DIVERSE LOW-TEMPERATURE DIFFUSION SAMPLING THROUGH REPLICA EXCHANGE

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models achieve strong generative performance across domains. However, achieving high sample quality under hard physical and dynamical constraints, as in protein design and robotic manipulation, remains difficult. Existing inference-time methods either require a costly or unavailable verifier, or avoid a verifier at the cost of reduced diversity. To relieve this tension, we introduce Score Tempering with Exchange between Exploratory Replicas (STEER), a verifier-free inference-time method that replaces single-chain sharpening with global search using multiple diffusion chains. During sampling, neighboring chains, each at different sharpening strengths, exchange particles according to an energy-based acceptance rate, propagating exploration to outputted samples. Across a diverse range of domains, from synthetic multimodal distributions to image generation, STEER achieves a better fidelity–diversity trade-off and improves success rates on continuous-control tasks in both verifier-based and verifier-free settings. Our findings suggest that effective diffusion tempering requires going beyond local updates to population-based global search that enhance fidelity and diversity.

1 INTRODUCTION

Diffusion models have emerged as a powerful generative framework, achieving strong results in domains such as image (Croitoru et al., 2023), video (Blattmann et al., 2023), and text generation (Lou et al., 2023). Generation has extended to embodied applications such as protein structure design (Guo et al., 2024) and robotic control (Janner et al., 2022; Chi et al., 2025), where samples must satisfy functional constraints, rather than approximately resemble the training distribution. In these regimes, viable solutions fall within a narrow, sharply bounded target: in protein generation, even a mismatch on the scale of a single atom's width between substrate and enzyme can prevent bonding (Watson et al., 2023); in dexterous robotic manipulation, policies can fail as small initial errors compound over the course of a trajectory (Ross et al., 2011). These challenges motivate a crucial challenge for diffusion modeling: how can we more effectively generate high-quality samples?

Existing works guide samples toward a narrow solution set by relying on an external objective function to prune or correct samples that that fall outside of target bounds. Verifier-based methods such as Best-of-N (Singhal et al., 2025) and evolutionary search (Yang et al., 2024) perform an informed search over a state space of multiple candidates, using an external scorer as a mechanism to guide traversal toward high-likelihood regions. Reward-gradient methods instead use the external objective's gradient to guide the denoising trajectory directly toward high-likelihood samples, precluding the use of multiple candidates (Zhang et al., 2026; Dhariwal & Nichol, 2021; Bansal et al., 2024; Chung et al., 2022). However, both of these approaches necessitate a defined objective, which can be prohibitively expensive to evaluate, as with protein sampling, or difficult to specify precisely, as with robotic manipulation, where a single scalar reward rarely captures its many implicit constraints.

Instead of using an external objective, temperature sampling concentrates probability mass on high-likelihood modes tightly enough to impose an implicit distributional bound (Ackley et al., 1985). This orthogonal technique adapts the effective signal-to-noise ratio of the denoising chain, whether by descaling stochasticity (Ajay et al., 2023; Yim et al., 2023; Geffner et al., 2025) or upscaling score (Xu et al., 2025; Li et al., 2026). Such methods share a core limitation: they search the generative manifold *locally* and thus fail to escape local optima. Ultimately, temperature scaling methods search only the vicinity of the diffusion chain's current state, guided solely by the model's own denoising dynamics (Dhariwal & Nichol, 2021; Bansal et al., 2024; Chung et al., 2022). Indeed,

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.

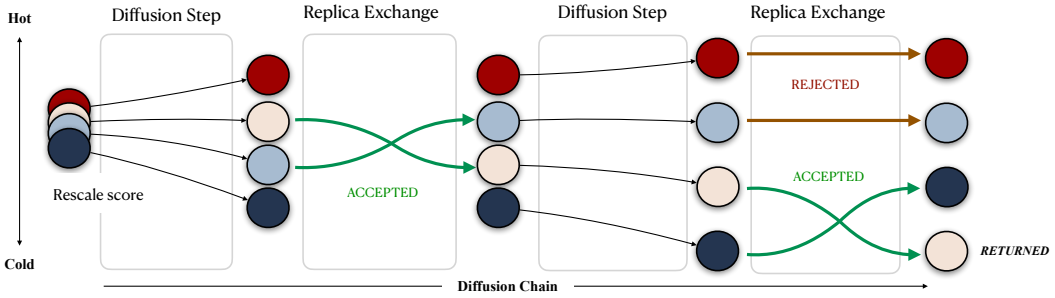


Figure 1: **Replica exchange allows high-temperature exploration to reach the low-temperature output.** High-temperature replicas yield improved global mode exploration, particularly in early timesteps when high-level details are cemented. Over successive rounds, a configuration discovered by the hottest chain is selectively swapped down into the lowest-temperature chain, the one ultimately reported.

temperature scaling methods commonly report reduced sample diversity or exhibit biased mode weights (Ajay et al., 2023; Yim et al., 2023; Geffner et al., 2025; Xu et al., 2025). Population-based methods attempt to recover this diversity; CREPE exchanges replicas indexed by diffusion timestep itself rather than an independent temperature (He et al., 2026b). Its hottest rung is merely a noised version of an existing replica, and thus its most exploratory swaps still compare against a state carrying minimal information about the target's structure. Thus, CREPE remains limited to its baseline diffusion chain and is fundamentally local. We argue that effective tempering requires combining the broad exploration of global search with the fine-grained refinement of local search.

We introduce Score Tempering with Exchange between Exploratory Replicas (STEER), which closes this gap. STEER runs multiple independent diffusion chains in parallel. Each chain runs at differing temperatures through approximate temperature sampling. A core property of our method is its integration of a swap mechanism between independent replicas. STEER accepts swaps between neighboring replicas under an energy-based acceptance rule, moving particles to a new chain only when it is well-suited there. By combining local search with global search, we can avoid being stuck in local optima, advancing the performance of base models. We verify the exchange mechanism's correctness analytically under simple distributions. Assuming the local energy landscape is Gaussian, the acceptance rule satisfies detailed balance and consequently samples from each chain's target distribution. Extensive empirical results across multiple domains, ranging from simple mixture of Gaussians to large-scale image generation, build upon the elemental theoretical picture described above. STEER outperforms existing sharpening baselines across every domain we study despite requiring no additional training or supervision, all while composing seamlessly with verifiers.

Contribution. We introduce STEER, a training- and verifier-free algorithm that couples local refinement by temperature-scaled diffusion with global exploration through replica exchange. Central to our method is an energy-based acceptance rule that guides exchanges across temperatures, for which we prove that swaps preserve the correct tempered distributions in the isotropic Gaussian case. Through synthetic experiments, we show that STEER reaches isolated modes that competing samplers struggle to access and reduces mode-coverage distortions observed under temperature-scaled sampling. We further demonstrate that STEER scales to high-dimensional image and protein generation, improving the diversity–fidelity tradeoff in Stable Diffusion and protein generation. Finally, we show that STEER improves downstream task success in imitation-learning settings, both with and without an external verifier. Together, these results highlight the value of coupling local refinement with global search through cross-chain exchange for low-temperature sampling.

2 Method

2.1 Preliminaries

Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020) corrupt $x_0 \sim p_{\text{data}}$ into pure noise via a forward process:

$$x_t = \alpha_t x_0 + \sigma_t \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I),$$

where α_t, σ_t are chosen so α_t^2/σ_t^2 decreases strictly in t, interpolating each marginal $x_t \sim p_t$ between data ($t = 0$) and noise ($t = T$). A network approximates the score, $s_\theta(x_t, t) \approx \nabla_{x_t} \log p_t(x_t)$, commonly as a noise predictor via $s_\theta(x_t, t) = -\epsilon_\theta(x_t, t)/\sigma_t$; generation reverses the process by following s_θ from x_T to x_0.

2.2 TEMPERATURE SAMPLING

We consider the problem of sampling from high-dimensional distributions in the low-temperature regime. As we concentrate samples around mode centers, an otherwise broad distribution sharpens and isolates modes from their neighbors, accentuating the target's multi-modal probability landscape. That is, we aim to sample from a distribution that satisfies scaling by temperature $\lambda > 0$:

$$\tilde{p}(x_0) \propto p(x_0)^{1/\lambda}. \tag{1}$$

Raising a density to a positive power preserves the location of each mode, but it notably reshapes the local curvature around mode centers. Locally, this scaling behaves like a Gaussian covariance rescaling by λ, sharpening the distribution for low-temperature values of $\lambda < 1$ and broadening it for high-temperature values of $\lambda > 1$, though the exact effect is mode-dependent in general.

Pseudo-Temperature Sampling. A natural expectation is that score rescaling is sufficient to recover $\tilde{p}$ exactly (Shih et al., 2023; Du et al., 2023). We show this to be false. Let $\tilde{p}_t$ denote the marginal at time t of $\tilde{p}$'s own forward diffusion process, and let $\tilde{s}_\theta(x_t, t)$ denote its associated score. Neither these marginals nor their associated score admit a simple closed form in terms of p and ϵ_θ:

$$\tilde{p}_t(x_t) \neq p_t(x_t)^{1/\lambda}, \quad \tilde{s}_\theta(x_t, t) \neq \frac{1}{\lambda} s_\theta(x_t, t).$$

Although the exact relation fails to hold in general, the marginals of $\tilde{p}$ still retain a useful time-dependent structure. At $t = 1$, where the signal has been fully suppressed by noise, a given score-rescaling constant $r_1(\lambda) = 1$, and the tempering has no effect; at $t = 0$, where the signal is clean, $r_0(\lambda) = \frac{1}{\lambda}$, recovering the exact rescaling of eq. 1. Intuitively, samples carry no information about x_0 near pure noise, so every temperature level induces the same score. As the signal-to-noise ratio grows, the marginal increasingly resembles the data distribution itself, where the exact tempering relation holds. $r_t(\lambda)$ correspondingly interpolates toward $1/\lambda$. We utilize this time-dependence to re-weight the effective signal to noise ratio, assuming a variance-preserving schedule ($\alpha_t^2 + \sigma_t^2 = 1$):

$$\tilde{s}_\theta(x_t, t) \approx \frac{1}{\alpha_t^2 \lambda + \sigma_t^2} s_\theta(x_t, t), \tag{2}$$

Other works use similar strategies (Xu et al., 2025; Li et al., 2026; Ajay et al., 2023) to achieve similar effective step-size scaling through score or noise rescaling. We show these methods are effectively equivalent in Appendix B.5 under a Langevin interpretation. Recovering $\tilde{s}_\theta$ exactly remains intractable, yet a cheap, guidance-free approximation to $\tilde{p}$ is still valuable. We ask instead how such rescaling can be paired with a correction mechanism that utilizes a population-wide search.

2.3 SCORE TEMPERING WITH REPLICA EXCHANGE

We integrate swaps between parallel replicas, formally known as *parallel tempering*, a method for multimodal Markov chain Monte Carlo (MCMC) sampling (Earl & Deem, 2005). Our algorithm simulates M chains with temperatures $\lambda_1 < \cdots < \lambda_M$, and adds a swap step between them: adjacent chains periodically propose to exchange states, accepted or rejected under an energy-based criterion we derive. The chain at λ_1 corresponds to the target distribution whose samples are retained, while the hotter chains survey broad volumes of the generative distribution to improve mixing.

What makes our method particularly adept at global exploration is this swap mechanism: while each individual chain applies only a pointwise, local correction to itself, swapping acts as a population-wide correction for the target. That is, because replica exchange compares all pairs over the diffusion trajectory, the outputted sample is whichever configuration the exchange dynamics determines is best-suited to the target temperature, not necessarily the one that started there. This leaves the critical question of how to propagate highly plausible samples through a selective acceptance ratio.

Energy Difference. What makes swapping probabilistically favorable? Suppose a high-temperature diffusion chain p^{λ_j} has discovered a higher-probability state x^j than the state x^i its adjacent, lower-temperature chain p^{λ_i} has discovered, where temperatures $i < j$. Under the low-temperature p^{λ_i}, the high-probability state x^j would hold greater likelihood than under its present high-temperature chain (by the sharpening properties described in eq. 1). In fact, under chain p^{λ_j}, state x^i would hold higher likelihood, since higher temperatures widen the variance that state x^i might plausibly fall under. In this example, swapping is favorable. We ultimately aim to swap when

Algorithm 1 STEER: Parallel Tempering with Replica Exchange

Require: M replicas, temperatures $\lambda_1 < \cdots < \lambda_M$

1: **Initialize** $x_T^{(i)} \sim \mathcal{N}(0, I)$ for each replica i
2: **for** each timestep $t = T, \ldots, 1$ **do**
3: $\quad \epsilon_\theta^{(i)} \leftarrow \epsilon_\theta(x_t^{(i)}, t)$ for each replica i ▷ Reused below
4: $\quad$ **for** each i where $(i + t) \bmod 2 = 0$ **do** ▷ Replica swap
5: $\quad\quad$ Swap $\lambda_i \leftrightarrow \lambda_{i+1}$ with probability $a(\epsilon_\theta^{(i)}, \epsilon_\theta^{(i+1)})$ ▷ eq. 4
6: $\quad x_{t-1}^{(i)} \leftarrow$ denoising step with $\epsilon_\theta^{(i)}$ rescaled by $r(\lambda_i)$, for each replica i
7: **return** Denoised samples from replica 1 (lowest temperature)

the joint probability of swapped states is higher than the joint probability of non-swapped states:

$$a_{ij} = \min\left(1, \frac{p^{\lambda_i}(x^j)\, p^{\lambda_j}(x^i)}{p^{\lambda_i}(x^i)\, p^{\lambda_j}(x^j)}\right).$$

Critically, this does not require an absolute energy value: the learned score fixes energy only up to an additive constant that cancels between the two replicas being compared. We make this precise.

Swap Acceptance. Let us define energy $E(x) := -\log p_t(x)$, and let $E_i := E(x_t^i)$ denote the energy of the state in chain i. For an adjacent pair of chains i and j, we define acceptance probability:

$$a_{ij} = \min\left(1,\, \exp\left[(E_i - E_j)\left(r_t(\lambda_i) - r_t(\lambda_j)\right)\right]\right), \tag{3}$$

where $r_t(\lambda)$ denotes the time-dependent scaling at diffusion timestep t defined in eq. 2. Swaps that increase the joint probability are always accepted, while other swaps are accepted probabilistically.

In practice, we only have access to the score, not its integral, energy. Modeling the score directly is more efficient for diffusion sampling and remains the standard approach. However, as described above, additive constants cancel out in the energy difference, and because the negative energy gradient score, the energy difference can be written exactly as a path-independent line integral of the score (Sjöberg et al., 2026). Substituting this path integral into the energy difference in eq. 3 yields:

$$a_{ij} = \min\left(1, \exp\left[-\left(\int_{x_t^j}^{x_t^i} s_\theta(x,t)\cdot dx\right)\cdot \left(r_t(\lambda_i) - r_t(\lambda_j)\right)\right]\right). \tag{4}$$

We adopt this procedure with a simple, two-point approximation to the line integral, which gives $E_i - E_j \approx -\frac{1}{2}\left(s_\theta(x_t^i, t) + s_\theta(x_t^j, t)\right)\cdot(x_t^i - x_t^j)$. This choice adds no computational overhead, since both score evaluations are already computed for each chain's own denoising step, whereas additional points would require evaluating interpolated score values solely for the swap decision. This is consistent with Sjöberg et al. (2026), who find that in a closely related line-integral MH correction, sampling-quality metrics are statistically indistinguishable between three and ten integration points, even as the underlying quadrature estimate itself keeps improving over that same range.

Computational effort. The swap step adds no extra score evaluations: ϵ_θ is computed once per replica per timestep, used to decide the swap on the pre-update states, then reused for that replica's tempered diffusion step – given a two-point integral, no score is ever evaluated twice, and each replica's injected noise stays independent. When a verifier is available, $M := N$ replicas are charged $M\times$ the number of model evaluations of a single chain, so all reported comparisons are compute-matched. Since chains run with batch parallelization, STEER shows no significant wall-clock overhead. Existing parallel tempering research shows that while M replicas cost roughly $M\times$ a single chain, exchange can still exceed $1/M$ the efficiency of standard sampling by letting low-temperature chains reach regions a single chain, even run $M\times$ longer, could not. Additionally, precluding the use of a possible costly verifier can reduce total inference-time computation.

2.4 Theoretical Rationale

Detailed Balance. While exact score rescaling presents the problems discussed in Section 2.2, under an isotropic Gaussian, the temperature-scaling approximation in eq. 4 becomes exact, as shown in Appendix B.1. Under this scenario, we show that STEER samples from the target distribution. We first show that swapping preserves detailed balance, a property that ensures the equilibrium among chains can be preserved under our swap kernel, and thus maintains that each replica to be representative of its respective temperature. We formalize how exchanges preserve detailed balance:

Theorem 2.1 (Detailed balance of the practical swap kernel under isotropic Gaussian data)**.** *Let $x_0 \sim \mathcal{N}(0, \mathbf{I})$, and consider two chains i, j tempered by λ_i, λ_j. The swap-acceptance rule of eq. 4 coincides exactly with the true Metropolis–Hastings acceptance probability for these chains' joint distribution, and therefore leaves it invariant.*

The full proof is in Appendix B.3. Practically: a configuration is more likely to be swapped into a chain whose temperature makes it probable and less likely into one where it would be an outlier. High-energy configurations then drift toward hotter chains, low-energy ones settle into colder chains.

Robustness to Tempering Bias. By the above theorem, we show that replica exchange cannot compound bias introduced by pseudo-temperature scaling. We show that swapping can only reshuffle what error already exists between chains, without a mechanism for inventing new error.

Proposition 2.1 (Swaps do not amplify tempering bias)**.** *Consider a swap-eligible pair of chains, and suppose the joint distribution of chains are off by the true joint target by at most $\varepsilon(\lambda)$ in total variation. Since our swap kernel K_{swap} is built from the exact acceptance rule of eq. 4 and admits π_t as a stationary point, applying it can only shrink the distance to π_t, never grow it. Thus, after any number of repeated swap attempts, the joint distribution stays within that same $\varepsilon(\lambda)$ of the true target.*

In Appendix B.4, we show this by comparing a biased joint distribution against the true joint distribution of the target, revealing that our swap mechanism can only shrink total-variation distance to the true joint. The proof follows from $\pi_t^{(i,j)}$'s stationarity under K_{swap}; the swap step can act on existing bias but cannot introduce new bias. This guarantee is for the idealized kernel, as our score-based estimator (eq. 4) carries its own separate approximation error (Section 2.3).

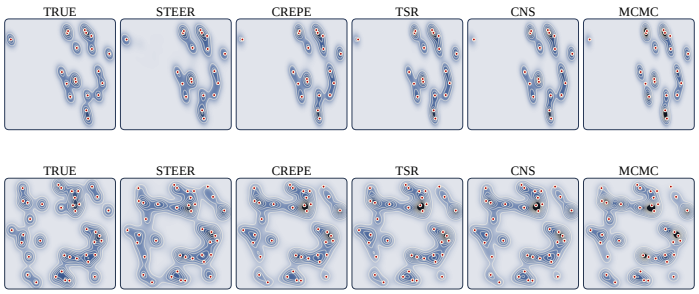
3 Experiments

In this section, we present experiments designed to answer three questions. (**Q1**) Does STEER yield improved fidelity while avoiding sacrifices in sample diversity? (**Q2**) How does STEER scale to high-dimensional generation benchmarks? (**Q3**) How does our algorithm improve alignment in downstream tasks? We structure our experiments to answer each of these questions. We first design two challenging Gaussian-mixture datasets in Section 3.1: a 20-mode dataset quantifying isolated-mode discovery and a 40-mode, equally-weighted dataset quantifying mode-weight preservation. We report experimental details in Appendix C.1 and additional results in Appendix D.1. Our second set of experiments in Section 3.2 expands this to Stable Diffusion 3 (Esser et al., 2024) and FoldingDiffusion protein generation (Wu et al., 2024). We include experimental details for Stable Diffusion in Appendix C.2 with extended precision-recall results, acceptance rate figures, and image samples in Appendix D.3. In Section 3.3, we evaluate STEER in continuous-control settings both with and without an external verifier: our PointMaze (Fu et al., 2020) benchmark in Section 3.3 uses a verifier to search successful candidate trajectories, while our Robomimic (Mandlekar et al., 2021) benchmark in Section 3.3 applies STEER directly during policy sampling. Across both, we measure how increased diversity aligns with strict success benchmarks in downstream tasks. Additional experimental details for PointMaze and Robomimic are in Appendices C.4 and C.5, respectively.

Baselines. We compare STEER against three inference-time temperature sampling approaches. First, we include single chain temperature baselines, including Constant Noise Scaling (CNS) (Ajay et al., 2023; Yim et al., 2023; Geffner et al., 2025) and Temporal Score Rescaling (TSR) (Xu et al., 2025; Li et al., 2026). While STEER can be combined with many temperature scaling methods, we choose to implement replica exchange over TSR, which shows strong temperature scaling gains with improved diversity over CNS (Xu et al., 2025). We additionally compare against CREPE (He et al., 2026b) to evaluate against a multiple-chain temperature approach. We evaluate CREPE under the same protocol, reducing its iteration count to match wall-clock time. Finally, we compare STEER against Unadjusted Langevin Algorithm (ULA), an MCMC corrector at each noise level, with temperature-scaled score (Roberts & Rosenthal, 1998; Du et al., 2023).

3.1 Toy Experiments

Isolated Mode Mixture Dataset To study STEER's effect on mode occupancy, we use a modified Gaussian mixture from Erives et al. (2025). Our dataset contains 19 low-density modes with shared



Method	W_2 (↓)	TV (↓)	Cov% (↑)
MCMC	1.023 ± 0.029	0.169 ± 0.005	28.4%
CNS	1.001 ± 0.083	0.114 ± 0.006	28.4%
TSR	0.941 ± 0.095	0.110 ± 0.004	36.8%
CREPE	0.922 ± 0.101	0.102 ± 0.006	37.9%
STEER	$\mathbf{0.545 \pm 0.047}$	$\mathbf{0.096 \pm 0.004}$	**89.7%**

Method	W_2 (↓)	TV (↓)	Entropy (↑)
CREPE	1.555 ± 0.103	0.207 ± 0.009	0.966 ± 0.003
TSR	1.578 ± 0.105	0.218 ± 0.009	0.962 ± 0.003
CNS	1.613 ± 0.104	0.218 ± 0.009	0.962 ± 0.003
MCMC	1.541 ± 0.069	0.316 ± 0.010	0.924 ± 0.004
STEER	$\mathbf{1.116 \pm 0.090}$	$\mathbf{0.204 \pm 0.003}$	$\mathbf{0.968 \pm 0.001}$

Figure 2: **STEER discovers isolated modes and preserves mode coverage.** Top: Mixture of 20-mode Gaussians with an isolated, high-density mode. Bottom: Mixture of 40-mode Gaussians with equal weight and variance. Across datasets, STEER outperforms other temperature-scaling methods on all metrics.

standard deviation, equal weights, and uniformly sampled means, as well as one isolated, high-density mode at a fixed distance away. A single MLP is trained once and reused across all methods (See Appendix C.1 for additional dataset and training details). We sample at $\lambda_1 = 0.9$, comparing tempered CREPE, TSR, CNS, and MCMC against STEER. We implement a temperature ladder consisting of 3 replicas linearly spaced until $\lambda_M = 16.0$, over 5 seeds, aggregating 5000 samples in total. We compare against the true tempered target, which we generate with fine grid sampling.

Results. Table 2 reports W_2, TV, and hard-mode coverage (Cov), measuring spatial mass placement, hard-mode probability allocation, and the rate of observed hard-mode samples, respectively. STEER outperforms across all three metrics. Local methods, including single-chain temperature scaling and CREPE, struggle to cross the low-probability region separating the isolated mode from the main cluster. STEER's high-temperature chains reduce effective energy barriers, facilitating discovery of the isolated mode, while replica exchange transfers these discoveries to the low-temperature chain. These results highlight the benefits of coupling local refinement with global search.

Uniform Mixture of 40 Gaussians We construct a second toy dataset consisting of 40 isotropic Gaussian components with equal weights and variances and means placed uniformly at random. Because all components are identical up to location, tempering by any λ preserves exact uniformity over all modes, isolating uneven concentration as a sampler artifact. We design a temperature ladder spaced linearly among 3 replicas from $\lambda_1 = 0.9$ to $\lambda_M = 4.0$. We run each method across five seeds, with $1,000$ samples each. The choice of temperatures is discussed in ablations in Appendix D.1.

Results. Because every mode in this mixture carries equal weight and variance, it provides a clean testbed for detecting whether temperature-scaled samples have collapsed onto a subset of modes. We evaluate each method with W_2 distance; entropy, a measure of the spread of a distribution in comparison to the target; and TV distance, a measure of the mass that never showed up in generated samples. STEER attains the strongest results across all methods and all metrics (Table 2), revealing how replica exchange partially mitigates the biases of pseudo-temperature sampling. We report auxiliary chain samples, as well as high-temperature and reward-tilting examples, in Appendix D.1.

Remarks. Across both datasets, STEER outperforms every other method, on every metric. Implementing replica exchange not only enables strong global search, as evinced by results in dataset 1, but also preserves mode weights in doing so, as shown in the second dataset. Altogether, our method proves itself to be a strong low-temperature sampler that preserves proper mode coverage.

3.2 High-Dimensional Generation

Higher-fidelity samples risk collapsing onto fewer modes, making diversity especially important to preserve as sample quality increases. To test this at scale, we evaluate sample diversity along the fidelity-alignment Pareto frontier in high-resolution image and protein generation (**Q2**), two high-dimensional settings distinct from the temporally coupled trajectories evaluated elsewhere.

Stable Diffusion As an experimental baseline at scale, we evaluate STEER on Stable Diffusion 3, reporting how fidelity and diversity are compromised with greater alignment between text prompts. Each point on the frontier reports an FID and CLIP score that corresponds to the lowest-temperature chain among M replicas. For a given number of replicas M, we set the highest replica temperature to $\lambda_M = \lambda_1 + 0.2$ for every configuration. This fixed temperature gap was chosen to hold the



Figure 3: **STEER enhances per-sample fidelity.** Comparison of 1 replica vs. 4 replicas at $\lambda = 0.95, 1.05, 1.1$, for a variety of different prompts (top to bottom); individual samples remain visually consistent across replica counts, with the increased number of replicas improving detail and diversity without degrading semantic quality.

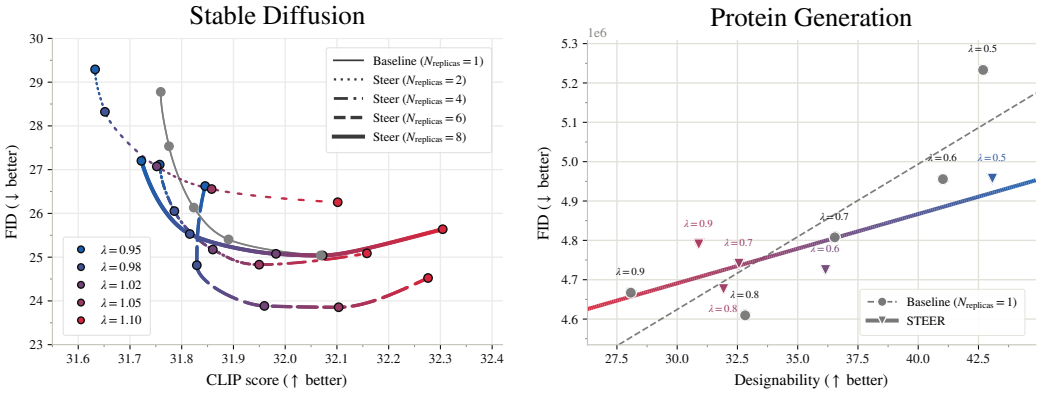


Figure 4: **STEER improves population-wide fidelity while maintaining alignment** Left: STEER shifts the fidelity-diversity frontier outward on Stable Diffusion. Right: each point is a temperature λ on protein backbone generation; STEER shifts the fidelity-designability trendline favorably.

temperature spread constant across configurations. Each point represents $10,000$ samples compared against the COCO 10K dataset (Lin et al., 2014). We include high-temperature chains, where $\lambda > 1$, in the pareto frontier, as high-temperature chains have been shown to improve diversity without much loss to fidelity (Xu et al., 2025). Additional experimental details are in Appendix C.2.

Results. Figure 4 shows the FID–CLIP frontier for TSR and STEER for varying numbers of replicas, though only the lowest temperature replica is evaluated. Replica exchange shifts the frontier outward most significantly at six replicas, where STEER beats TSR's pareto curve in both FID and CLIP; the gain shrinks at four and two replicas. Eight replicas reach a slightly higher maximum CLIP but a substantially worse FID than six chains. We believe that with a fixed temperature range, more replicas shrink the adjacent temperature ratio, pushing swap acceptance toward unity and causing replicas to overmix, a known exchange-rate/exploration trade-off in parallel tempering (Miasojedow

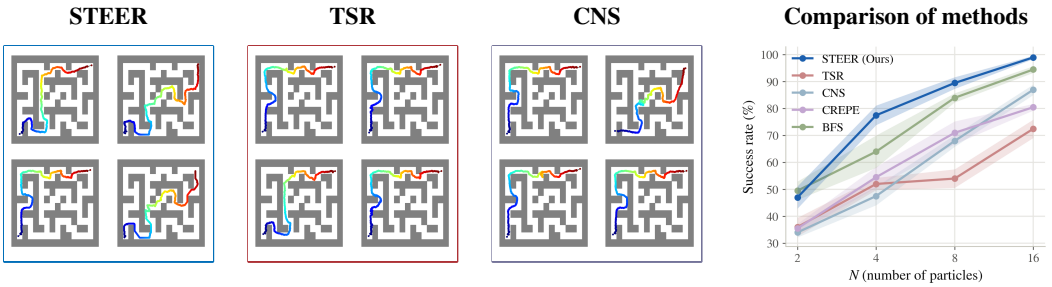


Figure 5: **STEER excels at long-range horizon planning when composed with verifiers**. **Left:** representative rollouts for CNS, TSR, and STEER, each shown as a 2×2 grid of frames. **Right:** Best-of-N success rate as a function of the number of replicas, with all methods run under a matched compute budget for each N.

et al., 2013). A well-chosen temperature ladder should therefore account for the number of replicas. We include additional precision–recall and acceptance rate results in App. D.3.

Protein Generation. We further evaluate STEER for unconditional protein backbone generation using FoldingDiff, a diffusion model that represents protein backbones through bond and dihedral angles and generates structures by denoising from an unstructured state (Wu et al., 2024). We generate short to long (50–128) residues, aggregating 780 structures. We run samples across a range of $\lambda_1 \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$ and ending at $\lambda_M = \lambda_1 + 0.2$, with a temperature ladder of eight linearly spaced replicas. We include additional training and evaluation details in Appendix C.3.

Results. To evaluate designability, we use ProteinMPNN (Dauparas et al., 2022) to generate candidate amino-acid sequences and fold these back into three-dimensional structures. We measure the TM-score between predicted refolded structures and generated backbones. We also compute ProtFID, a protein-specific Fréchet distance to measure distributional fidelity (Faltings et al., 2026). STEER shows improved results, with an improved overall trend line revealing how replica exchange improves the coverage of plausible, designable protein structures while preserving similarity to the reference protein distribution. In particular, STEER demonstrates both better FID and designability at $\lambda_1 = 0.5$, where our method is able to correct larger drift introduced by temperature rescaling.

Remarks. These results address **Q2**: in high-dimensional image and protein generation, STEER improves the trade-off between distributional fidelity and task-specific quality metrics. Replica exchange therefore preserves useful exploration as samples become sharper, improving alignment in image generation and designability in protein generation without discarding sample diversity.

3.3 Continuous Control

In this section, we evaluate STEER in continuous-control settings, where diversity is useful only where it produces coherent, executable behavior and consequently improves task success (**Q3**).

Verifiable Planning. We evaluate long-horizon planning on the PointMaze environment from the D4RL and Gymnasium-Robotics reinforcement-learning benchmarks. The environment stresses a planner's ability to construct coherent trajectories rather than greedy ones. In addition to our standard baselines, we compare against Breadth-First Search (BFS) (Zhang et al., 2026), which expands nodes with higher score estimates and backtracks from low-quality regions of the tree. All methods generate $N \in \{2, 4, 8, 16\}$ replicas over 5 seeds, with 40 rollouts each. We set target temperature $\lambda_1 = 0.9$ with linear spacing until $\lambda_N = \lambda_1 + 0.01 \cdot N$, and verify over all N samples. Because swaps reuse the score already computed at each timestep, STEER's Best-of-N runs at exactly the same score-evaluation cost as N independent chains, so its exploration gains do not come at the cost of extra compute. Additional experimental setup details are in Appendix C.4.

Results. Replica exchange improves Best-of-N success across all replica counts, most sharply at intermediate budgets: STEER with $N = 4$ already exceeds TSR's performance at $N = 16$, using only a quarter of the score evaluations. Among the non-exchange baselines, BFS is the closest competitor to STEER, with stronger gains at $N = 2$. This verifier setting is a particularly compelling benchmark, as our global search method enables the verifier to act on a population that has not collapsed. This builds on our toy examples in Section 3.1. Just as how samples in our toy datasets exhibit improved mode spread and isolated mode search, here, that same preserved diversity means the verifier can select among samples from viable modes that pseudo-temperature sampling would



Lift Can Square Tool Hang

Method	Lift	Can	Square	Tool Hang
Standard	99.2 ± 0.5	52.8 ± 3.2	86.0 ± 1.8	43.2 ± 2.4
CREPE	99.2 ± 0.4	50.8 ± 2.3	85.2 ± 2.9	46.0 ± 2.2
CNS	$\mathbf{99.6 \pm 0.4}$	50.4 ± 3.3	84.8 ± 3.0	42.0 ± 3.8
TSR	99.2 ± 0.5	50.8 ± 2.9	86.4 ± 2.8	44.0 ± 3.0
STEER	99.2 ± 0.5	$\mathbf{54.0 \pm 3.0}$	$\mathbf{89.6 \pm 2.3}$	$\mathbf{46.4 \pm 2.9}$

Figure 6: **STEER improves imitation learning success without any verifier or reward signal.** Success rate (%) on Robomimic tasks over five seeds; STEER exceeds every baseline on Can, Square, and Tool Hang, applying replica exchange directly to policy sampling with no post-hoc filtering.

have discarded. Thus, even with the use of other global search mechanisms like Best-of-N, our method shows a strong advantage in improving the population quality that search is performed over.

Imitation Learning. We evaluate STEER on Robomimic (Mandlekar et al., 2021), a benchmark for robot learning from demonstrations. This setting tests whether STEER can improve imitation policies trained without an explicit reward function. Unlike PointMaze, where a verifier searches over candidate trajectories, we apply swaps directly during policy sampling and execute the resulting action sequence without post-hoc filtering or reward guidance. This isolates whether replica exchange alone can improve trajectories generated by a diffusion policy. We evaluate Lift, Can, Square, and Tool Hang, training a separate diffusion policy for each task using the standard Robomimic pipeline. To avoid near-perfect success rates, we subsample the training demonstrations, retaining (10%) for Lift and Can and (75%) for Square and Tool Hang. We use four replicas with a linearly spaced temperatures from $\lambda_1 = 0.9$ to $\lambda_M = 1.1$. See Appendix C.5 for additional details.

Results. STEER improves success rate across three tasks (Figure 5). On Square, Tool Hang, and Can, where the subsampled task retains real difficulty, STEER's improvement is more pronounced; gains are obtained without an external verifier or post-hoc filtering, and therefore attributable to the sampling process itself rather than selection among independently generated candidates. For Lift, the lack of improvement reflects a saturation effect: success rates plateau near-perfect for every method, since even with subsampled demonstrations the task remains simple enough to leave no headroom for inference-time sampling to differentiate methods. While the absolute margins between STEER and baselines are modest, STEER's per-seed advantage is directionally consistent, as it matches or exceeds the baseline on the large majority of seeds across three tasks. This suggests the improvement is not simply noise. Altogether, STEER shows gains without the use of an external verifier, reward model, or post-hoc filtering in a challenging robotic simulation dataset.

Remarks. These experiments address **Q3** by showing that replica exchange can improve continuous-control performance when successful behaviors form sparse, coherent trajectory modes. STEER delivers meaningful gains in downstream task performance, both when paired with an external verifier and when applied directly without access to one.

4 Conclusion and Future Work

We introduced STEER, an inference-time method that improves sample diversity in low-temperature diffusion sampling via replica exchange: a population of chains at different temperatures periodically swap particles, propagating discoveries from hot, exploratory replicas into the reported low-temperature chain. The swap rule adds no extra network evaluations, can compose with an external verifier, and provably preserves detailed balance under an idealized Gaussian setting. Across synthetic, image, protein, and robot-learning domains, STEER consistently improves the fidelity–diversity trade-off over alternative low-temperature sampling methods. STEER works particularly well in high-dimensional spaces, where the combinatorics of building blocks create an overwhelmingly large design space, yet biologically feasible solutions remain sparse. This is particularly relevant in biological and physical sciences. In these settings, structures are also governed by genuine physical energy landscapes, not merely probabilistic ones. Parallel tempering was originally developed for exactly this setting, making it a natural fit for integration with diffusion models, as we do in this work. A particular domain where STEER could show strong results in future work is molecular and materials discovery (Sanchez-Lengeling & Aspuru-Guzik, 2018; Gromski et al., 2019). While verifiers exist in these domains, they remain computationally expensive to use. Future work could focus on implementing a broader temperature ladder towards discovering new molecules and materials, with applications in cancer research, carbon capture, and more.

AI USE STATEMENT

In this work, we used generative AI tools to aid in code for generating synthetic data sets and aid or polish writing. We have not used generative AI tools help develop theoretical models or conceptual frameworks, proving mathematical claims, formulate mathematical claims, qualitative and thematic data analysis, implement methods, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, clean and reformat dataset, and interpret results; translation is not applicable to this work. Additionally, we used generative AI tools to create or modify scientific figures or images, create or edit software code, and edit a research paper to improve readability. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human subjects or crowdsourced data collection. All datasets and pre-trained models used in our experiments (Stable Diffusion 3, FoldingDiff, the D4RL/Gymnasium-Robotics PointMaze environment, and Robomimic) are publicly available and used in accordance with their respective licenses and terms of use. We are not aware of ethical concerns specific to this work beyond those generally associated with generative modeling and image synthesis, which are not particular to the sampling method proposed here.

REPRODUCIBILITY STATEMENT

Implementation details sufficient to reproduce our experiments are provided throughout the paper: training and evaluation protocols for the toy Gaussian mixture appear in Appendix C.1; hyperparameters for the Stable Diffusion 3 and FoldingDiff experiments are given in Appendix C.2 and Appendix C.3. PointMaze experimental details are in Appendix C.4 and the Robomimic policy architecture and optimization settings are described in full in Appendix C.5. The full proofs supporting our theoretical claims (Theorem B.1 and Proposition 2.1) are given in Appendices B.3 and B.4. The full source code, configuration files, and scripts required to reproduce the experiments will be released publicly upon acceptance.

REFERENCES

David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169, 1985. 1

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making?, 2023. URL https://arxiv.org/abs/2211.15657. 1, 2, 3, 5

Tara Akhound-Sadegh, Jungyoon Lee, Avishek Joey Bose, Valentin De Bortoli, Arnaud Doucet, Michael M. Bronstein, Dominique Beaini, Siamak Ravanbakhsh, Kirill Neklyudov, and Alexander Tong. Progressive inference-time annealing of diffusion models for sampling from boltzmann densities, 2025. URL https://arxiv.org/abs/2506.16471. 15

Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Roni Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *International Conference on Learning Representations*, volume 2024, pp. 51304–51323, 2024. 1

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning, 2024. URL https://arxiv.org/abs/2305.13301. 15

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 1

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025. 1

Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022. 1

Kevin Clark, Paul Vicol, Kevin Swersky, and David J Fleet. Directly fine-tuning diffusion models on differentiable rewards, 2024. URL https://arxiv.org/abs/2309.17400. 15

Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 45(9): 10850–10869, 2023. 1

Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2022. 8, 21

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 1, 21

Zehao Dou and Yang Song. Diffusion posterior sampling for linear inverse problem solving: A filtering perspective. In *International Conference on Learning Representations*, volume 2024, pp. 12069–12095, 2024. 15

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 *International conference on machine learning*, pp. 8489–8510. PMLR, 2023. 3, 5, 18, 20

David J Earl and Michael W Deem. Parallel tempering: Theory, applications, and new perspectives. *Physical Chemistry Chemical Physics*, 7(23):3910–3916, 2005. 3

Ezra Erives, Bowen Jing, Peter Holderrieth, and Tommi Jaakkola. Continuously tempered diffusion samplers. *arXiv preprint arXiv:2509.00316*, 2025. 5, 15

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024. 5, 19

Felix Faltings, Hannes Stark, Tommi Jaakkola, and Regina Barzilay. Protein fid: Improved evaluation of protein structure generative models. *Bioinformatics*, 42(4):btag156, 2026. 8, 19

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020. 5

Tomas Geffner, Kieran Didi, Zuobai Zhang, Danny Reidenbach, Zhonglin Cao, Jason Yim, Mario Geiger, Christian Dallago, Emine Kucukbenli, Arash Vahdat, et al. Proteina: Scaling flow-based protein structure generative models. In *International Conference on Learning Representations*, volume 2025, pp. 98803–98851, 2025. 1, 2, 5

Piotr S Gromski, Alon B Henson, Jarosław M Granda, and Leroy Cronin. How to explore chemical space using algorithms and automation. *Nature Reviews Chemistry*, 3(2):119–128, 2019. 9

Zhiye Guo, Jian Liu, Yanli Wang, Mengrui Chen, Duolin Wang, Dong Xu, and Jianlin Cheng. Diffusion models in bioinformatics and computational biology. *Nature reviews bioengineering*, 2 (2):136–154, 2024. 1

Mohsin Hasan, Viktor Ohanesian, Artem Gazizov, Yoshua Bengio, Alán Aspuru-Guzik, Roberto Bondesan, Marta Skreta, and Kirill Neklyudov. Discrete feynman-kac correctors, 2026. URL https://arxiv.org/abs/2601.10403. 15

W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 1970. 16

Jiajun He, José Miguel Hernández-Lobato, Yuanqi Du, and Francisco Vargas. Rne: plug-and-play diffusion inference-time control and energy-based training, 2026a. URL https://arxiv.org/abs/2506.05668. 15

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 *International Conference on Learning Representations*, 2026b. 2, 5, 15

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2, 18

Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022. 1, 20

Sunwoo Kim, Minkyu Kim, and Dongmin Park. Test-time alignment of diffusion models without reward over-optimization. In *International Conference on Learning Representations*, volume 2025, pp. 87238–87272, 2025. 15

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32, 2019. 23

Cheuk Kit Lee, Paul Jeha, Jes Frellsen, Pietro Lio, Michael Samuel Albergo, and Francisco Vargas. Debiasing guidance for discrete diffusion with sequential monte carlo, 2025. URL https://arxiv.org/abs/2502.06079. 15

Peizhuo Li, Emre Aksan, Alexandru-Eugen Ichim, Thabo Beeler, and Olga Sorkine-Hornung. Diversify diffusion with temperature sampling and variance-corrective time shifting. *arXiv preprint arXiv:2607.10853*, 2026. 1, 3, 5

Xiner Li, Yulai Zhao, Chenyu Wang, Gabriele Scalia, Gokcen Eraslan, Surag Nair, Tommaso Biancalani, Shuiwang Ji, Aviv Regev, Sergey Levine, and Masatoshi Uehara. Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding, 2024. URL https://arxiv.org/abs/2408.08252. 15

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014. 7, 19

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2023. 1

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. *arXiv preprint arXiv:2108.03298*, 2021. 5, 9, 20

Błażej Miasojedow, Eric Moulines, and Matti Vihola. An adaptive parallel tempering algorithm. *Journal of Computational and Graphical Statistics*, 22(3):649–664, 2013. 7

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. In *International Conference on Learning Representations*, volume 2025, pp. 94937–94982, 2025. 20

Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation, 2022. URL https://arxiv.org/abs/2104.11222. 19

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020. 19

Severi Rissanen, RuiKang OuYang, Jiajun He, Wenlin Chen, Markus Heinonen, Arno Solin, and José Miguel Hernández-Lobato. Progressive tempering sampler with diffusion. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 51724–51746, 2025. 15

Gareth O Roberts and Jeffrey S Rosenthal. Optimal scaling of discrete approximations to langevin diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(1): 255–268, 1998. 5

Gareth O Roberts and Jeffrey S Rosenthal. General state space markov chains and mcmc algorithms. *Probability Surveys*, 2004. 17

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011. 1

Benjamin Sanchez-Lengeling and Alán Aspuru-Guzik. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361(6400):360–365, 2018. 9

Andy Shih, Dorsa Sadigh, and Stefano Ermon. Long horizon temperature scaling. In *International conference on machine learning*, pp. 31422–31434. PMLR, 2023. 3

Raghav Singhal, Zachary Horvitz, Ryan Teehan, Mengye Ren, Zhou Yu, Kathleen McKeown, and Rajesh Ranganath. A general framework for inference-time scaling and steering of diffusion models, 2025. URL https://arxiv.org/abs/2501.06848. 1, 15

Anders Sjöberg, Jakob Lindqvist, Magnus Önnheim, Mats Jirstrand, and Lennart Svensson. Mcmc correction of score-based diffusion models for model composition. *Entropy*, 28(3):351, 2026. 4

Marta Skreta, Tara Akhound-Sadegh, Viktor Ohanesian, Roberto Bondesan, Alán Aspuru-Guzik, Arnaud Doucet, Rob Brekelmans, Alexander Tong, and Kirill Neklyudov. Feynman-kac correctors in diffusion: Annealing, guidance, and product of experts. *arXiv preprint arXiv:2503.02819*, 2025. 15

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. pmlr, 2015. 2

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021. URL https://arxiv.org/abs/2011.13456. 18

Masatoshi Uehara, Yulai Zhao, Chenyu Wang, Xiner Li, Aviv Regev, Sergey Levine, and Tommaso Biancalani. Inference-time alignment in diffusion models with reward-guided generation: Tutorial and review, 2025. URL https://arxiv.org/abs/2501.09685. 15

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8228–8238, 2024. 15

Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023. 1

Kevin E Wu, Kevin K Yang, Rianne Van den Berg, Sarah Alamdari, James Y Zou, Alex X Lu, and Ava P Amini. Protein structure generation via folding diffusion. *Nature communications*, 15(1): 1059, 2024. 5, 8, 19

Luhuan Wu, Brian Trippe, Christian Naesseth, David Blei, and John P Cunningham. Practical and asymptotically exact conditional sampling in diffusion models. *Advances in Neural Information Processing Systems*, 36:31372–31403, 2023. 15

Luhuan Wu, Yi Han, Christian Andersson Naesseth, and John Cunningham. Reverse diffusion sequential monte carlo samplers. *Advances in Neural Information Processing Systems*, 38:69445–69459, 2026. 15

Yanbo Xu, Yu Wu, Sungjae Park, Zhizhuo Zhou, and Shubham Tulsiani. Temporal score rescaling for temperature sampling in diffusion and flow models. *arXiv preprint arXiv:2510.01184*, 2025. 1, 2, 3, 5, 7, 16

Brian Yang, Huangyuan Su, Nikolaos Gkanatsios, Tsung-Wei Ke, Ayush Jain, Jeff Schneider, and Katerina Fragkiadaki. Diffusion-es: Gradient-free planning with diffusion for autonomous driving and zero-shot instruction following. *arXiv preprint arXiv:2402.06559*, 2024. 1

Jason Yim, Brian L Trippe, Valentin De Bortoli, Emile Mathieu, Arnaud Doucet, Regina Barzilay, and Tommi Jaakkola. Se (3) diffusion model with application to protein backbone generation. *arXiv preprint arXiv:2302.02277*, 2023. 1, 2, 5

Xiangcheng Zhang, Haowei Lin, Haotian Ye, James Y Zou, Jianzhu Ma, Yitao Liang, and Yilun Du. Inference-time scaling of diffusion models through classical search. In *International Conference on Learning Representations*, volume 2026, pp. 39133–39166, 2026. 1, 8, 21

APPENDIX

In this appendix, we provide additional experimental, architectural, and theoretical details to complement the main text. Appendix A gives a more extensive review of the diffusion-tempering literature, organized along the four axes introduced there: SMC-based correction, replica exchange, training-time tempering, and reward-guided alignment. Appendix B provides full proofs of our theoretical results from Section 2.4, including the closed-form score-rescaling coefficient under an isotropic Gaussian, detailed balance of the swap kernel, and the non-amplification of tempering bias. Appendix C provides additional experimental details for each setting in the main text: the toy Gaussian mixtures (Appendix C.1), Stable Diffusion 3 (Appendix C.2), FoldingDiff protein generation (Appendix C.3), PointMaze (Appendix C.4), and Robomimic (Appendix C.5). Appendix D provides additional results and ablations: Appendix D.1 covers replica-count and λ_M ablations, high-temperature sampling, and a reward-tilted target, while Appendix D.2 covers how adapting number of replicas to FoldingDiffusion shows strong results and Appendix D.3 covers Stable Diffusion swap-acceptance diagnostics and precision–recall results.

A RELATED WORKS

Temperature sampling for diffusion models has been approached in two primary ways: correcting a pretrained model at inference time, whether via sequential Monte Carlo (SMC) or replica exchange; and baking tempering into training itself, whether directly or through reward-guided alignment.

Sequential Monte Carlo for diffusion sampling. SMC repurposes a pretrained diffusion model without retraining: the method guides model outputs with importance weights so samples draw from a corrected target. This idea underlies conditional generation (Wu et al., 2023), unbiased normalizing-constant estimation (Wu et al., 2026), inference-time temperature annealing via a Feynman–Kac scheme (Skreta et al., 2025), and related applications (Dou & Song, 2024; Singhal et al., 2025; Lee et al., 2025; He et al., 2026a; Hasan et al., 2026). However, SMC holds a high memory cost from holding and reweighting the whole particle set at every denoising step (He et al., 2026b). With STEER, on the other hand, cross-particle interaction only occurs between adjacent chains. SMC methods also typically filter with an external verifier, whereas STEER works with or without one.

Existing work on Controlling Diffusion with Replica Exchange (CREPE) (He et al., 2026b) adapts SMC by flipping its axis: CREPE indexes by diffusion timestep instead of by temperature. However, CREPE's burn-in recurs each time a new, roughly independent sample is needed from a running chain, since drawing one requires traversing much of the temperature ladder again, a mixing-time cost that a per-particle NFE comparison does not capture. STEER pays no equivalent cost: each replica's single trajectory is already a valid, unbiased draw, so generating a whole temperature-diverse batch is parallel with no mixing time between samples. Unlike reward-based methods, STEER also needs neither an external reward model nor a particle population. Additionally, CREPE has not been tested on the large, general-purpose pretrained models we target here.

Tempering along two other axes. Other existing tempering methods require new training or require an external objective. One bakes tempering into training: Erives et al. (2025) widen the training distribution across temperatures, while Rissanen et al. (2025) and Akhound-Sadegh et al. (2025) train a sequence of models one temperature at a time, each bootstrapped from the last. All three report strong Boltzmann-sampling results, but all require training a dedicated model for each density rather than adapting one pretrained model. The other line targets an external reward instead of a temperature: fine-tuning (Black et al., 2024; Clark et al., 2024; Wallace et al., 2024) optimizes the reward well but risks over-optimizing away diversity, while training-free guidance (Uehara et al., 2025; Li et al., 2024) avoids that risk but under-optimizes the reward instead. Kim et al. (2025) sits between these failure modes, using SMC-based tempering over reward-tilted targets to match fine-tuning-level reward without losing diversity, but it still inherits SMC's costs and still targets an external reward rather than the model's own temperature-sharpened distribution.

B Additional Proofs and Theoretical Results

B.1 Score Rescaling Under an Isotropic Gaussian

Section 2.2 showed that constant score rescaling cannot recover $\tilde{p}$ exactly for a general data distribution. In the following, we show that a time-varying score rescaling rule can recover correct score for an isotropic Gaussian distribution:

Proposition B.1 (Gaussian score rescaling, adapted from Xu et al. (2025)). *Let $x_0 \sim \mathcal{N}(0, \mathbf{I})$, and let $\tilde{p}(x_0) \equiv \mathcal{N}(0, \lambda \mathbf{I})$ denote the λ-scaled target. Then for every t and every $x_t \in \mathbb{R}^D$,*

$$\nabla_{x_t} \log \tilde{p}(x_t) = r_t(\lambda)\, \nabla_{x_t} \log p(x_t), \qquad r_t(\lambda) := \frac{\alpha_t^2 + \sigma_t^2}{\alpha_t^2 \lambda + \sigma_t^2}.$$

Under a variance-preserving schedule ($\alpha_t^2 + \sigma_t^2 = 1$), this simplifies to $r_t(\lambda) = 1/(\alpha_t^2 \lambda + \sigma_t^2)$, the form used in the main text.

Proof. Consider the tempered target $\tilde{p}(x_0) \propto p(x_0)^{1/\lambda}$. Raising an isotropic Gaussian to a power keeps it in the same family, only rescaling its variance: $\tilde{p}(x_0) = \mathcal{N}(0, \lambda \mathbf{I})$. Because $\tilde{p}_0$ is itself Gaussian, diffusing it forward just adds independent Gaussian noise to an already-Gaussian variable, so the tempered marginal stays in closed form at *every* t, not only at $t = 0$:

$$p(x_t) = \mathcal{N}\big(x_t; 0, (\alpha_t^2 + \sigma_t^2)\mathbf{I}\big), \qquad \tilde{p}(x_t) = \mathcal{N}\big(x_t; 0, (\alpha_t^2 \lambda + \sigma_t^2)\mathbf{I}\big). \tag{5}$$

Since $p(x_t)$ and $\tilde{p}(x_t)$ share the same mean, their scores point in exactly the same direction, $-x_t$, differing only in magnitude:

$$\nabla \log p(x_t) = -\frac{x_t}{\alpha_t^2 + \sigma_t^2}, \qquad \nabla \log \tilde{p}(x_t) = -\frac{x_t}{\alpha_t^2 \lambda + \sigma_t^2}.$$

Their ratio is therefore a pure scalar, with no leftover dependence on x_t:

$$\nabla_{x_t} \log \tilde{p}(x_t) = r_t(\lambda)\, \nabla_{x_t} \log p(x_t), \qquad r_t(\lambda) := \frac{\alpha_t^2 + \sigma_t^2}{\alpha_t^2 \lambda + \sigma_t^2}. \tag{6}$$

This coefficient recovers $r_1(\lambda) = 1$ at pure noise, using the standard boundary condition $\alpha_1 = 0$ (so the λ-dependent term vanishes from both numerator and denominator), and $r_0(\lambda) = 1/\lambda$ at the clean data distribution, using $\alpha_0 = 1, \sigma_0 = 0$ — both boundary conditions holding for any admissible noise schedule, independent of whether the schedule is variance-preserving throughout. □

B.2 Detailed Balance of the Swap Kernel

We exchange configurations between chains only when doing so respects the target distribution. This holds for any swap between two components with arbitrary densities, not only diffusion marginals; the specific case used in this paper follows as a substitution at the end. The key condition is *detailed balance*, where the rate of swapping from (x, x') to (x', x) must equal the rate of the reverse. Because a swap and its reverse are proposed with equal probability — swapping back is the same move as swapping forward — the usual Metropolis–Hastings correction for asymmetric proposals drops out. This leaves an acceptance rule that only depends on joint probabilities before and after the exchange, precisely the standard Metropolis–Hastings construction (Hastings, 1970).

Proposition B.2 (Detailed balance of a general swap kernel). *Take any two densities π_1, π_2, and let $\pi(x, x') := \pi_1(x)\, \pi_2(x')$ be their product before a swap is proposed. Writing $r := \pi(x', x)/\pi(x, x')$, define the swap-acceptance probabilities $a_{(x,x')} := \min(1, r)$ and $a_{(x',x)} := \min(1, 1/r)$: a swap that increases the joint density is thus always accepted, and one decreasing it is accepted in proportion to how much worse it would be. Under this rule, π is invariant: the swap kernel leaves it unchanged.*

Proof. **Case $r \leq 1$.** Then $a_{(x,x')} = r$ and $a_{(x',x)} = 1$, so

$$\pi(x, x') \cdot r = \pi(x, x') \cdot \frac{\pi(x', x)}{\pi(x, x')} = \pi(x', x) = \pi(x', x) \cdot 1.$$

Case $r > 1$. Then $a_{(x,x')} = 1$ and $a_{(x',x)} = 1/r$, so

$$\pi(x, x') \cdot 1 = \pi(x', x) \cdot \frac{\pi(x, x')}{\pi(x', x)} = \pi(x', x) \cdot \frac{1}{r}.$$

Both cases give

$$\pi(x, x')\, a_{(x,x')} = \pi(x', x)\, a_{(x',x)}, \tag{7}$$

so π is stationary under the swap kernel. □

Recovering our setting is a matter of substitution: taking $\pi_1 = p_t^{(k)}$, $\pi_2 = p_t^{(k')}$ for two chains at diffusion time t, and writing each chain's marginal in Boltzmann form relative to the base ($\lambda = 1$) chain, $p_t^{(k)}(x) \propto \exp(-r_t(\lambda_k)E(x))$ with $E(x) := -\log p_t^{(1)}(x)$, gives

$$r = \exp\big[(E(x) - E(x'))\big(r_t(\lambda_k) - r_t(\lambda_{k'})\big)\big], \tag{8}$$

exactly the acceptance rule of eq. 4. A configuration unlikely under its current temperature is thus more likely to be swapped into a more compatible chain, without disturbing the joint target.

B.3 Exact Detailed Balance under Isotropic Gaussian Data

Appendix B.1 derived $r_t(\lambda)$ in closed form for an isotropic Gaussian target; we now use that same closed form to check the practical swap kernel against the general detailed-balance result of Proposition B.2. In general, our two-endpoint estimator only approximates the true energy difference between swap candidates (Section 2.3), so the resulting acceptance probability only approximately satisfies detailed balance. For isotropic Gaussian data, however, the true score is exactly affine, so the two-endpoint estimate recovers the exact energy difference rather than approximating it — and consequently, the practical swap kernel achieves exact, not merely approximate, detailed balance.

Theorem B.1 (Detailed balance of the practical swap kernel under isotropic Gaussian data)**.** *Let $x_0 \sim \mathcal{N}(0, \mathbf{I})$, and consider two chains i, j tempered by λ_i, λ_j. The swap-acceptance rule of eq. 4 coincides exactly with the true Metropolis–Hastings acceptance probability for these chains' joint distribution, and therefore leaves it invariant.*

Proof. By Proposition B.1, the practical rule of eq. 4 is, in this case, literally

$$a_{ij} = \min\big(1,\ \exp[(E_i - E_j)(r_t(\lambda_i) - r_t(\lambda_j))]\big),$$

i.e. it uses the *true* energy difference, not an estimate of it. Writing each chain's marginal in Boltzmann form, $p_t^{(k)}(x) \propto \exp(-r_t(\lambda_k)E(x))$ (as in eq. 8), this is exactly the acceptance rule of Proposition B.2 applied to $\pi_1 = p_t^{(i)}$, $\pi_2 = p_t^{(j)}$. Proposition B.2 already establishes, for arbitrary densities π_1, π_2 with no Gaussian assumption, that this rule satisfies $\pi(x, x')\, a(x, x') = \pi(x', x)\, a(x', x)$ for $\pi(x, x') = \pi_1(x)\pi_2(x')$. Taking $\pi_1 = p_t^{(i)}$ and $\pi_2 = p_t^{(j)}$ gives detailed balance for the practical swap kernel exactly.

□

B.4 Swap Step Does Not Amplify Tempering Bias

Intuitively, the swap step can only redistribute whatever error already exists between chains; it has no mechanism for inventing new error. This follows from the swap kernel being built around the exact acceptance rule of Proposition B.2, which leaves the true joint target invariant: the kernel can move mass toward that target but never away from it. Thus the gap to the true target can shrink or hold steady under repeated swaps, but never grow. We formalize this below.

Proposition B.3 (Swaps do not amplify tempering bias, adapted from Roberts & Rosenthal (2004))**.** *Consider a swap-eligible pair of chains i, j, and suppose their joint distribution μ is within $\varepsilon(\lambda)$ of the true joint target $\pi_t^{(i,j)}$ in total variation. Since our swap kernel K_{swap} is built from the exact acceptance rule of eq. 4 and admits $\pi_t^{(i,j)}$ as a stationary point, applying it*

can only shrink the distance to $\pi_t^{(i,j)}$, never grow it. Thus, after any number of repeated swap attempts, the joint distribution stays within that same $\varepsilon(\lambda)$ of the true target.

Proof. Let us define μ as our current joint distribution of chain i and j. Write $\rho := \mu - \pi_t^{(i,j)}$ (the *error*), so $\|\rho\|_{TV} = \varepsilon(\lambda)$ by assumption. Let us split probability measure ρ into a positive and negative error, $\rho = \rho^+ - \rho^-$ (the *excess* and *deficit*), each carrying equal total mass $\|\rho\|_{TV}$. Intuitively, ρ^+ is the excess probability μ places above $\pi_t^{(i,j)}$, and ρ^- is the deficit.

This positive and negative error must balance since both μ and $\pi_t^{(i,j)}$ integrate to 1 (being a probability distribution). Applying our swap kernel K_{swap} to error ρ can only redistribute this excess and deficit across states — since every transition probability (the *move likelihood*) is between 0 and 1, no application of the kernel can manufacture new discrepancy beyond what ρ^+ and ρ^- already contain. Formally, for a plausible set of chains A,

$$(\mu K_{\text{swap}})(A) - \big(\pi_t^{(i,j)} K_{\text{swap}}\big)(A) = \int K_{\text{swap}}(x, A)\, \rho(dx) \tag{9}$$

$$\in \big[-\varepsilon(\lambda),\ \varepsilon(\lambda)\big], \tag{10}$$

and since $\pi_t^{(i,j)}$ is a fixed point (the *equilibrium*) of K_{swap} (Theorem B.1), the left side is exactly $(\mu K_{\text{swap}})(A) - \pi_t^{(i,j)}(A)$. Taking the supremum over A gives $\|\mu K_{\text{swap}} - \pi_t^{(i,j)}\|_{TV} \le \varepsilon(\lambda)$, as claimed. □

B.5 Effective Step-size Scaling

Temperature control in diffusion sampling is commonly implemented by rescaling either the score or the injected noise. While STEER adopts score rescaling, we show that these two mechanisms yield identical single-step updates under the unadjusted Langevin (ULA) formulation, provided that the step size and rescaling coefficients are matched appropriately.

Proposition B.4 (Score and noise rescaling are equivalent up to a step-size reparameterization)**.** *Under the ULA interpretation of diffusion sampling, one step of TSR at temperature λ and step size β_t is identical to one step of CNS at coefficient $\mu = 1/r_t(\lambda)$ and step size $\beta'_t = \beta_t\, r_t(\lambda)$. At $t = 0$, where $r_0(\lambda) = 1/\lambda$ (eq. 1), this specializes to: TSR at λ, step size β_0, equals CNS at the* same *λ, step size β_0/λ.*

Proof. Diffusion sampling can be read as an unadjusted Langevin sampler targeting p_t at each noise level (Song et al., 2021; Du et al., 2023): at step size β_t,

$$x_{t+1} = x_t + \beta_t\, s_\theta(x_t, t) + \sqrt{2\beta_t}\,\xi_t, \qquad \xi_t \sim \mathcal{N}(0, I). \tag{11}$$

Scaling the injected noise's variance by λ, and leaving the drift untouched, gives

$$x_{t+1} = x_t + \beta_t\, s_\theta(x_t, t) + \sqrt{2\beta_t\lambda}\,\xi_t. \tag{12}$$

Scaling the score by $r_t(\lambda)$ instead, and leaving the noise term as in eq. 11, gives

$$x_{t+1} = x_t + \big(\beta_t\, r_t(\lambda)\big)\, s_\theta(x_t, t) + \sqrt{2\beta_t}\,\xi_t. \tag{13}$$

Writing $\beta'_t := \beta_t\, r_t(\lambda)$, eq. 13 is exactly eq. 12's form – a step size β'_t applied to the drift, with the noise term $\sqrt{2\beta_t} = \sqrt{2\beta'_t/r_t(\lambda)}$ – evaluated at CNS's own coefficient $\mu = 1/r_t(\lambda)$. TSR and CNS are therefore the same discretized ULA step, differing only in which step size, β_t or $\beta'_t = \beta_t\, r_t(\lambda)$, each one calls its own. □

C Experimental Details

C.1 Toy Experiments.

We construct two synthetic 2D benchmarks. Benchmark 1 is a 20-mode Gaussian mixture: 19 easy modes (std 0.1) in a 5×5 square, plus one isolated hard mode ($1.5\times$ the spread away, std 15% as large, 3.5% of the mass, $30.6\times$ the peak density). We train based on samples from the untempered ground-truth distribution. We use a denoising diffusion probabilistic model (DDPM) (Ho et al.,

2020) with the standard ϵ-parameterization. $\epsilon_\theta(x, t)$ embeds $x \in \mathbb{R}^2$ to $\mathbb{R}^{128}$, adds a projected 64-d sinusoidal timestep embedding, applies four residual blocks each consisting of a single linear layer and activation, and projects back to $\mathbb{R}^2$. We train and sample with $T = 100$ timesteps with a cosine variance schedule ($s = 0.008$), clipping β_t to $[10^{-5}, 0.999]$ for numerical stability, optimized with Adam (2×10^{-4} learning rate, batch size 256, 10,000 updates); each model is trained once and reused across all sampling methods.

We sample from the true tempered target $p(x)^{1/\lambda}$ using fine grid sampling: we evaluate the mixture's log-density on an 800×800 grid spanning each component's support with generous padding, raise it to the tempering power pointwise (the tempered density does not decompose into per-component tempering, so this cannot be done analytically component-by-component), and draw samples via categorical (inverse-CDF) sampling over grid cells with sub-cell jitter added for continuity.

We report three metrics against the tempered reference $p(x)^{1/\lambda}$, over 5 seeds and 1000 samples/seed: 2-Wasserstein (W_2), mode-weight TV, and hard-mode coverage (Cov). We compute TV by assigning both the generated samples and a large batch of true tempered-target samples to their nearest modes, then taking half the distance between the two resulting occupancy distributions. Cov counts, across the pooled 5,000 samples, how many land within a radius of 0.5 of the hard mode's mean, reported as a percentage of the count its true weight would predict (175 samples, i.e. 3.5% of 5,000); 100% therefore indicates perfect calibration, with values above or below 100% indicating over- or under-discovery respectively. The radius of 0.5 is far larger than the hard mode's std (0.015) yet remains well inside its separation from the surrounding easy-mode cluster, so every counted point is unambiguously attributable to the hard mode itself.

Benchmark 2 tests weight preservation rather than discovery: 40 equal-weight, equal-variance Gaussians in a 15×15 square with a 6σ minimum separation, whose tempered target is exactly uniform for any λ. We report the same W_2 and TV metrics, plus normalized entropy, over 5 seeds and 1000 samples/seed. Entropy is computed from the generated samples' occupancy distribution alone, measuring how evenly spread the samples are across all modes.

C.2 Stable Diffusion 3 Experimental Setup.

We evaluate Stable Diffusion 3-medium (Esser et al., 2024) on 10,000 COCO prompt–image pairs (Lin et al., 2014), computing FID against real val2014 images with clean-fid (Parmar et al., 2022) and CLIPScore with ViT-B/32 (Radford et al., 2021), both capped to a common sample count across configurations. We sample with 30 steps, guidance scale 7.5, empty negative prompt, and seed 42. Since SD3 is flow-matching rather than DDPM, we bridge the two by defining $\bar{\alpha}_t := 1 - \sigma_t$ from the scheduler's sigma schedule and converting the CFG-combined velocity into a pseudo-noise term $\hat{\epsilon}_t = x_t + \bar{\alpha}_t v_\theta(x_t, t)$; swap and score-rescaling apply to $\hat{\epsilon}_t$ as in the DDPM setting, then invert back to a velocity for the scheduler's Euler step. Each of $n_{\text{particles}}$ replicas is one full image, with swaps acting on the flattened latent. We sweep $n_{\text{particles}} \in \{2, 4, 6, 8\}$ (STEER) against single-chain TSR, and $\lambda_{\text{start}} \in \{0.95, 0.98, 1.02, 1.05\}$ with $\lambda_{\text{end}} = \lambda_{\text{start}} + 0.2$. Each TSR sample reuses the exact initial latent that tracked λ_{start} in its matched STEER run, isolating the swap mechanism from initialization variance.

C.3 FoldingDiffusion Experimental Setup

We generate protein backbones using the pretrained FoldingDiff checkpoint (Wu et al., 2024), a diffusion model over per-residue backbone torsion and bond angles trained on CATH. We sample 10 backbones per length, uniformly spanning lengths 50–128 residues, using the model's default sampling configuration, a fixed random seed, and a single GPU.

For each sampled backbone, we evaluate self-consistency and distributional fidelity following the standard inverse-folding/refolding pipeline (Wu et al., 2024): ProteinMPNN generates a candidate amino-acid sequence for the backbone, OmegaFold predicts the resulting structure from that sequence, and we compute the TM-score between the predicted structure and the original sampled backbone as a measure of designability. We additionally compute ProtFID, a protein-specific Fréchet distance between generated and reference structures in a learned representation space (Faltings et al., 2026), as a measure of distributional fidelity beyond individual designability scores.

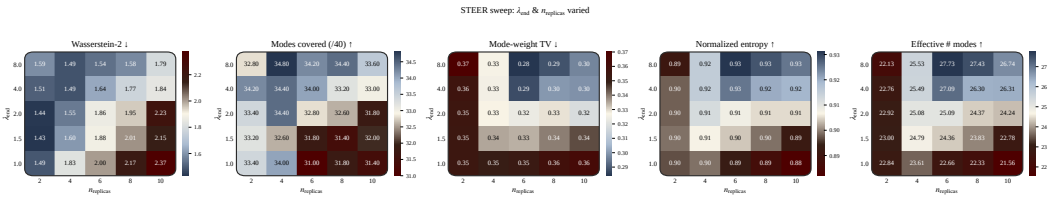


Figure 7: **Ablation over replica count and final temperature.** We sweep $n_{\text{replicas}} \in \{2, 4, 6, 8, 10\}$ and $\lambda_M \in \{1.0, 1.5, 2.0, 4.0, 8.0\}$ (with $\lambda_1 = 0.9$ fixed), reporting five metrics averaged over 5 seeds: Wasserstein-2 distance to real tempered samples (↓), number of modes covered (↑), normalized entropy of mode occupancy (↑), TV distance (↓), and effective number of modes (↑). Lower Wasserstein-2 and higher mode-coverage/entropy indicate samples that better match the target distribution's support and mass allocation.

C.4 PointMaze.

We plan on `PointMaze-giant-navigate-v0` (Park et al., 2025) using a pretrained diffuser-style (Janner et al., 2022) temporal U-Net planner (horizon 400, 256 diffusion steps, x_0-parameterized), with a wall-collision/goal-reaching verifier scoring candidate plans at inference time. For CREPE, we generate N independent ancestral samples and run odd/even replica-exchange iterations across all N chains in parallel, selecting the best-scoring plan via the verifier. STEER and TSR are compared against CREPE at matched particle counts $N \in \{2, 4, 8, 16\}$ over 5 seeds ($\{1, 50, 100, 150, 200\}$), alongside a BFS (resampling) tree-search baseline. We run 40 rollouts per seed as per the evaluation protocol of Du et al. (2023).

C.5 Robomimic.

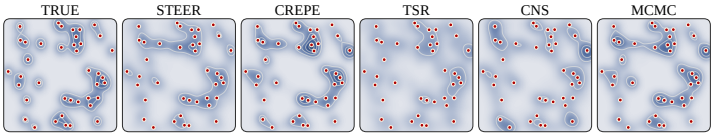
We evaluate on Robomimic (Mandlekar et al., 2021) for imitation learning, across four tasks—Lift, Can, Square, Tool Hang—training one diffusion policy per task (subsampled to 10% of demonstrations for Lift/Can, 75% for Square/Tool Hang) with an identical recipe across sampling methods. The policy is a 1D temporal U-Net (256/512/1024 channels, kernel 5, 8 groups, 256-d timestep embedding) with a 2-step observation history, predicting 16-step action chunks (8 executed per replan). Training uses the standard DDPM objective (100 steps, squared-cosine schedule, ϵ-prediction, clipping to $[-1, 1]$) for 2000 epochs, batch size 100, AdamW (10^{-4} lr, cosine decay, 500 warmup steps, 10^{-6} L_2), with EMA decay 0.75. At inference, all methods use the same 100-step schedule; STEER/TSR sweep $n_{\text{replicas}} \in \{1, 4, 6, 8\}$, $\lambda_{\text{start}} \in [0.2, 1.0]$ (1.0 = untempered baseline), λ_{end} up to 10 (linearly spacing). All methods use 50 rollouts $\times$ 5 seeds.

D Extended Results

D.1 Toy Experiments

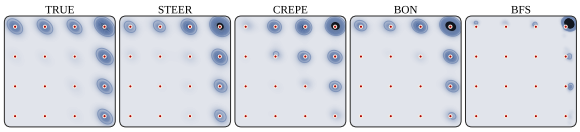
Ablations To understand how STEER's replica count and final inverse-temperature interact, we conduct a grid ablation over $n_{\text{replicas}} \in \{2, 4, 6, 8, 10\}$ and $\lambda_M \in \{1.0, 1.5, 2.0, 4.0, 8.0\}$, with the initial temperature fixed at $\lambda_1 = 0.9$ across all runs. For each of the 25 configurations, we draw 1000 samples under each of 5 random seeds and report metrics averaged across seeds. We evaluate sample quality and mode-coverage behavior along four axes: (i) Wasserstein-2 distance to real samples drawn from the same tempered reference distribution, computed on a 500-sample subset for tractability; (ii) the number of ground-truth modes covered, defined as a mode being covered if at least one generated sample falls within a fixed radius of its mean; (iii) the mode-weight TV distance; (iv) the normalized entropy of the empirical mode-occupancy distribution, which captures how evenly mass is spread across covered modes; and (v) the effective number of modes, the exponentiated entropy, which jointly reflects both breadth and balance of coverage. Together, these metrics let us see how increasing n_{replicas} or λ_M may create better-calibrated samples, broader mode coverage, or more balanced allocation of mass across modes.

High-Temperature Sampling As an additional toy benchmark, we test whether STEER can preserve fidelity in the opposite temperature regime: rather than sharpening samples toward the lowest-temperature chain, we task it with outputting the *highest*-temperature sample from a set of high-temperature chains. We follow the same training setup as 3.1. We set $\lambda_1 = 4.0$ and $\lambda_M = 8.0$ with 3 replicas, outputting λ_1. We compare against MCMC, CNS, and TSR under the same setup.



Method	W2 ↓	TV ↓	Ent. ↑	Eff. ↑
MCMC	1.42	0.19	0.97	32.38
CNS	1.76	0.21	0.95	27.56
TSR	1.67	0.16	0.98	33.02
CREPE	1.62	0.20	0.97	31.69
STEER	**1.35**	**0.13**	**0.98**	**35.20**

Figure 8: **STEER's low-temperature gains extend to high-temperature sampling.** Comparison of sampling methods when targeting *high*-temperatures ($\lambda_1 = 4.0$, $\lambda_M = 8.0$, 3 replicas) on the 40-mode toy dataset. Mean values across 5 seeds; full results with standard errors in Appendix D.



Method	W2 ↓	TV ↓	Entropy ↑	Eff. ↑
BFS	2.91 ± 0.014	0.87 ± 0.004	0.13 ± 0.006	1.15 ± 0.009
BoN	1.26 ± 0.024	0.57 ± 0.004	0.58 ± 0.005	3.37 ± 0.048
CREPE	1.18 ± 0.015	0.52 ± 0.004	0.68 ± 0.006	4.61 ± 0.078
STEER	$\mathbf{0.87 \pm 0.019}$	$\mathbf{0.50 \pm 0.002}$	$\mathbf{0.70 \pm 0.006}$	$\mathbf{4.99 \pm 0.135}$

Figure 9: **STEER aligns closest with reward-tilted target compared to other verifier methods.** Samples from STEER, CREPE, BON, and BFS against the resampled ground-truth reference (TRUE) for the reward-tilted target π_β of eq. 14. STEER's score-based acceptance rule (eq. 4) recovers the target distribution more faithfully than the baselines across all four metrics (right).

STEER outperforms all baselines across every metric, indicating that its benefits are symmetric: it is just as effective at flattening a distribution as it is at sharpening one.

Reward-Tilted Target Distribution The reward-tilted target distribution is defined as

$$\pi_\beta(x) \propto p(x) \cdot r(x)^\beta, \tag{14}$$

where $p(x)$ is the true data distribution, $r(x) \in [0, 1]$ is the reward function, and β controls the tilting strength: $\beta = 0$ recovers $p(x)$ exactly, and $\beta \to \infty$ collapses π_β onto $\arg\max_x r(x)$.

Since π_β has no closed form, it is approximated via self-normalized importance resampling. A pool $\{x_i\}_{i=1}^N$ is drawn from $p(x)$ (with $N = n \cdot \text{oversample}$), each point is assigned a weight

$$w_i = \frac{r(x_i)^\beta}{\sum_{j=1}^N r(x_j)^\beta}, \tag{15}$$

and n points are resampled with replacement according to $\{w_i\}_{i=1}^N$. This procedure converges to exact draws from π_β as $N \to \infty$; at finite N, fidelity degrades as β grows, since larger β concentrates weight on fewer pool points, shrinking the effective sample size. This resampled set serves as the ground-truth reference against which the diffusion-based methods (STEER, BoN, BFS (Zhang et al., 2026), classifier guidance (Dhariwal & Nichol, 2021)) are evaluated. Each method approximates π_β through the sampling trajectory rather than post-hoc reweighting. Strong results across all metrics are reported in Figure 9.

Auxiliary Replicas Figure 10 shows the remaining replicas for the 40-mode Gaussian benchmark, comparing STEER, TSR, and samples drawn directly from the true tempered mixture across the same linearly-spaced temperature ladder ($\lambda_1 = 0.9$ to $\lambda_M = 4.0$) used in the main text. Near $\lambda \approx 1$, both STEER and TSR closely reproduce the true tempered distribution's cluster structure, correctly recovering each mode's location and relative density. As λ increases into the sharpening regime ($\lambda = 2.97, 4.00$), however, both samplers' densities become visibly more diffuse and fail to reproduce the tight, well-localized clusters that the true target retains at every λ – indicating that sample fidelity degrades as the distribution is sharpened, even though STEER remains the more faithful of the two baselines shown here.

D.2 FoldingDiffusion

We assess designability with ProteinMPNN (Dauparas et al., 2022): we generate candidate sequences for each sampled backbone, refold them, and count a backbone as designable when the self-consistency TM-score exceeds 0.8. Figure **??** reports designability across seven length bins (50–128 residues), with the replica budget scaling from 4 to 10 with chain length. STEER matches or exceeds temperature rescaling in every bin and improves pooled designability from 35.9% to

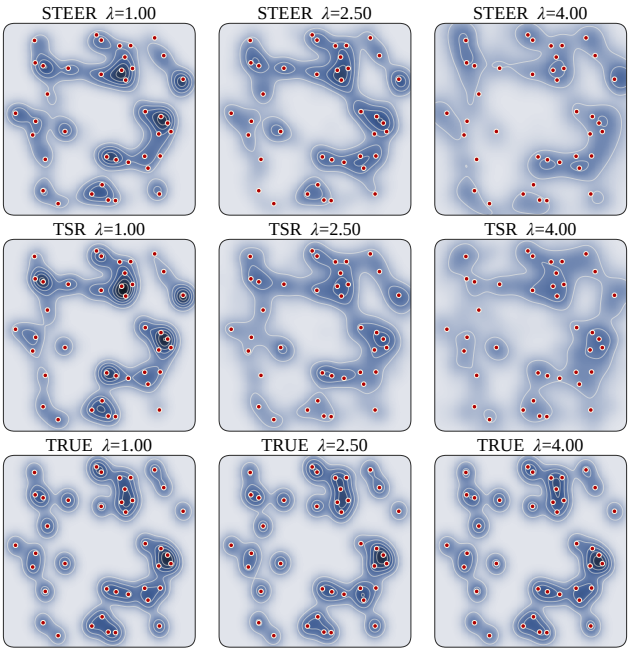


Figure 10: Additional replicas on the 40-mode benchmark, comparing STEER and TSR against samples from the true tempered mixture across a linearly-spaced λ ladder.

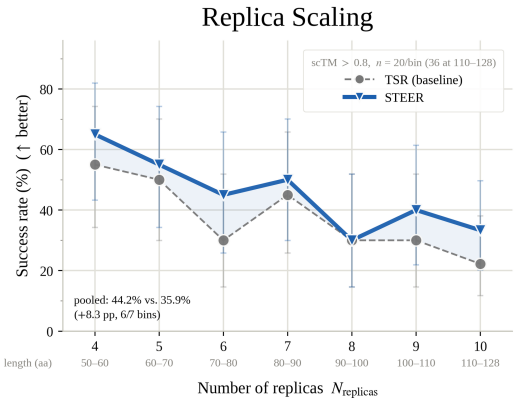


Figure 11: **STEER improves designability across all length bins.** Designability (scTM > 0.8), with N_{replicas} scaling from 4 to 10 with chain length. STEER matches or exceeds temperature rescaling in every bin. Error bars are 95% Wilson intervals. The decreasing trend reflects longer-backbone difficulty, not a cost of replicas.

44.2% (56/156 vs. 69/156). Since the replica budget is tied to length here, the downward trend reflects the difficulty of longer backbones, not a cost of additional replicas.

D.3 Stable Diffusion

Acceptance rates. Figure 12 shows swap-acceptance probability for our 6-replica DEO exchange scheme, with guidance scales $\lambda \in \{0.95020,\ 0.99463,\ 1.03906,\ 1.08398,\ 1.12793,\ 1.17285\}$ spaced linearly across the ladder. Swaps follow a deterministic even-odd schedule, and the x-axis uses SD3's actual logged flow-matching timestep ($T = 1000 \to t = 0$) rather than raw sweep index. The three lowest-λ bonds stay saturated near 1.0 throughout, while the outermost bond ($\lambda_4 = 1.12793, \lambda_5 = 1.17285$) collapses from 1.0 to ≈ 0 by mid-trajectory, and the adjacent bond degrades similarly near $t = 0$. The ladder-wide mean acceptance falls from 1.0 to ≈ 0.69, showing that linear λ-spacing under-resolves the high-guidance end of the ladder and that a denser schedule there would improve mixing. Additional image samples are provided below.

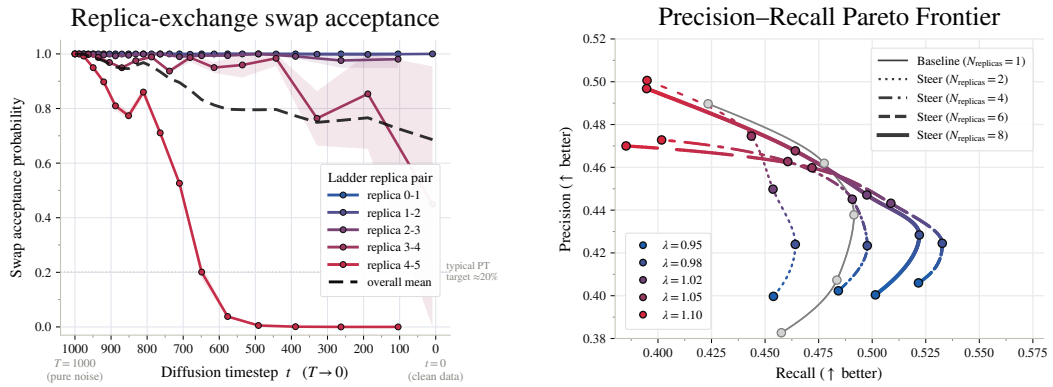
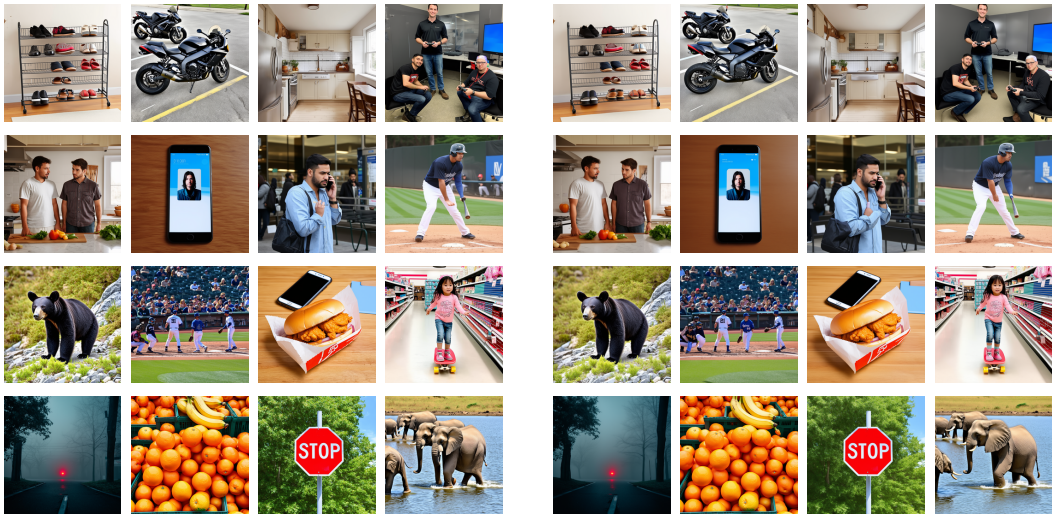


Figure 12: **Replica exchange shows greatest fidelity-diversity tradeoffs with 6 replicas.** *Left:* swap-acceptance probability over the sampling trajectory for a 6-replica temperature ladder. *Right:* precision–recall frontier across N_{replicas}; more replicas improve recall without hurting precision up to $N_{\text{replicas}} = 6$, with $N_{\text{replicas}} = 8$ adding nothing further.

Precision and recall. To complement the FID–CLIP frontier in Figure 4, we report precision and recall, a standard pair of metrics for disentangling sample *fidelity* from *diversity* in generative models (Kynkäänniemi et al., 2019). We follow the improved precision/recall recipe of Kynkäänniemi et al. (2019): both real and generated images are embedded with VGG-16, and precision (recall) is the fraction of generated (real) samples falling within the k-th-nearest-neighbor radius of some real (generated) sample, with $k = 3$.

We compute precision and recall for STEER against the $N_{\text{replicas}} = 1$ baseline on Stable Diffusion 3 sampling, sweeping the same guidance strength λ used in the main text and varying the number of replicas $N_{\text{replicas}} \in \{2, 4, 6, 8\}$. Figure 12 shows the resulting precision–recall frontier. The pattern mirrors the FID–CLIP trade-off reported in Figure 4: increasing N_{replicas} from 2 to 6 shifts the frontier outward. FID is sensitive to both the realism and the coverage of generated samples, so a mechanism that improves diversity without sacrificing fidelity should move both frontiers outward together, explaining why this pattern mirrors Figure 4. The same diminishing returns appear here for the same reason as in the main text: at a fixed temperature spread, more replicas shrink the gap between adjacent rungs, pushing swap acceptance toward saturation and leaving little additional exchange for extra replicas to contribute.



(a) STEER (b) TSR

Figure 13: Side-by-side comparison of *Steer* and *TSR* outputs on the same 16 COCO validation images.

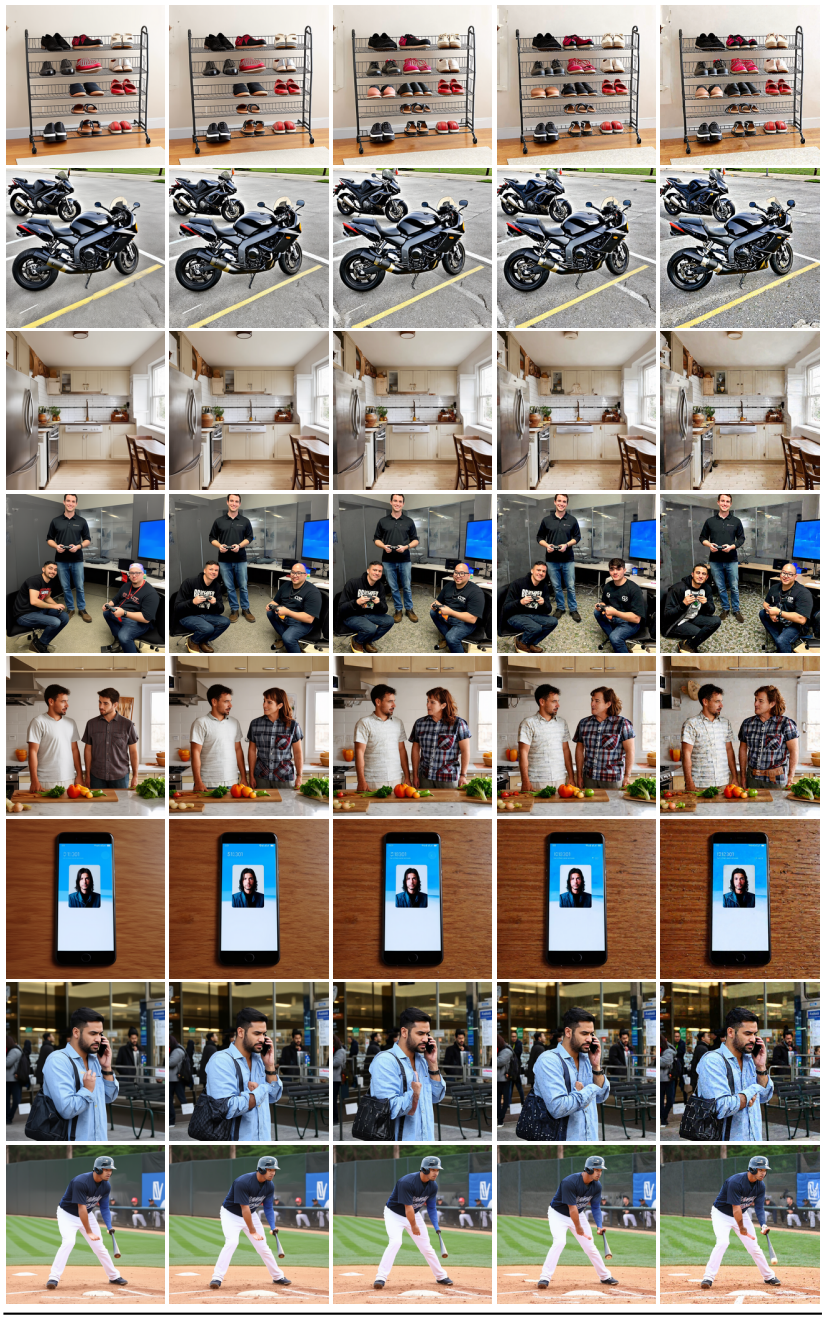
$\lambda = 0.950$	$\lambda = 0.980$	$\lambda = 1.020$	$\lambda = 1.050$	$\lambda = 1.100$
				

Table 1: Qualitative comparison of Stable Diffusion 3 samples across STEER temperature settings ($\lambda \in \{0.950, 0.980, 1.020, 1.050, 1.100\}$) on eight COCO val2014 prompts.