaz Khan (eds.), *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pp. 3259–3267. PMLR, 03–05 May 2025.

Siddarth Venkatraman, Moksh Jain, Luca Scimeca, Minsu Kim, Marcin Sendera, Mohsin Hasan, Luke Rowe, Sarthak Mittal, Pablo Lemos, Emmanuel Bengio, Alexandre Adam, Jarrid Rector-Brooks, Yoshua Bengio, Glen Berseth, and Nikolay Malkin. Amortizing intractable inference in diffusion models for vision, language, and control. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, Jan Peters, and Jürgen Schmidhuber. Natural evolution strategies. *The Journal of Machine Learning Research*, 15(1):949–980, 2014.

Luhuan Wu, Brian Trippe, Christian Naesseth, David Blei, and John Cunningham. Practical and asymptotically exact conditional sampling in diffusion models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 31372–31403. Curran Associates, Inc., 2023. doi: 10.52202/075280-1363.

Luhuan Wu, Yi Han, Christian Andersson Naesseth, and John Cunningham. Reverse diffusion sequential monte carlo samplers. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 69445–69459. Curran Associates, Inc., 2025. doi: 10.52202/085713-2336.

Yunzhi Zhang, Carson Murtuza-Lanier, Zizhang Li, Yilun Du, and Jiajun Wu. Product of experts for visual generation. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 111416–111443, 2026.

A METHODOLOGY DESCRIPTION.

A.1 ESTIMATION OF $\lambda_t(X_t)$

We choose $\lambda_t(X_t)$ to minimize the clean-prediction error while regularizing it. For a fixed X_t, all expectations below are taken over the randomness of the finite Stage-1 sample set. The objective is

$$\mathbb{E}\big[\|m_t^\pi(X_t) - \hat{m}_t(X_t)\|^2\big] + \eta(\lambda_t(X_t) - \lambda_0)^2 \tag{35}$$

$$= \mathbb{E}\left[\left\|\big(m_t^\pi(X_t) - \hat{m}_t^\pi(X_t)\big) - \lambda_t(X_t)\big(m_t^{(i)}(X_t) - \hat{m}_t^{(i)}(X_t)\big)\right\|^2\right] + \eta(\lambda_t(X_t) - \lambda_0)^2. \tag{36}$$

For fixed X_t, the minimizer is

$$\lambda_t^*(X_t) = \frac{\mathbb{E}\Big[\big(m_t^\pi(X_t) - \hat{m}_t^\pi(X_t)\big) \cdot \big(m_t^{(i)}(X_t) - \hat{m}_t^{(i)}(X_t)\big)\Big] + \eta\lambda_0}{\mathbb{E}\Big[\|m_t^{(i)}(X_t) - \hat{m}_t^{(i)}(X_t)\|^2\Big] + \eta} \tag{37}$$

$$= \frac{\operatorname{tr}\operatorname{Cov}\Big(\hat{m}_t^\pi(X_t), \hat{m}_t^{(i)}(X_t)\Big) + \mathbb{E}[m_t^\pi(X_t) - \hat{m}_t^\pi(X_t)] \cdot \mathbb{E}\Big[m_t^{(i)}(X_t) - \hat{m}_t^{(i)}(X_t)\Big] + \eta\lambda_0}{\operatorname{tr}\operatorname{Var}\Big(\hat{m}_t^{(i)}(X_t)\Big) + \Big\|\mathbb{E}\Big[m_t^{(i)}(X_t) - \hat{m}_t^{(i)}(X_t)\Big]\Big\|^2 + \eta}. \tag{38}$$

Since $m_t^\pi(X_t)$ is unavailable, we estimate the bias and covariance terms using a leave-one-out jackknife approximation Efron (1982). Let

$$\hat{m}_t^\pi[-m] = \frac{\sum_{k\neq m} w_t^\pi(x_0^{[k]}, X_t) x_0^{[k]}}{\sum_{k\neq m} w_t^\pi(x_0^{[k]}, X_t)}, \qquad \bar{m}_t^\pi[-] = \frac{1}{M}\sum_{m=1}^M \hat{m}_t^\pi[-m], \tag{39}$$

$$\hat{m}_t^{(i)}[-m] = \frac{\sum_{k\neq m} w_t^{(i)}(x_0^{[k]}, X_t) x_0^{[k]}}{\sum_{k\neq m} w_t^{(i)}(x_0^{[k]}, X_t)}, \qquad \bar{m}_t^{(i)}[-] = \frac{1}{M}\sum_{m=1}^M \hat{m}_t^{(i)}[-m]. \tag{40}$$

The resulting jackknife estimate is

$$\begin{aligned}\hat{\lambda}_t^{\mathrm{JK}}(X_t) = \Bigg\{ &(M-1)^2\big(\bar{m}_t^\pi[-] - \hat{m}_t^\pi(X_t)\big) \cdot \big(\bar{m}_t^{(i)}[-] - \hat{m}_t^{(i)}(X_t)\big) \\ &+ \frac{M-1}{M}\sum_{m=1}^M \big(\hat{m}_t^\pi[-m] - \bar{m}_t^\pi[-]\big) \cdot \big(\hat{m}_t^{(i)}[-m] - \bar{m}_t^{(i)}[-]\big) + \eta\lambda_0 \Bigg\} \\ \times \Bigg\{ &(M-1)^2 \left\|\bar{m}_t^{(i)}[-] - \hat{m}_t^{(i)}(X_t)\right\|^2 \\ &+ \frac{M-1}{M}\sum_{m=1}^M \left\|\hat{m}_t^{(i)}[-m] - \bar{m}_t^{(i)}[-]\right\|^2 + \eta \Bigg\}^{-1}.\end{aligned} \tag{41}$$

A.2 FINITE-CACHE ERROR ANALYSIS

We provide the finite-cache analysis underlying Theorems 1 and 2. Throughout this section, we fix $t \in (0,1)$ and X_t, and all expectations, variances, and convergence statements are with respect to the randomness of the Stage-1 cache. For the theoretical analysis, we assume i.i.d. Stage-1 anchors

$$x_0^{[1]}, \ldots, x_0^{[M]} \overset{\text{iid}}{\sim} \pi_0. \tag{42}$$

We suppress (t, X_t) from the notation whenever no ambiguity arises.

Define

$$w_\pi(x_0) = w_t^\pi(x_0, X_t), \qquad w_i(x_0) = w_t^{(i)}(x_0, X_t), \tag{43}$$

and their normalizing constants

$$\mu_\pi = \mathbb{E}_{x_0\sim\pi_0}[w_\pi(x_0)], \qquad \mu_i = \mathbb{E}_{x_0\sim\pi_0}[w_i(x_0)]. \tag{44}$$

Then

$$m_t^\pi(X_t) = \frac{\mathbb{E}_{x_0\sim\pi_0}[w_\pi(x_0)x_0]}{\mu_\pi}, \tag{45}$$

$$m_t^{(i)}(X_t) = \frac{\mathbb{E}_{x_0\sim\pi_0}[w_i(x_0)x_0]}{\mu_i}. \tag{46}$$

The corresponding finite-cache estimators are

$$\hat{m}_t^\pi(X_t) = \frac{\sum_{m=1}^M w_\pi(x_0^{[m]})x_0^{[m]}}{\sum_{m=1}^M w_\pi(x_0^{[m]})}, \tag{47}$$

$$\hat{m}_t^{(i)}(X_t) = \frac{\sum_{m=1}^M w_i(x_0^{[m]})x_0^{[m]}}{\sum_{m=1}^M w_i(x_0^{[m]})}. \tag{48}$$

Regularity conditions. We impose the following standard conditions for smooth self-normalized importance-sampling estimators.

Assumption 3. For fixed $t \in (0,1)$ and X_t:

1. The weight normalizers are strictly positive:

$$\mu_\pi > 0, \qquad \mu_i > 0. \tag{49}$$

2. There exists $\delta > 0$ such that

$$\mathbb{E}_{x_0\sim\pi_0}\left[\left\|\begin{pmatrix} w_\pi(x_0)x_0 \\ w_\pi(x_0) \\ w_i(x_0)x_0 \\ w_i(x_0)\end{pmatrix}\right\|^{4+\delta}\right] < \infty. \tag{50}$$

3. The asymptotic variance of the reference estimator is nonzero:

$$\operatorname{tr}\operatorname{Var}_{x_0\sim\pi_0}[\psi_i(x_0)] > 0. \tag{51}$$

The moment condition above is stronger than required for asymptotic normality alone, but is convenient for controlling the second-order terms used below and for applying standard jackknife consistency results. Since $\mu_\pi, \mu_i > 0$, both estimators are smooth ratio functionals in a neighborhood of their population values.

Define the corresponding influence functions

$$\psi_\pi(x_0) = \frac{w_\pi(x_0)}{\mu_\pi}\big(x_0 - m_t^\pi(X_t)\big), \tag{52}$$

$$\psi_i(x_0) = \frac{w_i(x_0)}{\mu_i}\big(x_0 - m_t^{(i)}(X_t)\big). \tag{53}$$

By construction,

$$\mathbb{E}_{x_0\sim\pi_0}[\psi_\pi(x_0)] = \mathbb{E}_{x_0\sim\pi_0}[\psi_i(x_0)] = 0. \tag{54}$$

Lemma 4 (Joint asymptotic linearity)**.** *Under Assumption 3,*

$$\hat{m}_t^\pi(X_t) - m_t^\pi(X_t) = \frac{1}{M}\sum_{m=1}^M \psi_\pi(x_0^{[m]}) + o_p(M^{-1/2}), \tag{55}$$

$$\hat{m}_t^{(i)}(X_t) - m_t^{(i)}(X_t) = \frac{1}{M}\sum_{m=1}^M \psi_i(x_0^{[m]}) + o_p(M^{-1/2}). \tag{56}$$

More precisely, defining the remainders

$$r_{\pi,M} := \hat{m}_t^{\pi}(X_t) - m_t^{\pi}(X_t) - \frac{1}{M}\sum_{m=1}^{M} \psi_{\pi}(x_0^{[m]}), \tag{57}$$

$$r_{i,M} := \hat{m}_t^{(i)}(X_t) - m_t^{(i)}(X_t) - \frac{1}{M}\sum_{m=1}^{M} \psi_{i}(x_0^{[m]}), \tag{58}$$

the remainders are negligible in second moment:

$$\mathbb{E}\|r_{\pi,M}\|^2 = o(M^{-1}), \qquad \mathbb{E}\|r_{i,M}\|^2 = o(M^{-1}). \tag{59}$$

Consequently,

$$\sqrt{M}\begin{pmatrix} \hat{m}_t^{\pi}(X_t) - m_t^{\pi}(X_t) \\ \hat{m}_t^{(i)}(X_t) - m_t^{(i)}(X_t) \end{pmatrix} \Rightarrow \mathcal{N}(0, \Omega_t(X_t)), \tag{60}$$

where

$$\Omega_t(X_t) = \mathrm{Var}_{x_0\sim\pi_0}\begin{pmatrix} \psi_{\pi}(x_0) \\ \psi_i(x_0) \end{pmatrix}. \tag{61}$$

Proof. We prove the result for $\hat{m}_t^{\pi}(X_t)$; the argument for $\hat{m}_t^{(i)}(X_t)$ is identical. Define

$$A_{\pi,M} = \frac{1}{M}\sum_{m=1}^{M} w_{\pi}(x_0^{[m]})x_0^{[m]}, \qquad B_{\pi,M} = \frac{1}{M}\sum_{m=1}^{M} w_{\pi}(x_0^{[m]}). \tag{62}$$

Then

$$\hat{m}_t^{\pi}(X_t) = g(A_{\pi,M}, B_{\pi,M}), \qquad g(a,b) = \frac{a}{b}. \tag{63}$$

Since $\mu_{\pi} > 0$, g is continuously differentiable in a neighborhood of the population value. A first-order Taylor expansion gives

$$\hat{m}_t^{\pi}(X_t) - m_t^{\pi}(X_t) = \frac{1}{\mu_{\pi}}\left[A_{\pi,M} - \mathbb{E}[A_{\pi,M}] - m_t^{\pi}(X_t)(B_{\pi,M} - \mu_{\pi})\right] + o_p(M^{-1/2}) \tag{64}$$

$$= \frac{1}{M}\sum_{m=1}^{M} \frac{w_{\pi}(x_0^{[m]})}{\mu_{\pi}}\left(x_0^{[m]} - m_t^{\pi}(X_t)\right) + o_p(M^{-1/2}) \tag{65}$$

$$= \frac{1}{M}\sum_{m=1}^{M} \psi_{\pi}(x_0^{[m]}) + o_p(M^{-1/2}). \tag{66}$$

It remains to verify the second-moment control of the Taylor remainders. On a neighborhood of the population values with denominators bounded away from zero, the ratio map $g(a,b) = a/b$ has bounded second derivatives. Hence, the second-order Taylor remainder is bounded by a constant times the squared deviation of the corresponding empirical averages from their population values. Under Assumption 3, the $(4+\delta)$-moment condition implies that the fourth moments of these empirical-average deviations are $O(M^{-2})$. Together with $\mu_{\pi}, \mu_i > 0$, this yields

$$\mathbb{E}\|r_{\pi,M}\|^2 = o(M^{-1}), \qquad \mathbb{E}\|r_{i,M}\|^2 = o(M^{-1}). \tag{67}$$

Applying the multivariate central limit theorem jointly to the empirical averages of the target and reference statistics, followed by the multivariate delta method, yields the stated joint Gaussian limit. □

Proof of Theorem 1. Define

$$e_{\pi,M} = \hat{m}_t^{\pi}(X_t) - m_t^{\pi}(X_t), \qquad e_{i,M} = \hat{m}_t^{(i)}(X_t) - m_t^{(i)}(X_t). \tag{68}$$

The corrected predictor satisfies

$$\hat{m}_t(X_t) - m_t^{\pi}(X_t) = e_{\pi,M} - \hat{\lambda}_{t,M}(X_t)e_{i,M}. \tag{69}$$

By Lemma 4,

$$e_{\pi,M} = O_p(M^{-1/2}), \qquad e_{i,M} = O_p(M^{-1/2}). \tag{70}$$

Thus, if

$$\hat{\lambda}_{t,M}(X_t) \xrightarrow{p} \lambda_t(X_t), \tag{71}$$

then

$$\hat{m}_t(X_t) \xrightarrow{p} m_t^\pi(X_t). \tag{72}$$

For the asymptotic distribution,

$$\sqrt{M}\big(\hat{m}_t(X_t) - m_t^\pi(X_t)\big) \tag{73}$$

$$= \sqrt{M}\big(e_{\pi,M} - \lambda_t(X_t)e_{i,M}\big) - \big(\hat{\lambda}_{t,M}(X_t) - \lambda_t(X_t)\big)\sqrt{M}e_{i,M}. \tag{74}$$

The second term is $o_p(1)$ since $\hat{\lambda}_{t,M}(X_t) - \lambda_t(X_t) = o_p(1)$ and $\sqrt{M}e_{i,M} = O_p(1)$. Hence, using Lemma 4,

$$\sqrt{M}\big(\hat{m}_t(X_t) - m_t^\pi(X_t)\big) = \frac{1}{\sqrt{M}} \sum_{m=1}^{M} \Big[\psi_\pi(x_0^{[m]}) - \lambda_t(X_t)\psi_i(x_0^{[m]})\Big] + o_p(1). \tag{75}$$

The multivariate central limit theorem therefore gives

$$\sqrt{M}\big(\hat{m}_t(X_t) - m_t^\pi(X_t)\big) \Rightarrow \mathcal{N}\big(0, \Sigma_t^\lambda(X_t)\big), \tag{76}$$

where

$$\Sigma_t^\lambda(X_t) = \mathrm{Var}_{x_0\sim\pi_0}\left[\psi_\pi(x_0) - \lambda_t(X_t)\psi_i(x_0)\right]. \tag{77}$$

This proves Theorem 1.

Proof of Theorem 2. For fixed M, the unregularized objective is

$$R_M(\lambda) = \mathbb{E}\left[\|e_{\pi,M} - \lambda e_{i,M}\|^2\right]. \tag{78}$$

Expanding the square,

$$R_M(\lambda) = \mathbb{E}\|e_{\pi,M}\|^2 - 2\lambda\mathbb{E}[e_{\pi,M}^\top e_{i,M}] + \lambda^2\mathbb{E}\|e_{i,M}\|^2. \tag{79}$$

Since $\mathbb{E}\|e_{i,M}\|^2 > 0$, the objective is strictly convex in λ, and its unique minimizer is

$$\lambda_M^\star = \frac{\mathbb{E}[e_{\pi,M}^\top e_{i,M}]}{\mathbb{E}\|e_{i,M}\|^2}. \tag{80}$$

To characterize its asymptotic limit, define

$$c_{\pi i} = \mathbb{E}_{x_0\sim\pi_0}\left[\psi_\pi(x_0)^\top\psi_i(x_0)\right], \tag{81}$$

$$v_i = \mathbb{E}_{x_0\sim\pi_0}\left[\|\psi_i(x_0)\|^2\right]. \tag{82}$$

By the second-moment remainder bounds in Lemma 4 and the Cauchy–Schwarz inequality,

$$M\mathbb{E}[e_{\pi,M}^\top e_{i,M}] \longrightarrow c_{\pi i}, \tag{83}$$

$$M\mathbb{E}\|e_{i,M}\|^2 \longrightarrow v_i. \tag{84}$$

For the cross term, the leading contribution is

$$\mathbb{E}\left[\left(\frac{1}{M}\sum_{m=1}^{M}\psi_\pi(x_0^{[m]})\right)^\top\left(\frac{1}{M}\sum_{n=1}^{M}\psi_i(x_0^{[n]})\right)\right] \tag{85}$$

$$= \frac{1}{M^2}\sum_{m,n=1}^{M}\mathbb{E}\left[\psi_\pi(x_0^{[m]})^\top\psi_i(x_0^{[n]})\right]. \tag{86}$$

For $m \neq n$, independence and $\mathbb{E}[\psi_\pi] = \mathbb{E}[\psi_i] = 0$ imply that the corresponding terms vanish. Therefore only the M diagonal terms remain, yielding

$$\mathbb{E}[e_{\pi,M}^\top e_{i,M}] = \frac{c_{\pi i}}{M} + o(M^{-1}). \tag{87}$$

Similarly,

$$\mathbb{E}\|e_{i,M}\|^2 = \frac{v_i}{M} + o(M^{-1}). \tag{88}$$

Since $v_i > 0$,

$$\lambda_M^\star \longrightarrow \frac{c_{\pi i}}{v_i}. \tag{89}$$

Finally, define

$$v_\pi = \mathbb{E}_{x_0 \sim \pi_0}\left[\|\psi_\pi(x_0)\|^2\right]. \tag{90}$$

The asymptotic covariance in Theorem 1 satisfies

$$\operatorname{tr} \Sigma_t^\lambda(X_t) = \operatorname{tr} \operatorname{Var}_{x_0 \sim \pi_0}\left[\psi_\pi(x_0) - \lambda \psi_i(x_0)\right] \tag{91}$$

$$= v_\pi - 2\lambda c_{\pi i} + \lambda^2 v_i. \tag{92}$$

This quadratic is uniquely minimized at

$$\lambda = \frac{c_{\pi i}}{v_i}. \tag{93}$$

Therefore,

$$\lambda_M^\star \longrightarrow \arg\min_\lambda \operatorname{tr} \Sigma_t^\lambda(X_t), \tag{94}$$

which proves Theorem 2.

Consistency of the jackknife coefficient. Let $\hat{m}_{t,[-m]}^\pi(X_t)$ and $\hat{m}_{t,[-m]}^{(i)}(X_t)$ denote the leave-one-out estimators obtained after removing $x_0^{[m]}$, and define

$$\bar{m}_{t,[-]}^\pi(X_t) = \frac{1}{M}\sum_{m=1}^M \hat{m}_{t,[-m]}^\pi(X_t), \tag{95}$$

$$\bar{m}_{t,[-]}^{(i)}(X_t) = \frac{1}{M}\sum_{m=1}^M \hat{m}_{t,[-m]}^{(i)}(X_t). \tag{96}$$

The delete-one jackknife estimates of the cross-covariance and reference variance are

$$\widehat{C}_{\pi i}^{\mathrm{JK}} = \frac{M-1}{M}\sum_{m=1}^M \left(\hat{m}_{t,[-m]}^\pi - \bar{m}_{t,[-]}^\pi\right)^\top \left(\hat{m}_{t,[-m]}^{(i)} - \bar{m}_{t,[-]}^{(i)}\right), \tag{97}$$

$$\widehat{V}_i^{\mathrm{JK}} = \frac{M-1}{M}\sum_{m=1}^M \left\|\hat{m}_{t,[-m]}^{(i)} - \bar{m}_{t,[-]}^{(i)}\right\|^2. \tag{98}$$

The corresponding jackknife bias estimates are

$$\widehat{b}_\pi^{\mathrm{JK}} = (M-1)\left(\bar{m}_{t,[-]}^\pi - \hat{m}_t^\pi\right), \tag{99}$$

$$\widehat{b}_i^{\mathrm{JK}} = (M-1)\left(\bar{m}_{t,[-]}^{(i)} - \hat{m}_t^{(i)}\right). \tag{100}$$

The unregularized jackknife coefficient is therefore

$$\hat{\lambda}_{t,M}^{\mathrm{JK}} = \frac{\widehat{C}_{\pi i}^{\mathrm{JK}} + (\widehat{b}_\pi^{\mathrm{JK}})^\top \widehat{b}_i^{\mathrm{JK}}}{\widehat{V}_i^{\mathrm{JK}} + \|\widehat{b}_i^{\mathrm{JK}}\|^2}. \tag{101}$$

Both $\hat{m}_t^\pi$ and $\hat{m}_t^{(i)}$ are smooth ratio functionals of empirical averages. Standard delete-one jackknife consistency results for smooth statistical functionals (Efron, 1982; Parr, 1985) therefore imply

$$M\widehat{C}_{\pi i}^{\mathrm{JK}} \xrightarrow{p} c_{\pi i}, \tag{102}$$

$$M\widehat{V}_i^{\mathrm{JK}} \xrightarrow{p} v_i, \tag{103}$$

while

$$\widehat{b}_{\pi}^{\mathrm{JK}} = O_p(M^{-1}), \qquad \widehat{b}_{i}^{\mathrm{JK}} = O_p(M^{-1}). \tag{104}$$

Consequently,

$$\hat{\lambda}_{t,M}^{\mathrm{JK}} \xrightarrow{p} \frac{c_{\pi i}}{v_i}. \tag{105}$$

Since Theorem 2 also gives

$$\lambda_M^{\star} \longrightarrow \frac{c_{\pi i}}{v_i}, \tag{106}$$

we obtain

$$\hat{\lambda}_{t,M}^{\mathrm{JK}} - \lambda_M^{\star} \xrightarrow{p} 0. \tag{107}$$

A.3 Practical Algorithm.

In practice, existing exact composition methods encounter numerical instability near the data endpoint, i.e., as $t \to 0$ (Skreta et al., 2025a; Blohm & Garg, 2026; Ren et al., 2026; Lee et al., 2026; He et al., 2026b). A common practical remedy is to disable the composition correction sufficiently close to $t = 0$. Consequently, the exact-composition guarantee applies only away from this numerically unstable regime. Following this convention, we modify Algorithm 1 to obtain the practical implementation in Algorithm 2.

B Experimental Details.

B.1 Details of the 1D Composition Experiment

We use the stochastic interpolant

$$X_t = \alpha_t Z + \beta_t X_0, \qquad \alpha_t = t, \qquad \beta_t = 1 - t, \tag{108}$$

with $Z \sim \mathcal{N}(0, 1)$.

For

$$q_0^{(1)} = \mathrm{Unif}[-0.4, 0.4], \qquad q_0^{(2)} = \mathcal{N}(0, 0.3^2), \tag{109}$$

the composed target

$$\pi_0^{(\beta)}(x) \propto \left(q_0^{(1)}(x) q_0^{(2)}(x)\right)^{\beta} \tag{110}$$

is a symmetric truncated Gaussian,

$$\pi_0^{(\beta)}(x) \propto \mathbf{1}|x| \leq 0.4 \exp\left(-\frac{\beta x^2}{2(0.3)^2}\right). \tag{111}$$

We numerically evaluate its normalization and the corresponding clean prediction

$$m_t^{(\beta)}(x) = \mathbb{E}_{\pi_0^{(\beta)}}[X_0 \mid X_t = x] \tag{112}$$

using one-dimensional quadrature. This provides an effectively exact reference for evaluating the Stage-2 approximation.

For each target strength $\beta \in \{0.5, 1.0, 2.0, 3.0\}$, we independently draw $M \in \{4, 8, 16, 32\}$ Stage-1 anchors exactly from the corresponding composed target distribution $\pi_0^{(\beta)}$ and store the constituent density values at those anchors. We use $q^{(1)}$ as the reference constituent for the finite-sample correction.

Stage 2 uses $T' = 128$ discretization steps with $\epsilon = 0$, $\eta = 0.05$, and $\lambda_0 = 0.5$. We compare no correction, $\lambda_t(X_t) = 0$; fixed correction, $\lambda_t(X_t) = \lambda_0$; and the adaptive jackknife correction described in Appendix A.1.

For every combination of M, target strength, and correction mode, we generate $N = 500$ Stage-2 samples and repeat the experiment over 100 independently sampled Stage-1 caches. We compare the resulting empirical distribution against the exact target distribution and report statistics across the 100 independent runs.

Algorithm 2 AmCo without numerical issue

// Stage 1
Use any base composition method to obtain $x_\epsilon^{[m]} \sim \pi_\epsilon(x_\epsilon)$ and $q_\epsilon^{(1)}(x_\epsilon^{[m]}), q_\epsilon^{(2)}(x_\epsilon^{[m]}), \cdots, q_\epsilon^{(I)}(x_\epsilon^{[m]})$ $(m = 1, 2, \cdots, M)$.
for $m \in \{1, 2, \cdots, M\}$ **do**
    $\tilde{\pi}_\epsilon(x_\epsilon^{[m]}) \leftarrow F[q_\epsilon^{(1)}(x_\epsilon^{[m]}), q_\epsilon^{(2)}(x_\epsilon^{[m]}), \cdots, q_\epsilon^{(I)}(x_\epsilon^{[m]})]$
end for
// Stage 2
for $n \in \{1, 2, \cdots, N\}$ **do**
    $X_{1-\delta}^{[n]} \overset{iid}{\sim} \mathcal{N}(0, \alpha_{1-\delta}^2 I)$ (We should take $\delta > 0$ if $\beta_1 = 0$.)
    for $t \in \{1-\delta, 1-\delta-\Delta t, \cdots, \epsilon + 2\Delta t\}$ **do**
        $\bar{\beta}_t \leftarrow \frac{\beta_t}{\beta_\epsilon}, \bar{\alpha}_t \leftarrow \sqrt{\alpha_t^2 - \frac{\beta_t^2}{\beta_\epsilon^2}\alpha_\epsilon^2}, \bar{\beta}_t^{(i)} \leftarrow \frac{\beta_t^{(i)}}{\beta_\epsilon^{(i)}}, \bar{\alpha}_t^{(i)} \leftarrow \sqrt{(\alpha_t^{(i)})^2 - \frac{(\beta_t^{(i)})^2}{(\beta_\epsilon^{(i)})^2}(\alpha_\epsilon^{(i)})^2}$
        $m_t^{(i)}(X_t^{[n]}) \leftarrow \frac{X_t^{[n]} + (\bar{\alpha}_t^{(i)})^2 s_t^{(i)}(X_t^{[n]})}{\bar{\beta}_t^{(i)}}$
        $\{w_t^\pi(x_\epsilon^{[m]}, X_t^{[n]})\}_{m=1}^M \leftarrow \{\mathcal{N}(X_t^{[n]}; \bar{\beta}_t x_\epsilon^{[m]}, (\bar{\alpha}_t)^2 I)\}_{m=1}^M$
        $\hat{m}_t^\pi(X_t^{[n]}) \leftarrow \frac{\sum_{m=1}^M w_t^\pi(x_\epsilon^{[m]}; X_t^{[n]}) x_\epsilon^{[m]}}{\sum_{m=1}^M w_t^\pi(x_\epsilon^{[m]}; X_t^{[n]})}$
        $\{w_t^{(i)}(x_\epsilon^{[m]}, X_t^{[n]})\}_{m=1}^M \leftarrow \{\frac{q_\epsilon^{(i)}(x_\epsilon^{[m]}) \mathcal{N}(X_t^{[n]}; \bar{\beta}_t^{(i)} x_\epsilon^{[m]}, (\bar{\alpha}_t^{(i)})^2 I)}{\tilde{\pi}_\epsilon(x_\epsilon^{[m]})}\}_{m=1}^M$
        $\hat{m}_t^{(i)}(X_t^{[n]}) \leftarrow \frac{\sum_{m=1}^M w_t^{(i)}(x_\epsilon^{[m]}; X_t^{[n]}) x_\epsilon^{[m]}}{\sum_{m=1}^M w_t^{(i)}(x_\epsilon^{[m]}; X_t^{[n]})}$
        $\{\hat{m}_t^\pi[-m]\}_{m=1}^M \leftarrow \{\frac{\sum_{k \neq m} w_t^\pi(x_\epsilon^{[k]}, X_t^{[n]}) x_\epsilon^{[k]}}{\sum_{k \neq m} w_t^\pi(x_\epsilon^{[k]}, X_t^{[n]})}\}_{m=1}^M, \quad \bar{m}_t^\pi[-] \leftarrow \frac{1}{M}\sum_{m=1}^M \hat{m}_t^\pi[-m]$
        $\{\hat{m}_t^{(i)}[-m]\}_{m=1}^M \leftarrow \{\frac{\sum_{k \neq m} w_t^{(i)}(x_\epsilon^{[k]}, X_t^{[n]}) x_\epsilon^{[k]}}{\sum_{k \neq m} w_t^{(i)}(x_\epsilon^{[k]}, X_t^{[n]})}\}_{m=1}^M, \quad \bar{m}_t^{(i)}[-] \leftarrow \frac{1}{M}\sum_{m=1}^M \hat{m}_t^{(i)}[-m]$
        $\hat{\lambda}_t^{\text{JK}}(X_t^{[n]}) \leftarrow \{(M-1)^2(\bar{m}_t^\pi[-] - \hat{m}_t^\pi(X_t^{[n[}])) \cdot (\bar{m}_t^{(i)}[-] - \hat{m}_t^{(i)}(X_t^{[n[}])) + \frac{M-1}{M}\sum_{m=1}^M(\hat{m}_t^\pi[-m] - \bar{m}_t^\pi[-]) \cdot (\hat{m}_t^{(i)}[-m] - \bar{m}_t^{(i)}[-]) + \eta\lambda_0\} \times \{(M-1)^2\|\bar{m}_t^{(i)}[-] - \hat{m}_t^{(i)}(X_t^{[n[}])\|^2 + \frac{M-1}{M}\sum_{m=1}^M \|\hat{m}_t^{(i)}[-m] - \bar{m}_t^{(i)}[-]\|^2 + \eta\}^{-1}$
        $\hat{m}_t(X_t^{[n]}) \leftarrow \hat{m}_t^\pi(X_t^{[n]}) + \hat{\lambda}_t^{\text{JK}}(X_t^{[n]})[m_t^{(i)}(X_t^{[n]}) - \hat{m}_t^{(i)}(X_t^{[n]})]$
        $X_{t-\Delta t}^{[n]} \leftarrow \frac{\bar{\alpha}_{t-\Delta t}}{\bar{\alpha}_t} X_t^{[n]} + (\bar{\beta}_{t-\Delta t} - \frac{\bar{\alpha}_{t-\Delta t}}{\bar{\alpha}_t}\bar{\beta}_t)\hat{m}_t(X_t^{[n]})$
    end for
    for $t \in \{\epsilon + \Delta t, \epsilon, \cdots, \Delta t\}$ **do**
        $v_t^{(i)}(X_t^{[n]}) \leftarrow \frac{\dot{\beta}_t^{(i)} X_t^{[n]} + \alpha_t^{(i)}[\alpha_t^{(i)}\dot{\beta}_t^{(i)} - \dot{\alpha}_t^{(i)}\beta_t^{(i)}] s_t^{(i)}(X_t^{[n]})}{\beta_t^{(i)}}$
        $X_{t-\Delta t}^{[n]} \leftarrow X_t^{[n]} - v_t^{(i)}(X_t^{[n]})\Delta t$
    end for
end for
return $\{X_0^{[n]}\}_{n=1}^N$

Table 3: Endpoint W_2 distance between Stage-2 samples and the exact target $F = F_\beta$, mean $\pm$ SEM over 100 independent Stage-1 caches ($N = 500$ samples per cache).

β	M	None	Constant	Adaptive
0.5	4	0.1515 ± 0.0060	0.1123 ± 0.0047	$\mathbf{0.0994 \pm 0.0048}$
	8	0.1080 ± 0.0045	0.0709 ± 0.0029	$\mathbf{0.0648 \pm 0.0029}$
	16	0.0756 ± 0.0029	0.0435 ± 0.0015	$\mathbf{0.0414 \pm 0.0014}$
	32	0.0530 ± 0.0020	0.0301 ± 0.0011	$\mathbf{0.0297 \pm 0.0011}$
1	4	0.1421 ± 0.0045	0.1067 ± 0.0034	$\mathbf{0.0959 \pm 0.0036}$
	8	0.1022 ± 0.0034	0.0662 ± 0.0021	$\mathbf{0.0604 \pm 0.0022}$
	16	0.0717 ± 0.0027	0.0414 ± 0.0014	$\mathbf{0.0393 \pm 0.0014}$
	32	0.0497 ± 0.0018	0.0273 ± 0.0010	$\mathbf{0.0268 \pm 0.0010}$
2	4	0.1310 ± 0.0041	0.0989 ± 0.0032	$\mathbf{0.0910 \pm 0.0034}$
	8	0.0897 ± 0.0030	0.0597 ± 0.0020	$\mathbf{0.0549 \pm 0.0021}$
	16	0.0707 ± 0.0027	0.0423 ± 0.0014	$\mathbf{0.0401 \pm 0.0014}$
	32	0.0482 ± 0.0017	0.0276 ± 0.0009	$\mathbf{0.0271 \pm 0.0009}$
3	4	0.1128 ± 0.0041	0.0831 ± 0.0031	$\mathbf{0.0763 \pm 0.0033}$
	8	0.0818 ± 0.0026	0.0556 ± 0.0016	$\mathbf{0.0506 \pm 0.0017}$
	16	0.0642 ± 0.0017	0.0414 ± 0.0012	$\mathbf{0.0391 \pm 0.0013}$
	32	0.0437 ± 0.0014	0.0275 ± 0.0009	$\mathbf{0.0265 \pm 0.0009}$

To evaluate the clean-prediction approximation directly, we compute at each diffusion time

$$\mathrm{MSE}(t) = \frac{1}{N} \sum_{n=1}^{N} \|\hat{m}_t(X_t^{[n]}) - m_t^\pi(X_t^{[n]})\|^2, \tag{113}$$

where m_t^π is evaluated using one-dimensional numerical quadrature. We report the mean MSE across the 100 independent Stage-1 caches.

Table 3 shows the endpoint W_2 distance between the generated samples and the exact target, while Figure 3 shows the MSE of clean-prediction approximation.

B.2 Details on Image Generation (Classifier free guidance)

We use CREPE (He et al., 2026b) to sample from the classifier-free guidance (CFG) composition

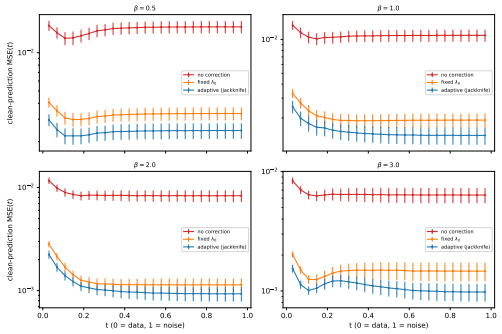
$$F_w[q_0^{(1)}(x_0), q_0^{(2)}(x_0)] = \frac{q_0^{(1)}(x_0)^w}{q_0^{(2)}(x_0)^{w-1}}, \tag{114}$$

where $q_t^{(1)}$ and $q_t^{(2)}$ denote the conditional and unconditional diffusion models, respectively. We use $\epsilon = 0.04377472$, $\eta = 0.05$, $\lambda_0 = 0.5$, and $T = 2048$.

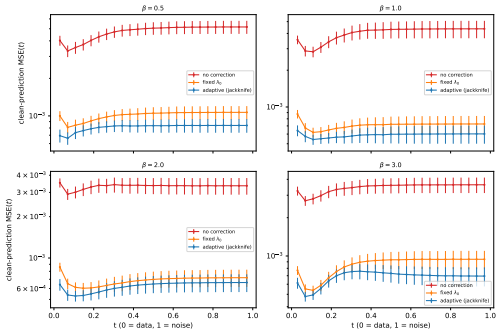
We evaluate five guidance scales, $w \in \{1.5, 1.75, 2.0, 2.25, 2.5\}$. For each guidance scale and each of 15 independent seeds, we run CREPE for 168 samples, discard the first 40 samples as burn-in, and retain the subsequent $K = 128$ samples. This gives $128 \times 15 \times 5 = 9600$ post-burn-in CREPE samples in total. For AmCo, we evaluate $M \in \{4, 8, 16\}$.

AmCo. For each seed and each guidance scale w, we use the first M post-burn-in CREPE samples as Stage 1 and generate an additional $N = K - M$ samples using Stage 2. We use $T' = 64$ and the conditional model, $i = 1$, as the reference model for the finite-sample correction. Thus, each guidance scale uses a separate Stage-1 computation and produces a set of K samples consisting of M Stage-1 samples and $N = K - M$ Stage-2 samples.

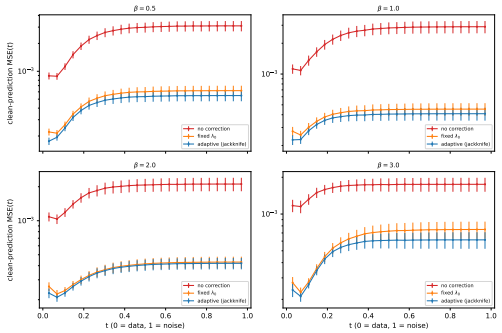
Evaluation metrics. For each method, seed, and guidance scale, we evaluate a set of K generated images. We report CLIP score and ImageReward at the individual-sample level, including their mean, standard deviation, and maximum. We additionally report the mean and standard deviation of diversity in the CLIP image-embedding space Venkatraman et al. (2024) across seeds. Table 4 shows that AmCo gives similar or better performance.



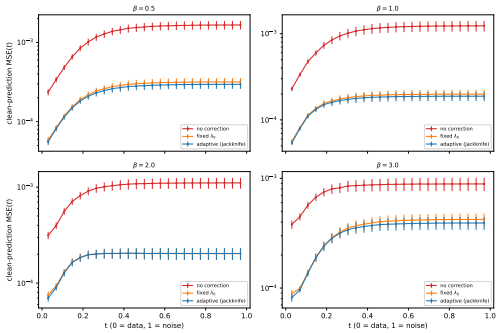
(a) Clean-prediction error, $M = 4$



(b) Clean-prediction error, $M = 8$



(c) Clean-prediction error, $M = 16$



(d) Clean-prediction error, $M = 32$

Figure 3: MSE of clean-prediction approximation.

Computational cost. We report (i) the average FLOPs required to generate one post-burn-in CREPE sample, (ii) the average FLOPs required to generate one Stage-2 AmCo sample, and (iii) the total FLOPs required to run the full experiment for CREPE and AmCo. In addition to the per-step model-evaluation cost, CREPE incurs an additional cost C due to chain initialization and the generation of burn-in samples; the same Stage-1 CREPE procedure is also used by AmCo. Thus, the total cost is $C + (M + N)TI$ for CREPE and $C + MTI + NT'$ for AmCo. Table 5 shows the resulting cost comparison.

B.3 Details on Ligand Generation (Product of Experts)

We evaluate AmCo on product-of-experts (PoE) ligand generation using the Feynman–Kac Corrector (FKC) of Skreta et al. (2025a) as the base composition method. We consider the composition

$$F_\beta[q_0^{(1)}(x_0), q_0^{(2)}(x_0)] = \big(q_0^{(1)}(x_0) q_0^{(2)}(x_0)\big)^\beta, \tag{115}$$

where $q_0^{(1)}$ and $q_0^{(2)}$ correspond to the ligand distributions associated with the two protein pockets. We evaluate three composition strengths, $\beta \in \{1.0, 1.5, 2.0\}$.

We use $T = 1000$, $T' = 64$, $\eta = 0.05$, $\lambda_0 = 0.5$, and $\epsilon = 0.05$. For the baseline FKC runs, we turn off resampling at $t = \epsilon$. We consider seven protein pairs and one molecule size, and evaluate $(M, N) \in \{(4, 4), (4, 12), (4, 28), (8, 8), (8, 24), (16, 16)\}$.

For each configuration, let $K = M + N$ denote the number of samples evaluated for each target composition strength.

Baseline FKC. For each protein pair, each β, and each (M, N) configuration, we run FKC with $K = M + N$ particles. If an FKC run collapses and produces invalid molecules, we rerun it with the next seed until a valid set of molecules is obtained. (The average number of retries was 0.226.)

Table 4: Image generation results across guidance scales w and Stage-1 anchor counts M ($K = 128$, 15 seeds, $T' = 64$, $\eta = 0.05$, $\lambda_0 = 0.5$). CLIP and ImageReward (IR) are reported as mean $\pm$ std / max over all pooled samples; Diversity is the mean $\pm$ std (across seeds) of pairwise CLIP-embedding distance. Bold marks the best (highest) value per column within each w block. CREPE is the baseline and does not depend on M.

w	Method	M	CLIP (↑)	CLIP max	IR (↑)	IR max	Diversity (↑)
1.5	CREPE	–	66.12 ± 5.80	76.81	-0.058 ± 0.833	1.455	0.0467 ± 0.0214
	AmCo	4	**67.31 ± 5.65**	78.44	-0.049 ± 0.802	1.331	0.0538 ± 0.0240
		8	67.12 ± 5.71	**78.75**	-0.039 ± 0.814	**1.492**	0.0568 ± 0.0208
		16	**67.31 ± 5.29**	76.75	**-0.014 ± 0.822**	1.350	**0.0569 ± 0.0194**
1.75	CREPE	–	67.00 ± 5.31	77.94	0.116 ± 0.949	1.504	0.0395 ± 0.0205
	AmCo	4	**68.19 ± 5.79**	**82.06**	0.138 ± 0.887	1.518	**0.0582 ± 0.0300**
		8	68.12 ± 5.60	77.94	0.182 ± 0.892	**1.569**	0.0550 ± 0.0193
		16	**68.19 ± 5.50**	79.44	**0.188 ± 0.901**	1.552	0.0528 ± 0.0219
2.0	CREPE	–	66.31 ± 4.78	77.62	0.125 ± 0.974	1.482	0.0313 ± 0.0150
	AmCo	4	**67.62 ± 4.74**	**81.06**	0.143 ± 0.892	1.576	0.0457 ± 0.0265
		8	67.44 ± 4.58	**81.06**	0.155 ± 0.896	1.598	**0.0498 ± 0.0232**
		16	67.19 ± 4.54	79.06	**0.179 ± 0.923**	**1.674**	0.0470 ± 0.0228
2.25	CREPE	–	67.06 ± 4.62	76.50	0.196 ± 0.854	1.449	0.0264 ± 0.0135
	AmCo	4	68.06 ± 3.96	77.31	0.260 ± 0.815	1.609	0.0396 ± 0.0241
		8	**68.19 ± 3.84**	**79.06**	**0.268 ± 0.809**	1.596	0.0373 ± 0.0113
		16	68.00 ± 3.79	77.69	0.260 ± 0.817	**1.635**	**0.0420 ± 0.0149**
2.5	CREPE	–	66.75 ± 4.49	76.50	0.225 ± 0.863	1.525	0.0280 ± 0.0168
	AmCo	4	67.75 ± 3.87	76.00	**0.247 ± 0.860**	1.625	0.0301 ± 0.0114
		8	**67.88 ± 3.73**	76.31	0.233 ± 0.871	1.648	0.0363 ± 0.0160
		16	67.62 ± 3.81	**76.62**	0.243 ± 0.858	**1.653**	**0.0413 ± 0.0161**

Table 5: Computational cost (FLOPs). CREPE's cost is the marginal FLOPs to obtain one additional post-burn-in sample by continuing the same PT chain; AmCo's Stage-2 cost is the marginal FLOPs to generate one additional sample from the amortized Stage 2, and is the same regardless of M. Total experiment FLOPs are for the full 5-scale, 15-seed protocol at each M.

	FLOPs/sample	Total experiment FLOPs (Speedup×)		
Method		$M = 4$	$M = 8$	$M = 16$
CREPE (excluding C)	4.174×10^{14}	4.007×10^{18}		
AmCo (excluding C)	6.522×10^{12} (64.0×)	1.859×10^{17} (21.6×)	3.091×10^{17} (13.0×)	5.557×10^{17} (7.2×)
CREPE (including C)	4.174×10^{14}	5.089×10^{18}		
AmCo (including C)	6.522×10^{12} (64.0×)	1.268×10^{18} (4.0×)	1.391×10^{18} (3.7×)	1.638×10^{18} (3.1×)

AmCo. For each protein pair and each composition strength β, we independently run FKC with M particles and use these samples as Stage 1. We then generate an additional $N = K - M$ samples using Stage 2 with the same target β. Thus, each value of β uses a separate Stage-1 computation and produces K samples consisting of M Stage-1 samples and N Stage-2 samples.

Evaluation metrics. For each method, protein pair, composition strength, and (M, N) configuration, we evaluate a set of K generated molecules. We select the molecule with the largest $P_1 \times P_2$, where P_1 and P_2 denote the target scores for the two protein pockets, and report its docking score against pocket 1 and pocket 2. We additionally report molecular diversity over the K generated samples. Table 6 shows that AmCo gives similar or better performance.

Computational cost. We report (i) the average FLOPs required to generate one FKC sample, including the cost of failed FKC generations, (ii) the average FLOPs required to generate one Stage-

Table 6: Ligand generation results across composition strengths β, particle budgets $K = M + N$, and methods (7 protein pairs, $T = 1000$, $T' = 64$, $\eta = 0.05$, $\lambda_0 = 0.5$, $\epsilon = 0.05$). For each protein pair, we pick the single sample maximizing $P_1 \times P_2$ and report its docking affinities; P_1/P_2 (kcal/mol) are the mean $\pm$ std of that value over the 7 pairs. Diversity is the mean $\pm$ std (across pairs) of $1-$mean-pairwise-Tanimoto-similarity. FKC (the baseline) depends only on K, not on how it is split into (M, N); AmCo is broken out by every (M, N) split of that K. Bold marks the best value per column within each (β, K) block.

β	K	Method	(M, N)	P_1 (↓)	P_2 (↓)	Diversity (↑)
1.0	8	FKC	–	-9.57 ± 1.79	-8.69 ± 1.24	0.23 ± 0.18
		AmCo	$(4, 4)$	**-9.89 ± 1.31**	**-9.13 ± 1.35**	**0.55 ± 0.07**
	16	FKC	–	-8.96 ± 1.51	-8.69 ± 1.27	0.21 ± 0.18
		AmCo	$(4, 12)$	**-9.93 ± 1.35**	**-9.34 ± 1.33**	**0.60 ± 0.08**
			$(8, 8)$	-9.83 ± 1.70	-9.15 ± 1.12	0.57 ± 0.04
	32	FKC	–	-9.24 ± 1.08	-9.05 ± 1.21	0.18 ± 0.13
		AmCo	$(4, 28)$	-9.83 ± 1.44	**-9.55 ± 1.33**	0.61 ± 0.10
			$(8, 24)$	**-10.07 ± 1.63**	-9.13 ± 0.97	**0.64 ± 0.05**
			$(16, 16)$	-9.95 ± 1.77	-9.46 ± 1.40	0.57 ± 0.10
1.5	8	FKC	–	-8.72 ± 1.24	-9.18 ± 1.45	0.14 ± 0.10
		AmCo	$(4, 4)$	**-9.66 ± 1.35**	**-9.40 ± 0.92**	**0.45 ± 0.11**
	16	FKC	–	-9.13 ± 1.36	-9.04 ± 1.29	0.28 ± 0.10
		AmCo	$(4, 12)$	**-9.95 ± 1.10**	-9.35 ± 1.07	**0.52 ± 0.07**
			$(8, 8)$	-9.67 ± 1.93	**-9.76 ± 1.57**	0.48 ± 0.08
	32	FKC	–	-8.79 ± 1.03	-8.79 ± 1.53	0.18 ± 0.18
		AmCo	$(4, 28)$	**-10.00 ± 1.19**	-9.57 ± 0.77	0.55 ± 0.05
			$(8, 24)$	-9.95 ± 1.97	**-9.78 ± 1.51**	**0.56 ± 0.09**
			$(16, 16)$	-9.87 ± 1.24	-9.60 ± 1.37	0.55 ± 0.07
2.0	8	FKC	–	-9.41 ± 0.93	-8.91 ± 1.02	0.39 ± 0.17
		AmCo	$(4, 4)$	**-10.07 ± 1.42**	**-9.67 ± 1.74**	**0.53 ± 0.08**
	16	FKC	–	-9.27 ± 1.49	-8.78 ± 1.43	0.27 ± 0.15
		AmCo	$(4, 12)$	**-10.15 ± 1.37**	**-9.81 ± 1.62**	0.57 ± 0.07
			$(8, 8)$	-9.91 ± 1.00	-9.35 ± 1.06	**0.59 ± 0.09**
	32	FKC	–	-8.82 ± 1.49	-8.99 ± 1.67	0.32 ± 0.16
		AmCo	$(4, 28)$	**-10.30 ± 1.38**	**-10.01 ± 1.54**	0.57 ± 0.07
			$(8, 24)$	-10.15 ± 0.96	-9.51 ± 1.35	**0.60 ± 0.09**
			$(16, 16)$	-9.65 ± 1.26	-9.42 ± 1.65	0.50 ± 0.10

Table 7: Computational cost (network-forward FLOPs). FKC's cost is the average FLOPs to obtain one sample, amortizing the cost of any collapsed (0-valid) generation attempts that had to be retried with a new seed; AmCo's Stage-2 cost is the marginal FLOPs to generate one additional sample from the amortized Stage 2. Total experiment FLOPs are for the full three-β ($\beta \in \{1.0, 1.5, 2.0\}$) protocol at each (M, N), averaged over the 7 protein pairs.

	FLOPs/sample	Total experiment FLOPs (speedup vs. FKC)					
Method		$(4, 4)$	$(4, 12)$	$(4, 28)$	$(8, 8)$	$(8, 24)$	$(16, 16)$
FKC	1.828×10^{14}	3.497×10^{15}	1.039×10^{16}	1.685×10^{16}	1.039×10^{16}	1.685×10^{16}	1.685×10^{16}
AmCo	5.466×10^{12}	2.228×10^{15}	2.359×10^{15}	2.621×10^{15}	3.629×10^{15}	3.891×10^{15}	1.065×10^{16}

2 AmCo sample, and (iii) the total FLOPs required to run the full three-β experiment for FKC and AmCo. Table 7 shows the cost comparison.

Table 8: Ablation on the reference correction $\lambda_t(X_t)$ for AmCo at each guidance scale w ($K = 128$, 15 seeds, $T' = 64$), with CREPE included as a baseline. CLIP and ImageReward (IR) are reported as mean $\pm$ std / max over all pooled samples, and Diversity is the mean (across seeds) pairwise distance in the CLIP image-embedding space. Nearest-anchor distance is the mean CLIP-embedding distance from each Stage-2 sample to its closest Stage-1 sample. For CREPE, we treat the first M post-burn-in samples as Stage-1 samples and the remaining $N = K - M$ samples as Stage-2 samples for this metric. Bold marks the best (highest) value per column within each M group among the three AmCo correction modes; the baseline is shown for reference.

$w = 1.5$

M	$\lambda_t(X_t)$ mode	CLIP (↑)	CLIP max (↑)	IR (↑)	IR max (↑)	Diversity (↑)	Nearest dist.
4	CREPE	66.12 ± 5.80	76.81	-0.058 ± 0.833	1.455	0.0467 ± 0.0214	0.0329 ± 0.0189
	No corr.	67.31 ± 5.43	77.38	**0.088 ± 0.876**	1.353	0.0153 ± 0.0077	0.0793 ± 0.0334
	Constant	**67.44 ± 5.38**	78.25	-0.017 ± 0.808	1.315	0.0435 ± 0.0179	0.0821 ± 0.0306
	Adaptive	67.31 ± 5.65	**78.44**	-0.049 ± 0.802	1.331	**0.0538 ± 0.0240**	**0.0837 ± 0.0315**
8	CREPE	66.12 ± 5.80	76.81	-0.058 ± 0.833	1.455	0.0467 ± 0.0214	0.0295 ± 0.0164
	No corr.	67.12 ± 5.51	77.56	**0.092 ± 0.873**	1.380	0.0224 ± 0.0101	0.0735 ± 0.0304
	Constant	**67.31 ± 5.39**	77.12	-0.011 ± 0.803	1.354	0.0484 ± 0.0185	0.0767 ± 0.0277
	Adaptive	67.12 ± 5.71	**78.75**	-0.039 ± 0.814	**1.492**	**0.0568 ± 0.0208**	**0.0783 ± 0.0286**
16	CREPE	66.12 ± 5.80	76.81	-0.058 ± 0.833	1.455	0.0467 ± 0.0214	0.0270 ± 0.0153
	No corr.	67.12 ± 5.46	**77.00**	**0.090 ± 0.870**	1.365	0.0331 ± 0.0128	0.0707 ± 0.0282
	Constant	**67.31 ± 5.33**	76.94	-0.014 ± 0.810	1.350	0.0569 ± 0.0201	**0.0744 ± 0.0262**
	Adaptive	**67.31 ± 5.29**	76.75	-0.014 ± 0.822	1.350	**0.0569 ± 0.0194**	0.0717 ± 0.0262

$w = 1.75$

M	$\lambda_t(X_t)$ mode	CLIP (↑)	CLIP max (↑)	IR (↑)	IR max (↑)	Diversity (↑)	Nearest dist.
4	CREPE	67.00 ± 5.31	77.94	0.116 ± 0.949	1.504	0.0395 ± 0.0205	0.0277 ± 0.0175
	No corr.	**68.44 ± 5.29**	77.56	**0.269 ± 0.902**	1.536	0.0134 ± 0.0077	0.0793 ± 0.0321
	Constant	68.31 ± 5.48	78.62	0.193 ± 0.887	**1.559**	0.0398 ± 0.0200	0.0881 ± 0.0393
	Adaptive	68.19 ± 5.79	**82.06**	0.138 ± 0.887	1.518	**0.0582 ± 0.0300**	**0.0916 ± 0.0398**
8	CREPE	67.00 ± 5.31	77.94	0.116 ± 0.949	1.504	0.0395 ± 0.0205	0.0245 ± 0.0156
	No corr.	**68.31 ± 5.33**	77.62	**0.270 ± 0.907**	1.552	0.0208 ± 0.0106	0.0735 ± 0.0345
	Constant	68.19 ± 5.52	**78.19**	0.199 ± 0.893	1.543	0.0453 ± 0.0213	0.0815 ± 0.0402
	Adaptive	68.12 ± 5.60	77.94	0.182 ± 0.892	**1.569**	**0.0550 ± 0.0193**	**0.0835 ± 0.0396**
16	CREPE	67.00 ± 5.31	77.94	0.116 ± 0.949	1.504	0.0395 ± 0.0205	0.0221 ± 0.0146
	No corr.	**68.25 ± 5.35**	77.56	**0.254 ± 0.913**	1.546	0.0312 ± 0.0134	0.0687 ± 0.0308
	Constant	68.12 ± 5.47	78.44	0.183 ± 0.899	1.542	0.0525 ± 0.0216	**0.0757 ± 0.0361**
	Adaptive	68.19 ± 5.50	**79.44**	0.188 ± 0.901	**1.552**	**0.0528 ± 0.0219**	0.0738 ± 0.0366

$w = 2.0$

M	$\lambda_t(X_t)$ mode	CLIP (↑)	CLIP max (↑)	IR (↑)	IR max (↑)	Diversity (↑)	Nearest dist.
4	CREPE	66.31 ± 4.78	77.62	0.125 ± 0.974	1.482	0.0313 ± 0.0150	0.0217 ± 0.0126
	No corr.	67.38 ± 4.20	77.50	**0.241 ± 0.900**	1.608	0.0122 ± 0.0072	0.0810 ± 0.0457
	Constant	67.50 ± 4.50	78.31	0.187 ± 0.902	**1.624**	0.0352 ± 0.0168	0.0869 ± 0.0416
	Adaptive	**67.62 ± 4.74**	**81.06**	0.143 ± 0.892	1.576	**0.0457 ± 0.0265**	**0.0884 ± 0.0390**
8	CREPE	66.31 ± 4.78	77.62	0.125 ± 0.974	1.482	0.0313 ± 0.0150	0.0200 ± 0.0117
	No corr.	67.25 ± 4.30	77.50	**0.232 ± 0.918**	1.590	0.0185 ± 0.0102	0.0759 ± 0.0434
	Constant	**67.44 ± 4.54**	78.69	0.185 ± 0.908	**1.601**	0.0406 ± 0.0198	0.0828 ± 0.0405
	Adaptive	**67.44 ± 4.58**	**81.06**	0.155 ± 0.896	1.598	**0.0498 ± 0.0232**	**0.0843 ± 0.0385**
16	CREPE	66.31 ± 4.78	77.62	0.125 ± 0.974	1.482	0.0313 ± 0.0150	0.0179 ± 0.0107
	No corr.	67.19 ± 4.38	77.50	**0.224 ± 0.931**	1.588	0.0285 ± 0.0147	0.0706 ± 0.0416
	Constant	**67.31 ± 4.55**	**79.12**	0.175 ± 0.919	1.639	**0.0487 ± 0.0212**	**0.0773 ± 0.0381**
	Adaptive	67.19 ± 4.54	79.06	0.179 ± 0.923	**1.674**	0.0470 ± 0.0228	0.0740 ± 0.0385

$w = 2.25$

M	$\lambda_t(X_t)$ mode	CLIP (↑)	CLIP max (↑)	IR (↑)	IR max (↑)	Diversity (↑)	Nearest dist.
4	CREPE	67.06 ± 4.62	76.50	0.196 ± 0.854	1.449	0.0264 ± 0.0135	0.0186 ± 0.0116
	No corr.	67.81 ± 3.58	75.69	**0.332 ± 0.784**	1.612	0.0092 ± 0.0034	0.0679 ± 0.0335
	Constant	**68.12 ± 3.79**	77.06	0.268 ± 0.810	**1.615**	0.0305 ± 0.0102	0.0864 ± 0.0406
	Adaptive	68.06 ± 3.96	**77.31**	0.260 ± 0.815	1.609	**0.0396 ± 0.0241**	**0.0865 ± 0.0431**
8	CREPE	67.06 ± 4.62	76.50	0.196 ± 0.854	1.449	0.0264 ± 0.0135	0.0169 ± 0.0102
	No corr.	67.75 ± 3.65	75.69	**0.320 ± 0.797**	1.607	0.0144 ± 0.0056	0.0630 ± 0.0309
	Constant	68.12 ± 3.88	76.88	0.264 ± 0.814	**1.610**	0.0349 ± 0.0119	**0.0820 ± 0.0351**
	Adaptive	**68.19 ± 3.84**	**79.06**	0.268 ± 0.809	1.596	**0.0373 ± 0.0113**	0.0794 ± 0.0353
16	CREPE	67.06 ± 4.62	76.50	0.196 ± 0.854	1.449	0.0264 ± 0.0135	0.0157 ± 0.0097
	No corr.	67.75 ± 3.70	76.31	**0.305 ± 0.811**	1.610	0.0230 ± 0.0097	0.0606 ± 0.0292
	Constant	**68.00 ± 3.87**	77.62	0.250 ± 0.822	1.631	**0.0438 ± 0.0157**	**0.0787 ± 0.0335**
	Adaptive	**68.00 ± 3.79**	**77.69**	0.260 ± 0.817	**1.635**	0.0420 ± 0.0149	0.0751 ± 0.0332

$w = 2.5$

M	$\lambda_t(X_t)$ mode	CLIP (↑)	CLIP max (↑)	IR (↑)	IR max (↑)	Diversity (↑)	Nearest dist.
4	CREPE	66.75 ± 4.49	**76.50**	0.225 ± 0.863	1.525	0.0280 ± 0.0168	0.0204 ± 0.0146
	No corr.	67.75 ± 3.70	75.12	**0.314 ± 0.885**	**1.636**	0.0083 ± 0.0034	0.0679 ± 0.0311
	Constant	**67.81 ± 3.86**	75.75	0.248 ± 0.855	1.630	**0.0302 ± 0.0119**	**0.0889 ± 0.0379**
	Adaptive	67.75 ± 3.87	76.00	0.247 ± 0.860	1.625	0.0301 ± 0.0114	0.0872 ± 0.0382
8	CREPE	66.75 ± 4.49	**76.50**	0.225 ± 0.863	1.525	0.0280 ± 0.0168	0.0182 ± 0.0127
	No corr.	67.69 ± 3.73	75.69	**0.298 ± 0.893**	1.636	0.0133 ± 0.0051	0.0637 ± 0.0298
	Constant	67.75 ± 3.80	**76.50**	0.242 ± 0.862	**1.650**	0.0347 ± 0.0127	**0.0853 ± 0.0381**
	Adaptive	**67.88 ± 3.73**	76.31	0.233 ± 0.871	1.648	**0.0363 ± 0.0160**	0.0833 ± 0.0368
16	CREPE	66.75 ± 4.49	76.50	0.225 ± 0.863	1.525	0.0280 ± 0.0168	0.0166 ± 0.0111
	No corr.	67.62 ± 3.72	75.69	**0.295 ± 0.891**	1.630	0.0226 ± 0.0091	0.0610 ± 0.0292
	Constant	**67.69 ± 3.86**	**76.75**	0.241 ± 0.854	**1.654**	**0.0441 ± 0.0160**	**0.0818 ± 0.0364**
	Adaptive	67.62 ± 3.81	76.62	0.243 ± 0.858	1.653	0.0413 ± 0.0161	0.0767 ± 0.0372

B.4 Ablation on $\lambda_t(X_t)$.

To assess the role of the reference correction, we compare no correction $\lambda_t(X_t) = 0$, a fixed correction $\lambda_t(X_t) = \lambda_0$, and our adaptive jackknife estimate $\lambda_t(X_t) = \hat{\lambda}_t^{\mathrm{JK}}(X_t)$. As shown in Tables 8 and 9, removing the correction leads to a substantial loss of diversity, as Stage-2 samples tend to concentrate around a small number of Stage-1 samples. To quantify this behavior directly, we additionally report the nearest-anchor distance, defined for each Stage-2 sample as its distance to the closest Stage-1 sample. For the baseline methods, which do not have separate Stage-1 and Stage-2 procedures, we treat the first M samples as Stage-1 samples and the remaining N samples as Stage-2 samples when computing this metric. For image generation, distances are measured in the CLIP image-embedding space, while for ligand generation we use $1-$Tanimoto similarity. In image generation, applying the reference correction consistently increases the nearest-anchor distance relative to no correction, together with a substantial recovery in diversity. A similar trend is observed in ligand generation, indicating that the correction helps reduce the tendency of Stage-2 samples to remain concentrated around the finite Stage-1 cache.

Table 9: Effect of the λ_t correction mode on AmCo (dual-target ligand generation), including the FKC baseline and the nearest-anchor distance ($1-$Tanimoto similarity to the closest Stage-1 sample; for the baseline, the first M of its $K = M + N$ generated particles are treated as Stage-1 and the remaining N as Stage-2).

$\beta = 1.0$

(M, N)	$\lambda_t(X_t)$	$P_1 \times P_2$ (↑)	P_1 (↓)	P_2 (↓)	Div. (↑)	Anchor dist. (↑)
(4,4)	Baseline	84.44	-9.566	-8.692	0.228	0.141
	No corr.	88.83	-9.506	**-9.258**	0.518	0.648
	Constant	90.60	-9.766	-9.169	0.545	**0.675**
	Adaptive	**91.40**	**-9.886**	-9.132	**0.554**	0.670
(4,12)	Baseline	79.23	-8.961	-8.690	0.213	0.062
	No corr.	91.41	-9.677	**-9.367**	0.568	0.648
	Constant	**95.05**	**-10.109**	-9.299	0.586	**0.651**
	Adaptive	93.71	-9.931	-9.340	**0.604**	0.644
(4,28)	Baseline	84.42	-9.235	-9.053	0.182	0.074
	No corr.	92.77	-9.742	-9.444	0.551	0.649
	Constant	**97.10**	**-10.190**	-9.455	0.603	**0.650**
	Adaptive	94.58	-9.826	**-9.548**	**0.612**	0.642
(8,8)	Baseline	79.23	-8.961	-8.690	0.213	0.044
	No corr.	88.33	-9.741	-8.973	0.549	**0.615**
	Constant	88.99	-9.582	**-9.220**	0.542	0.581
	Adaptive	**90.64**	**-9.827**	-9.147	**0.569**	0.609
(8,24)	Baseline	84.42	-9.235	-9.053	0.182	0.068
	No corr.	91.27	-9.683	**-9.368**	0.577	**0.609**
	Constant	90.54	-9.703	-9.275	0.609	0.570
	Adaptive	**92.97**	**-10.072**	-9.133	**0.644**	0.608
(16,16)	Baseline	84.42	-9.235	-9.053	0.182	0.037
	No corr.	87.43	-9.497	-9.031	0.520	0.593
	Constant	91.22	-9.671	-9.222	0.561	0.644
	Adaptive	**95.64**	**-9.955**	**-9.459**	**0.572**	**0.662**

$\beta = 1.5$

(M, N)	$\lambda_t(X_t)$	$P_1 \times P_2$ (↑)	P_1 (↓)	P_2 (↓)	Div. (↑)	Anchor dist. (↑)
(4,4)	Baseline	80.34	-8.724	-9.183	0.137	0.059
	No corr.	87.23	-9.250	-9.375	0.450	0.495
	Constant	**94.10**	**-9.822**	**-9.472**	**0.463**	**0.511**
	Adaptive	91.71	-9.660	-9.400	0.451	0.493
(4,12)	Baseline	82.87	-9.129	-9.043	0.282	0.164
	No corr.	90.37	-9.530	-9.436	0.457	0.479
	Constant	**94.77**	-9.876	**-9.490**	0.512	**0.528**
	Adaptive	93.77	**-9.949**	-9.355	**0.518**	0.528
(4,28)	Baseline	77.41	-8.793	-8.793	0.175	0.086
	No corr.	92.00	-9.622	-9.506	0.440	0.480
	Constant	**96.59**	**-10.042**	-9.521	0.534	0.533
	Adaptive	96.41	-10.000	**-9.574**	**0.548**	**0.540**
(8,8)	Baseline	82.87	-9.129	-9.043	0.282	0.077
	No corr.	88.47	-9.163	-9.559	0.429	0.499
	Constant	91.83	-9.483	-9.551	0.470	0.545
	Adaptive	**95.57**	**-9.671**	**-9.757**	**0.480**	**0.572**
(8,24)	Baseline	77.41	-8.793	-8.793	0.175	0.033
	No corr.	89.43	-9.242	-9.580	0.465	0.512
	Constant	**100.85**	-9.895	**-10.092**	0.534	0.570
	Adaptive	98.24	**-9.946**	-9.784	**0.556**	**0.600**
(16,16)	Baseline	77.41	-8.793	-8.793	0.175	0.031
	No corr.	92.28	-9.576	-9.534	0.486	0.546
	Constant	94.16	-9.850	-9.445	0.531	0.573
	Adaptive	**95.45**	**-9.869**	**-9.599**	**0.550**	**0.604**

$\beta = 2.0$

(M, N)	$\lambda_t(X_t)$	$P_1 \times P_2$ (↑)	P_1 (↓)	P_2 (↓)	Div. (↑)	Anchor dist. (↑)
(4,4)	Baseline	84.08	-9.411	-8.912	0.386	0.126
	No corr.	91.65	-9.586	-9.474	0.459	0.514
	Constant	98.54	-10.032	**-9.689**	**0.534**	0.559
	Adaptive	**98.67**	**-10.073**	-9.668	0.527	**0.561**
(4,12)	Baseline	82.73	-9.272	-8.782	0.270	0.101
	No corr.	95.05	-9.888	-9.507	0.441	0.510
	Constant	**100.76**	**-10.181**	-9.784	**0.579**	**0.579**
	Adaptive	100.71	-10.154	**-9.806**	0.570	0.572
(4,28)	Baseline	80.51	-8.819	-8.986	0.321	0.164
	No corr.	94.99	-9.881	-9.507	0.418	0.508
	Constant	103.91	**-10.368**	-9.921	0.565	0.565
	Adaptive	**104.21**	-10.302	**-10.010**	**0.569**	**0.566**
(8,8)	Baseline	82.73	-9.272	-8.782	0.270	0.031
	No corr.	92.11	-9.644	-9.544	0.519	0.539
	Constant	**93.78**	-9.745	**-9.620**	0.592	0.614
	Adaptive	93.01	**-9.914**	-9.353	**0.594**	**0.640**
(8,24)	Baseline	80.51	-8.819	-8.986	0.321	0.073
	No corr.	93.48	-9.711	-9.598	0.501	0.539
	Constant	96.90	-9.969	**-9.692**	**0.607**	0.643
	Adaptive	**97.00**	**-10.151**	-9.509	0.604	**0.657**
(16,16)	Baseline	80.51	-8.819	-8.986	0.321	0.047
	No corr.	87.58	-9.300	-9.313	0.469	0.509
	Constant	91.43	**-9.689**	-9.334	0.492	0.526
	Adaptive	**91.90**	-9.645	**-9.415**	**0.504**	**0.556**