CMAD: COOPERATIVE MULTI-AGENT DIFFUSION VIA STOCHASTIC OPTIMAL CONTROL

Riccardo Barbano, Alexander Denker & Željko Kereta
Department of Computer Science
University College London
r.barbano@cs.ucl.ac.uk

Runchang Li
Department of Mathematics
The Chinese University of Hong Kong

Francisco Vargas
University of Cambridge & Xaira Technologies

ABSTRACT

Continuous-time generative models have achieved remarkable success in image restoration and synthesis. However, controlling the composition of multiple pretrained models remains an open challenge. Current approaches largely treat composition as an algebraic composition of probability densities, such as via products or mixtures of experts. This perspective assumes the target distribution is known explicitly, which is almost never the case. In this work, we propose a different paradigm that formulates compositional generation as a cooperative *Stochastic Optimal Control* problem. Rather than combining probability densities, we treat pre-trained diffusion models as interacting agents whose diffusion trajectories are jointly steered, via optimal control, toward a shared objective defined on their aggregated output. We validate our framework on conditional MNIST generation and compare it against a naïve inference-time DPS-style baseline replacing learned cooperative control with per-step gradient guidance [1].

1 Introduction

Continuous-time generative models, in particular score-based diffusion models (Song et al., 2020; Ho et al., 2020) and flow-based models (Lipman et al., 2023), have become ubiquitous in imaging, achieving state-of-the-art results in image restoration and synthesis. Building on this, substantial progress has been made in controllable generation, with techniques such as classifier guidance, conditional score modelling, and parameter-efficient fine-tuning (Chung et al., 2022; Zhang et al., 2023; Ruiz et al., 2023; Denker et al., 2024), allowing controlled generation under given constraints.

However, techniques for aggregating multiple pretrained diffusion models remain limited. Most frameworks for compositional generation operate directly on densities. Recent approaches cast algebraic composition via energy-based modelling or mixtures of experts (Liu et al., 2022; Du et al., 2023). For instance, given pretrained models with marginals $\{q_t^i(x)\}_{i=1}^N$, Feynman–Kac correctors (Skreta et al., 2025; Thornton et al., 2025) construct sampling schemes for an explicit target density. Representative sampling schemes include geometric averages $p_t^{\text{geo}}(x) \propto \prod_{i=1}^N q_t^i(x)^{\beta_i}$ with $\sum_{i=1}^N \beta_i = 1$, or product of experts $p_t^{\text{prod}}(x) \propto \prod_{i=1}^N q_t^i(x)$; see Appendix A for an in-depth discussion. These methods require the composed model to be specified explicitly at the level of diffusion-time densities. However, in practice the assumption that the target density is a known algebraic composition is often unrealistic and restrictive. For example, there is no *a priori* reason why combining a model capturing visual realism, with a model capturing consistency to a text prompt, leads to a geometric average, a product of experts, or any other predefined combination of densities.

A compositional model can instead be defined implicitly, as the minimiser of a task-specific objective, such as $\Psi(x) = \lambda_{\text{real}}\ell_{\text{real}}(x) + \lambda_{\text{align}}\ell_{\text{align}}(x)$, where ℓ_{real} measures realism and ℓ_{align} measures consistency to the text prompt. The aim here is for a composition whose samples minimise the ex-

[1] https://github.com/rb876/multiagent-diffusion-soc

pected value of Ψ. Importantly, such a minimisation does not necessarily correspond to any simple algebraic composition of the underlying marginals.

In this work, we propose a shift in perspective. In particular, we cast compositional generation as a cooperative *Stochastic Optimal Control* (SOC) problem, in which the reverse-time dynamics of multiple diffusion models are jointly steered toward a task-defined objective. We model pre-trained diffusion models as independent agents, and define the generated object as the aggregation of their interacting trajectories. By treating the composition as a control problem we can optimise for complex objectives defined by loss functions without requiring explicit knowledge of the form of the resulting composite density.

We make the following contributions:

1. We introduce a *cooperative multi-agent framework* for compositional generation that casts inference with multiple pre-trained diffusion models as an SOC problem.
2. We propose a *coordinate-wise optimisation scheme* based on iterative diffusion optimisation that enables multi-agent control.

We demonstrate the effectiveness of the proposed framework with initial experiments on MNIST.

2 Cooperative Multi-Agent Diffusion

Score-based diffusion models generate samples by simulating the time-reversal of a diffusion process whose marginals converge to a simple reference distribution, typically a standard Gaussian (Song et al., 2020). Let q_t denote the marginal density of the forward diffusion at time $t \in [0, T]$, with $q_0 = p_{\text{data}}$ and $q_T \approx \mathcal{N}(0, I)$. The forward diffusion process is characterised by the drift $\bar{f} : \mathbb{R}^n \times [0, T] \to \mathbb{R}^n$ and a diffusion coefficient $\bar{g} : [0, T] \to \mathbb{R}$. Following recent work in SOC (Domingo-Enrich et al., 2024a;b; Nüsken & Richter, 2021), we parameterise the reverse process to run forward in time. Initialised from noise, $X_0 \sim p_0 := q_T \approx \mathcal{N}(0, I)$, the time-reversal is given by

$$\mathrm{d}X_t = b(X_t, t)\mathrm{d}t + g(t)\,\mathrm{d}W_t, \quad \text{with } b(x,t) = -\left[f(X_t,t) - g(t)^2 \nabla_x \log p_t(X_t)\right]. \tag{1}$$

For brevity, we use $f(x,t) := \bar{f}(x, T-t)$, $g(t) := \bar{g}(T-t)$ and $p_t = q_{T-t}$. The score $\nabla_x \log p_t(x)$ is learned with a neural network $S(x, t; \theta) \approx \nabla_x \log p_t(x)$; refer to Appendix B.

Our goal is to enable composition by coordinating multiple pre-trained diffusion models. The composition is achieved by jointly steering their individual trajectories towards a joint objective. For this, we consider the following objective function for a collection of N agents $\{(X_t^{u,i})_t\}_{i=1,\dots,N}$,

$$\mathcal{J}(\{u^i\}_i, \vartheta) = \mathbb{E}\left[\int_0^T \left(\sum_{i=1}^N \lambda^i \left\| u^i(X_t^{u,i}, t; \{X_t^{u,j}\}_j)\right\|^2 + c(Y_t, t)\right) \mathrm{d}t + \Psi(Y_T)\right] \tag{2}$$

$$\mathrm{d}X_t^{u,i} = \left[b^i(X_t^{u,i}, t) + g(t) u^i(X_t^{u,i}, t; \{X_t^{u,j}\}_j)\right] \mathrm{d}t + g(t)\,\mathrm{d}W_t, \quad X_0^{u,i} \sim p_0, \tag{3}$$

which is minimised over control $\{u^i\}$ and parameters ϑ. Each agent is modelled as a *controlled* SDE with control u^i and drift b^i, given by the time-reversal in (3). As the control u^i depends on the state of all other agents, (3) is a coupled SDE system. Our goal is to estimate the control $u^i, i = 1, \dots, N$ such that the agents optimise some task-specific objective function. For this, we define an aggregator

$$Y_t = \varphi\Big(\{X_t^{u,i}\}_{i=1}^N,\, t;\, \vartheta\Big), \tag{4}$$

which may depend on additional learnable parameters $\vartheta \in \Theta$. The objective (2) consists of three terms: a weighted quadratic control cost , a running cost $c : \mathbb{R}^d \times [0, 1] \to \mathbb{R}$, and a terminal cost $\Psi : \mathbb{R}^d \to \mathbb{R}$. The terminal cost is evaluated on the aggregated output at terminal time $t = T$.

The aggregated state Y_t enters the objective through $c(Y_t, t)$ and $\Psi(Y_T)$, allowing both the running and terminal costs to encode the reward. The explicit running cost c acts as a stabiliser by providing dense-in-time gradients along the reverse-time dynamics, mitigating reliance on a potentially sparse terminal signal. We implement c as a surrogate of Ψ, evaluated at a joint Tweedie estimate

$$c(Y_t, t) = \alpha_t \Psi(\hat{Y}_0(\{X_t^{u,i}\}_i, t)), \text{ with } \hat{Y}_0(\{X_t^{u,i}\}_i, t) = \varphi\Big(\{\hat{X}_0^{u,i}\}_{i=1}^N,\, t;\, \vartheta\Big), \tag{5}$$

where $\hat{X}_0^{u,i}$ is an unconditional Tweedie estimate (Efron, 2011) and α_t is time-dependent scaling.

Algorithm 1 Control-wise Optimisation with Iterative Diffusion Optimisation (IDO)

Initialize $\{u_0^i\}_{i=1}^N$, aggregator parameters ϑ, step sizes η and η_ϑ, number of update steps M
for $k = 0, 1, \ldots$ until convergence **do**
  for $i = 1, \ldots, N$ **do**
    Freeze agents $\{u_k^j\}_{j \neq i}$
    for $m = 1, \ldots, M$ **do** ▷ Inner loop
      Sample trajectories $\{X_t^{u,j}\}_{j=1}^N$ by simulating the coupled SDEs (3)
      Compute Monte Carlo approximation $\widehat{\mathcal{J}}(\{u^i\}_i, \vartheta)$ of (2)
      Compute gradients $\nabla_{u^i}\widehat{\mathcal{J}}$ and $\nabla_\vartheta \widehat{\mathcal{J}}$
      Update control: $u^i \leftarrow u^i - \eta \nabla_{u^i}\widehat{\mathcal{J}}$
      Update aggregation parameters: $\vartheta \leftarrow \vartheta - \eta_\vartheta \nabla_\vartheta \widehat{\mathcal{J}}$
    end for
    Set $u_{k+1}^i \leftarrow u^i$
  end for
end for

Connection to classical SOC. The aggregated state Y_t itself follows an SDE, and the optimisation problem can be interpreted as an SOC problem directly on Y_t. A detailed discussion of (2), including how it is obtained and relates to an SOC objective on Y_t, is in Appendix C. When the aggregation operator corresponds to a disjoint concatenation of agent states, as in Section 3, (2) coincides with a classical SOC problem on the aggregated state.

2.1 Control-wise Descent via Iterative Diffusion Optimisation

We optimise (2) using a coordinate-descent scheme over the agent controls. At each outer iteration, we select an agent $i \in \{1, \ldots, N\}$ and update its control u^i, keeping the other controls $\{u^j\}_{j \neq i}$ fixed. This yields a sequence of single-agent SOC sub-problems. Each control-wise update is performed using Iterative Diffusion Optimisation (IDO) (Nüsken & Richter, 2021). IDO is a stochastic gradient method for SOC that estimates the objective and the gradient via Monte Carlo simulation of SDE trajectories. Given the current iterate u_k^i, one IDO step consists of:

1. Sampling trajectories of the coupled SDE system,
2. Computing the empirical loss and its gradient with respect to u^i along these trajectories,
3. Performing a gradient descent update on u^i.

The full optimisation alternates over agents, repeatedly applying IDO updates for each control network until convergence. The parameters ϑ of the aggregator can be updated concurrently to the control updates. The algorithm is given in Algorithm 1.

3 Experimental Evaluation and Discussion

We demonstrate our framework on a proof-of-concept task to illustrate its suitability. Additional results are reported in Appendix E.

We consider a composition task on MNIST (LeCun, 1998). The task is to generate a specific MNIST digit using multiple agents, each responsible for a different part of the image. The aggregation operator φ is implemented as fixed projection operator with non-overlapping regions. Every agent contributes with one horizontal stripe of the final aggregated images, see also the green regions in Figure 1. The terminal cost is given by the negative log-likelihood of a pre-trained MNIST classifier, i.e., $\Psi(\boldsymbol{x}) = -\log p(a|x)$ for some class a. To encourage spatial coherence across the composed image, we additionally include a seam-continuity loss, which penalises intensity and vertical gradient discontinuities along the boundaries between agent-controlled regions. The intermediate cost is implemented using the Tweedie projection (5). Since the cost is evaluated only on the aggregated state, each individual agent $X_t^{u,i}$ may generate a digit of any class, provided the aggregate belongs to the target class. We compare the learned control with an inference-time approximation, similar to DPS (Chung et al., 2022). Here, we use the heuristic approximation, referred to as CDPS, to the

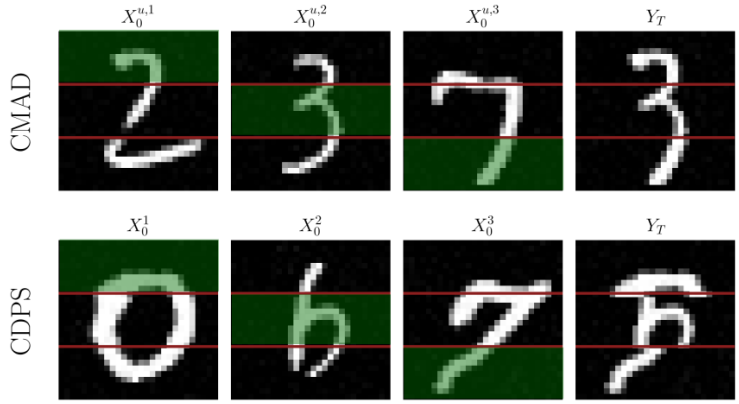

Figure 1: A single sample generated with 3 agents for the target 3. Every agent controls one horizontal stripe (■ coded) of the aggregated state Y_t. We show the state $X_0^{u,i}$ for every agent.

Table 1: Performance comparison across different numbers of agents. We report mean classification accuracy (%) and mean terminal loss, computed over 1024 samples.

		digit 0		digit 3		digit 9	
# Agents	Method	Acc.	Ψ	Acc.	Ψ	Acc.	Ψ
2	CDPS	98.54	0.378	95.70	0.420	98.44	0.387
	CMAD (joint)	99.71	0.166	99.41	0.094	99.32	0.232
	CMAD (control-wise)	98.93	0.173	99.90	0.064	98.34	0.249
3	CDPS	97.95	0.708	97.17	0.638	96.97	0.635
	CMAD (joint)	99.61	0.393	97.85	0.381	98.93	0.192
	CMAD (control-wise)	99.80	0.190	99.71	0.131	98.14	0.400

control u^i as

$$\hat{u}^i_{\text{CDPS}}(X_t^{u,i}, t) = \alpha \nabla_{X_t^{u,i}} \Psi(\hat{Y}_0(\{X_t^{u,i}\}_i, t)), \tag{6}$$

with $\alpha > 0$. We parameterise the control using a *reward-informed inductive bias*, combining a learned drift correction with an explicit reward-gradient guidance term, see Appendix D for details (Denker et al., 2025; Venkatraman et al., 2024; Zhang & Chen, 2021).

Table 1 reports the mean classification accuracy and mean terminal loss Ψ for CDPS and CMAD. For CMAD, we evaluate both the control-wise optimization scheme and a joint optimization approach, where all controls are updated jointly. All methods achieve high classification accuracy across digits. However, CMAD consistently attains a lower terminal loss. Qualitative results in Figure 1 further indicate that CDPS occasionally produces visually unnatural samples, whereas CMAD generates more realistic digit images, albeit with reduced diversity compared to CDPS.

4 Conclusion and Future Work

This work introduces a novel multi-agent framework for compositional generation of multiple pretrained diffusion models. Our initial experiments on compositional generation for MNIST digits show that this framework provides control even if the target task cannot be explicitly written as an algebraic composition (e.g., geometric average, product of experts) of the marginals of the pretrained diffusion models. This requires extending common path-wise gradient estimators (Clark et al., 2023; Schulman et al., 2017) or techniques such as adjoint matching (Domingo-Enrich et al., 2024a) to our setting. Our numerical results in Section 3 show that the control-wise update is able to optimise the SOC objective. We want to study under which conditions the control-wise update actually leads to a provable convergent algorithm. Further, we want to expand the connection to differential games and fictitious-play dynamics (Hu, 2019). Finally, it would be interesting to investigate settings in which the aggregation operator is itself parametrised and learned jointly with the control.

Acknowledgments

RB and AD acknowledge support from the EPSRC (EP/V026259/1). ZK acknowledges support from the EPSRC (EP/X010740/1).

References

Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

Richard E Bellman and Stuart E Dreyfus. *Applied dynamic programming*. Princeton university press, 2015.

Denis Blessing, Julius Berner, Lorenz Richter, Carles Domingo-Enrich, Yuanqi Du, Arash Vahdat, and Gerhard Neumann. Trust region constrained measure transport in path space for stochastic optimal control and inference. *arXiv preprint arXiv:2508.12511*, 2025.

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.

Kevin Clark, Paul Vicol, Kevin Swersky, and David J Fleet. Directly fine-tuning diffusion models on differentiable rewards. *arXiv preprint arXiv:2309.17400*, 2023.

Alexander Denker, Francisco Vargas, Shreyas Padhy, Kieran Didi, Simon Mathis, Vincent Dutordoir, Riccardo Barbano, Emile Mathieu, Urszula Julia Komorowska, and Pietro Lio. DEFT: Efficient fine-tuning of diffusion models by learning the generalised h-transform. In *Advances in Neural Information Processing Systems*, volume 37, pp. 19636–19682. Curran Associates, Inc., 2024. doi: 10.52202/079017-0620.

Alexander Denker, Shreyas Padhy, Francisco Vargas, and Johannes Hertrich. Iterative importance fine-tuning of diffusion models. *arXiv preprint arXiv:2502.04468*, 2025.

Carles Domingo-Enrich, Michal Drozdzal, Brian Karrer, and Ricky TQ Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. *arXiv preprint arXiv:2409.08861*, 2024a.

Carles Domingo-Enrich, Jiequn Han, Brandon Amos, Joan Bruna, and Ricky T. Q. Chen. Stochastic optimal control matching. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b.

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.

Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011.

Wendell H Fleming and Raymond W Rishel. *Deterministic and stochastic optimal control*, volume 1. Springer Science & Business Media, 2012.

Ulrich G Haussmann and Etienne Pardoux. Time reversal of diffusions. *The Annals of Probability*, pp. 1188–1205, 1986.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arxiv:2006.11239*, 2020.

Ruimeng Hu. Deep fictitious play for stochastic differential games. *arXiv preprint arXiv:1903.09376*, 2019.

Chieh-Hsin Lai, Yang Song, Dongjun Kim, Yuki Mitsufuji, and Stefano Ermon. The principles of diffusion models. *arXiv preprint arXiv:2510.21890*, 2025.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. Compositional visual generation with composable diffusion models. In *European conference on computer vision*, pp. 423–439. Springer, 2022.

Edward Nelson. *Dynamical theories of Brownian motion*, volume 3. Princeton university press, 1967.

Nikolas Nüsken and Lorenz Richter. Solving high-dimensional hamilton–jacobi–bellman pdes using neural networks: perspectives from the theory of controlled diffusions and measures on path space. *Partial differential equations and applications*, 2(4):48, 2021.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22500–22510, 2023.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

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.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

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. *arXiv preprint arXiv:2502.12786*, 2025.

Siddarth Venkatraman, Moksh Jain, Luca Scimeca, Minsu Kim, Marcin Sendera, Mohsin Hasan, Luke Rowe, Sarthak Mittal, Pablo Lemos, Emmanuel Bengio, et al. Amortizing intractable inference in diffusion models for vision, language, and control. *Advances in neural information processing systems*, 37:76080–76114, 2024.

H.M. Soner Wendell H. Fleming. *Controlled Markov Processes and Viscosity Solutions*. Springer New York, NY, 2 edition, 2006.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3836–3847, 2023.

Qinsheng Zhang and Yongxin Chen. Path integral sampler: a stochastic control approach for sampling. *arXiv preprint arXiv:2111.15141*, 2021.

A Related Work

Much of the literature on compositional generation focuses on sampling methods for drawing from specific combinations of the underlying diffusion model densities (Liu et al., 2022; Du et al., 2023; Skreta et al., 2025). For example, Du et al. (2023) considers sampling from product-of-experts (PoE), mixtures of densities or negation. Conceptually, we differ from this line of work. In particular, we do not define a specific combination of densities, we rather define an objective function which should be minimised by the composed model. By doing this, we are implicitly learning the type of composition, which works best for the task at hand.

The work of Du et al. (2023) formulates compositional generation as a sampling problem. Composition is performed at the level of densities (scores or energies), and correctness is restored by combining reverse diffusion with MCMC steps. Consider a PoE distribution $p_t^{\text{prod}}(x) \propto \prod_{i=1}^{N} q_t^i(x)$. In the PoE framework, one takes a product of N distributions and renormalizes to form a new distribution representing the intersection of the component supports, such that regions of high probability under p^{prod} correspond to regions of high probability under all q_t^i.

However, sampling from this product distribution using standard reverse diffusion would require access to the score of the diffused product distribution. In general, this score does not decompose as

$$\nabla_{x_t} \log p_t^{\text{prod}}(x_t) \;\neq\; \sum_{i=1}^{N} \nabla_{x_t} \log q_t^i(x_t). \tag{7}$$

Indeed, under a forward Itô SDE with transition density $q(x_t|x_0)$, the time–t marginal of the PoE distribution is

$$p_t^{\text{prod}}(x_t) = \int \left(\prod_{i=1}^{N} q_0^i(x_0) \right) q(x_t|x_0)\, dx_0.$$

Its score is therefore

$$\nabla_{x_t} \log p_t^{\text{prod}}(x_t) = \nabla_{x_t} \log \int \left(\prod_{i=1}^{N} q_0^i(x_0) \right) q(x_t|x_0)\, dx_0.$$

Summing the individual scores corresponds to implicitly pushing the logarithm inside the integral, which invokes Jensen's inequality

$$\log \int \left(\prod_{i=1}^{N} q_0^i(x_0) \right) q(x_t|x_0)\, dx_0 \;\geq\; \int \left(\sum_{i=1}^{N} \log q_0^i(x_0) \right) q(x_t|x_0)\, dx_0.$$

This manipulation enforces an additive structure at the cost of introducing a biased lower bound, so the resulting score is only a crude approximation. Differentiating both sides would yield

$$\nabla_{x_t} \log p_t^{\text{prod}}(x_t) \;\approx\; \sum_{i=1}^{N} \nabla_{x_t} \int \log q_0^i(x_0)\, q(x_t|x_0)\, dx_0.$$

Such approximations are known to lead to poor generation quality in practice Du et al. (2023); Liu et al. (2022). This observation further motivates our approach. Rather than correcting the sampling procedure via MCMC, we adopt a pragmatic control-based formulation that avoids diffusion-time PoE sampling.

B Background

B.1 Generative Models as Continuous-Time Stochastic Processes

Score-based diffusion models admit a continuous-time formulation via stochastic calculus (Lai et al., 2025; Song et al., 2020). Let $(\Omega, \mathcal{F}, (\mathcal{F}_s)_{s\geq 0}, \mathcal{P})$ be a filtered probability space carrying a Wiener

process $W = (W_s)_{s\geq 0}$, and consider the SDE-based generative model $X = (X_s)_{s\in[0,T]}$, an $\mathbb{R}^d$-valued stochastic process obtained by integrating the reverse-time SDE associated with a forward (noising) mechanism. The forward Itô SDE is

$$\mathrm{d}X_s = \bar{f}(X_s, s)\,\mathrm{d}s + \bar{g}(s)\,\mathrm{d}W_s, \qquad X_0 \sim q_0, \tag{8}$$

where the drift $\bar{f} : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$ follows the standard regularity assumptions for well-posedness, the (isotropic) diffusion scale $\bar{g} : [0,T] \to \mathbb{R}_+$ is continuous, and q_s denotes the marginal density of X_s. Intuitively, we can view the forward (noising) process as an Itô SDE, which gradually perturbs the data distribution $q_0 \coloneqq p_{\text{data}}$ into $q_T \approx \mathcal{N}(0, I)$. The law of the SDE solution X_s, together with its marginal at $s = 0$, induces a forward path measure on the space of continuous trajectories $\mathcal{C}([0,1], \mathbb{R}^d)$. Under suitable regularity assumptions (Nelson, 1967; Anderson, 1982; Haussmann & Pardoux, 1986), there exists a time-reversal defined as

$$\mathrm{d}X_t = b(X_t, t)\,\mathrm{d}t + g(t)\,\mathrm{d}W_t, \qquad X_0 \sim p_0 \coloneqq q_T \approx \mathcal{N}(0, I), \tag{9}$$

with backward drift:

$$b(x,t) \coloneqq \bar{f}(x, T-t) - \bar{g}(T-t)^2\,\nabla_x \log q_{T-t}(x),$$

where $g(t) \coloneqq \bar{g}(T-t)$, and the marginal in reverse time is defined as $p_t(x) \coloneqq q_{T-t}(x)$. Similar to the recent SOC literature (e.g., (Domingo-Enrich et al., 2024a)) we write the time-reversal in forward time, i.e., transforming noise at $t = 0$ to a sample of the data distribution at $t = T$.

Sampling therefore consists of drawing $X_0 \sim p_0$ and evolving the SDE (9) to obtain X_T, whose law approximates the target data distribution. Using the denoising score-matching objective

$$\mathbb{E}_{X_0\sim p_{\text{data}},\, X_s\sim p_s(\cdot|X_0)}\left[\left\|\nabla_{X_s}\log p_s(X_s) - S(X_s, s; \theta)\right\|^2\right],$$

one trains the network S to approximate the score $\nabla_x \log q_s$.

B.2 Stochastic Optimal control

The study of optimisation problems over SDEs is known as stochastic optimal control (Bellman & Dreyfus, 2015; Fleming & Rishel, 2012). The quadratic cost, affine-control SOC formulation is

$$\min_{u\in\mathcal{U}}\left\{\mathcal{J}(u) \coloneqq \mathbb{E}_{\mathbb{P}_u}\left[\int_0^T \left(\tfrac{1}{2}\|u(X_t^u, t)\|^2 + c(X_t^u, t)\right)\mathrm{d}t + \Psi(X_T^u)\right]\right\}, \tag{10}$$

$$\text{s.t.}\quad \mathrm{d}X_t^u = (b(X_t^u, t) + g(t)u(X_t^u, t))\,\mathrm{d}t + g(t)\mathrm{d}W_t, \qquad X_0^u \sim p_0. \tag{11}$$

where $X_t^u \in \mathbb{R}^d$ is the state of the controlled stochastic process and $\mathbb{P}^u$ denotes the probability measure on trajectories $\{(X_t^u)_{t\in[0,T]}(\omega)\}_{\omega\in\Omega} \subset C([0,T], \mathbb{R}^d)$, where each ω denotes a sampled trajectory generated by the controlled SDE. Here $u : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$ is the control, which belongs to the set of admissible controls $\mathcal{U}$.

As part of the objective we have a quadratic control cost $\frac{1}{2}\|u(X_t^u, t)\|^2$, the state running cost $c : \mathbb{R}^d \times [0,T] \to \mathbb{R}$ and a terminal cost $\Psi : \mathbb{R}^d \to \mathbb{R}$. The control is often parametrised using a neural network, i.e., $u(\cdot, \cdot; \phi)$ with ϕ denoting a finite-dimensional parameter vector, and learned by minimising Eq. (10). In other words, the problem of finding a minimizer of the SOC problem in Eq. (10) over the space of admissible controls u is recast as the problem of finding a minimizer ϕ over a parameter space.

One of classical methods to solve this optimization problem is by defining the value function

$$V(x,t) \coloneqq \inf_{u\in\mathcal{U}} \mathbb{E}\left[\int_t^T \left(\frac{1}{2}\|u(X_s^u, s)\|^2 + c(X_s^u, s)\right) ds + \Psi(X_T^u)\middle| X_t^u = x\right]$$

Then, under standard regularity assumptions, the value function V is the solution to the Hamilton–Jacobi–Bellman (HJB) equation

$$\partial_t V(x,t) + \inf_{u\in\mathcal{U}}(\mathcal{L}^u V(x,t) + c(x,t) + \frac{1}{2}\|u\|^2) = 0, V(x,0) = \Psi(x)$$

where $\mathcal{L}^u$ is the infinitesimal generator of the time-reverse controlled SDE. For more details, we refer the reader to (Wendell H. Fleming, 2006, Chap. 4) combined with the fact that time-reverse diffusion process is a well-defined Markov diffusion process first shown in Haussmann & Pardoux (1986).

Another equivalent view is to consider the SOC as optimizing a measure on trajectories. Let $\mathbb{P}$ denote the law of uncontrolled SDE

$$\mathrm{d}X_t = b(X_t, t)\mathrm{d}t + g(t)\mathrm{d}W_t, \qquad X_0 \sim p_0$$

Girsanov theorem gives the Radon-Nikodym derivative (RND) of the law $\mathbb{P}$ of uncontrolled SDE w.r.t. the law $\mathbb{P}^u$ of the controlled SDE to be

$$\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{P}^u} = \exp\left(-\int_0^T u(X_s^u)\cdot \mathrm{d}\overline{W}_t - \frac{1}{2}\int_0^T \|u(X_s^u, s)\|^2 \mathrm{d}s\right)$$

By defining the work functional

$$\mathcal{W}(X) := \int_0^T c(X_s, s)\mathrm{d}s + \Psi(X_T),$$

the RND of the optimal path measure $\mathbb{P}^{u_*}$ is given as

$$\frac{\mathrm{d}\mathbb{P}^{u_*}}{\mathrm{d}\mathbb{P}}(X) = \frac{\exp\left(-\mathcal{W}(X)\right)}{\mathcal{Z}}$$

where $\mathcal{Z} = \mathcal{Z}(X_0^u)$ is the normalising constant ensuring consistency between the optimal path measure and the "initial condition". In the setting of our work, it ensures $\mathbb{P}_0^{u_*}(X_0) = p_0$, and thus it is completely determined by p_0 here. Now, combining two Radon-Nikodym derivatives together, we could compute the reverse Kullback-Leibler divergence

$$D_{\mathrm{KL}}(\mathbb{P}|\mathbb{P}^{u_*}) = \mathbb{E}\left[\int_0^T \left(\frac{1}{2}\|u(X_s^u, s)\|^2 + c(X_s^u, s)\right)\mathrm{d}s + \Psi(X_T) + \log\mathcal{Z}\right].$$

By comparing $D_{\mathrm{KL}}(\mathbb{P}|\mathbb{P}^{u_*})$ to $\mathcal{J}(u)$, it can be noted that they share the same minimiser, i.e., minimisation of the original objective functional can be equivalently converted to the minimisation of the reverse KL divergence $D_{\mathrm{KL}}(\mathbb{P}|\mathbb{P}^{u_*})$.

For more details, we refer the reader to (Blessing et al., 2025, App. D) and Nüsken & Richter (2021).

IDO viewpoint and connection to our algorithms. In practice, we use neural network to parameterise $u(x,t)$ by $u(x,t;\phi)$ and minimise $\mathcal{J}(u)$ over ϕ. This is framework of iterative diffusion optimisation (IDO) shown in Algorithm 2: simulate controlled rollouts of (11) (Euler–Maruyama), differentiate a Monte Carlo estimate of (10), and update parameter ϕ to optimise our objective functional $\mathcal{J}(u(\phi))$. In our multi-agent setting, the control is $u = (u^1, ..., u^N)$, the state is $(X^{u,1}, ..., X^{u,N})$, and costs are rewritten as depending on the aggregated state $Y_t = \varphi(\{X_t^{u,i}\}_{i=1}^N, t)$, see (2). In Algorithm 2, $\mathcal{J}$ is a single-trajectory Monte Carlo estimator of $\mathcal{J}(u)$

Algorithm 2 Iterative diffusion optimisation for SOC via `BPTT`

Require: Initial control parameters ϕ; number of gradient steps M; batch size B; number of time steps K; SOC objective $\mathcal{J}$

1: **for** $m = 1$ **to** M **do**
2: Simulate B rollouts of the controlled process using K Euler–Maruyama steps
3: Compute the Monte Carlo estimate $\widehat{\mathcal{J}}$ from the B rollouts
4: Update ϕ with a stochastic gradient step on $\widehat{\mathcal{J}}$
5: **end for**
6: **return** learned control $u(\cdot, \cdot; \phi)$

after time discretization,

$$\mathcal{J}(u; X \sim \mathbb{P}^u) := \int_0^T \left(\tfrac{1}{2}\|u(X_t^u, t)\|^2 + c(X_t^u, t)\right)\mathrm{d}t + \Psi(X_T^u). \tag{12}$$

with the aim of computing the gradient of Eq. (12) with respect to the parameters ϕ of the control.

Numerically, Eq. (12) is implemented via backpropagation through time (BPTT). This approach uses a numerical solver (e.g., Euler–Maruyama) for simulating the SDE, stores the whole trajectory in memory and then differentiates through all time steps. Alternatively, one can leverage the continuous-time nature of the SDE and use the continuous adjoint method, in which the gradient of the control objective with respect to the state trajectory is first derived analytically as an adjoint ODE and subsequently discretized and solved numerically.

For more machine-learning-oriented discussions of the matter, we refer the reader to Domingo-Enrich et al. (2024b), (Domingo-Enrich et al., 2024a, Sec. 5.1), and (Blessing et al., 2025, App. D1).

C Complementary Material for Section 2

C.1 Characterisation of the Aggregated Process Y_t in Eq. (4)

We characterise the dynamics of the aggregated state process Y_t, first for scalar states ($d = 1$) and then for vector-valued states ($d > 1$).

Case $d = 1$: Let the state be represented in vectorized form as $X_t^u \in \mathbb{R}^{Nd}$, corresponding to the vectorization of an underlying matrix in $\mathbb{R}^{N\times d}$, with blocks $X_t^{u,i} \in \mathbb{R}^d$, $i = 1, \dots, N$. Define the real-valued process $Y_t = \varphi(X_t^u, t) \in \mathbb{R}$ and assume that $\varphi : \mathbb{R}^{Nd} \times \mathbb{R}^+ \to \mathbb{R}$ is continuously differentiable in time and twice continuously differentiable with respect to the state variable. By Itô's formula,

$$\mathrm{d}Y_t = \partial_t\varphi(X_t^u, t)\,\mathrm{d}t + (\nabla_X\varphi(X_t^u, t))^\top \cdot \mathrm{d}X_t^u + \frac{1}{2}(\mathrm{d}X_t^u)^\top H_X X_t^u$$

Substituting the controlled dynamics

$$\mathrm{d}X_t^u = \big(b(X_t^u, t) + g(t)u(X_t^u, t)\big)\,\mathrm{d}t + g(t)\,\mathrm{d}W_t,$$

we get

$$\begin{aligned}\mathrm{d}Y_t = &\partial_t\varphi(X_t^u, t)\,\mathrm{d}t + (\nabla_X\varphi(X_t^u, t))^\top \cdot \big(b(X_t^u, t) + g(t)u(X_t^u, t)\big)\,\mathrm{d}t \\ &+ g(t)(\nabla_X\varphi(X_t^u, t))^\top \cdot \mathrm{d}W_t + \frac{1}{2}g^2(t)\Delta_X\varphi(X_t^u, t)\mathrm{d}t\end{aligned}$$

In the above derivation, we have assumed a scalar diffusion coefficient $g(t)$ acting identically on each component of the state, i.e. an isotropic diffusion of the form $g(t)I$.

Case $d > 1$: Let the state be represented in vectorized form as $X_t^u \in \mathbb{R}^{Nd}$, corresponding to the vectorization of an underlying matrix in $\mathbb{R}^{N\times d}$, with blocks $X_t^{u,i} \in \mathbb{R}^d$, $i = 1, \dots, N$. Define the vector-valued process $Y_t = \varphi(X_t^u, t) \in \mathbb{R}^d$ and assume that $\varphi : \mathbb{R}^{Nd} \times \mathbb{R}^+ \to \mathbb{R}^d$ is continuously differentiable in time and twice continuously differentiable with respect to the state variable. Let also $W_t \in \mathbb{R}^{Nd}$ be a (standard) Wiener process. Applying Itô's formula component-wise,

$$\mathrm{d}Y_t = \partial_t\varphi(X_t^u, t)\,\mathrm{d}t + D_X\varphi(X_t^u, t)\,\mathrm{d}X_t^u + \frac{1}{2}\bar{g}(t)^2 \begin{pmatrix}\mathrm{Tr}(D_X^2\varphi^1(X_t^u, t))\\ \vdots \\ \mathrm{Tr}(D_X^2\varphi^d(X_t^u, t))\end{pmatrix}\mathrm{d}t.$$

Here φ^j denotes the j-th component of φ. The Jacobian of φ with respect to the state variable is $D_X\varphi(X, t) \in \mathbb{R}^{d\times Nd}$, and, for each j the Hessian with respect to the state variable is $D_X^2\varphi^j(X, t) \in \mathbb{R}^{Nd\times Nd}$ and the trace is defined as,

$$\mathrm{Tr}(D_X^2\varphi^j) = \sum_{\ell=1}^{Nd} \frac{\partial^2\varphi^j}{\partial x_\ell^2},$$

where x_ℓ are the coordinates of X_t^u. Substituting the controlled dynamics

$$\mathrm{d}X_t^u = \big(b(X_t^u, t) + g(t)u(X_t^u, t)\big)\,\mathrm{d}t + g(t)\,\mathrm{d}W_t,$$

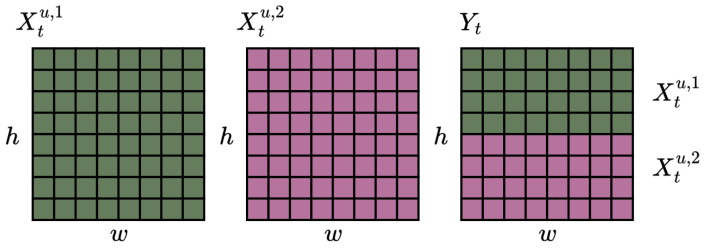

Figure 2: The schematic above illustrates linear stacking induced by a non-overlapping selection mask. For clarity, we do not use vectorised state representations.

where $b(\cdot,\cdot)$ and $u(\cdot,\cdot)$ take values in $\mathbb{R}^{Nd}$, matching the dimension of the state and the driving Wiener process, we obtain

$$\begin{aligned}\mathrm{d}Y_t = \partial_t\varphi(X_t^u,t)\,\mathrm{d}t + D_X\varphi(X_t^u,t)\big(b(X_t^u,t)+g(t)u(X_t^u,t)\big)\,\mathrm{d}t \\ + g(t)\,D_X\varphi(X_t^u,t)\,\mathrm{d}W_t + \frac{1}{2}g(t)^2\begin{pmatrix}\mathrm{Tr}(D_X^2\varphi^1)\\ \vdots \\ \mathrm{Tr}(D_X^2\varphi^d)\end{pmatrix}\mathrm{d}t.\end{aligned}$$

Masking selection with non-overlapping regions. Suppose Y_t is a linear masking (selection) map $Y_t = \varphi(X_t^u,t) := M\,X_t^u \in \mathbb{R}^k$, with $M \in \{0,1\}^{k\times Nd}$, where k is the dimensionality of the aggregated process. In this work, k is set to be always d. Note that by *non-overlapping* means no coordinate of X_t^u is selected twice, equivalently $MM^\top = I_k$. Since φ is linear in X, all second-order Itô terms vanish and

$$\mathrm{d}Y_t = M\mathrm{d}X_t^u.$$

If the controlled dynamics are

$$\mathrm{d}X_t^u = \big(b(X_t^u,t)+g(t)u(X_t^u,t)\big)\,\mathrm{d}t + g(t)\,\mathrm{d}W_t, \qquad \overline{W}_t \in \mathbb{R}^{Nd},$$

then

$$\mathrm{d}Y_t = M\big(b(X_t^u,t)+g(t)u(X_t^u,t)\big)\,\mathrm{d}t + g(t)\,M\,\mathrm{d}W_t.$$

Moreover,

$$\mathrm{Cov}(M\,\mathrm{d}W_t) = MM^\top\,\mathrm{d}t = I_k\,\mathrm{d}t,$$

so $M\,\mathrm{d}W_t$ is a standard k-dimensional Wiener increment. Hence there exists a k-dimensional Wiener motion $\widetilde{W}_t$ such that

$$M\,\mathrm{d}W_t = \mathrm{d}\widetilde{W}_t,$$

and equivalently

$$\mathrm{d}Y_t = M\big(b(X_t^u,t)+g(t)u(X_t^u,t)\big)\,\mathrm{d}t + g(t)\,\mathrm{d}\widetilde{W}_t.$$

Example. Assume $N=2$ and each controlled process $X_t^{u,i} \in \mathbb{R}^d$ is a vectorised image ($d = w\times h$). Let

$$X_t^u = \begin{bmatrix}X_t^{u,1}\\ X_t^{u,2}\end{bmatrix} \in \mathbb{R}^{2d}, \qquad Y_t \in \mathbb{R}^d.$$

Assume d is even and set $d_1 = d/2$. We define the aggregated process Y_t by selecting the first half of the pixels from $X_t^{u,1}$ and the second half from $X_t^{u,2}$. This corresponds to a masking matrix $M \in \{0,1\}^{d\times 2d}$ of the form

$$M = \begin{bmatrix}I_{d_1} & 0 & 0 & 0\\ 0 & 0 & 0 & I_{d_1}\end{bmatrix},$$

where the blocks have sizes compatible with the decomposition $\mathbb{R}^{2d} = \mathbb{R}^{d_1}\oplus\mathbb{R}^{d_1}\oplus\mathbb{R}^{d_1}\oplus\mathbb{R}^{d_1}$., where $\oplus$ denotes the direct sum of vector spaces. Then

$$Y_t = MX_t^u = \begin{bmatrix}X_t^{u,1}[1{:}d_1]\\ X_t^{u,2}[d_1{+}1{:}d]\end{bmatrix}.$$

Each row of M contains exactly one nonzero entry and no input coordinate is selected twice, so the mask is non-overlapping and satisfies

$$MM^\top = I_d.$$

C.2 Discussion on SOC Objectives in Eq. (2)

The simple cooperative objective introduced in Section 2 and formalised in Eq. (2) is further discussed below, together with its relation to a SOC objective. We begin by defining an aggregated process $Y_t = \varphi(\{X_t^{u,i}\}_{i=1}^N)$, together with the stochastic differential equation it satisfies. The aggregation operator φ in this work corresponds either to a non-overlapping masked concatenation, as detailed in Appendix C.1. Note that the controls act on the individual agent dynamics and induce an effective control on the aggregated process through φ. Rather than deriving an SOC objective directly from Eq. (2), we introduce the following SOC problem posed on the aggregated dynamics as a modelling choice,

$$\min_{u_Y \in \mathcal{U}} \mathbb{E}\left[\int_0^1 \Big(\|u_Y(Y_t,t)\|^2 + \lambda\, c(Y_t,t)\Big)\,\mathrm{d}t + \Psi(Y_0)\right], \tag{13}$$

where $u_Y(Y_t,t) : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ denotes the control induced on the aggregated state by the agent-wise controls. Importantly, the form of the induced control $u_Y(Y_t,t)$ depends on the choice of aggregation operator φ.

We consider the stacking case, which can be viewed as a special instance of the non-overlapping masked concatenation discussed in Appendix C.1. Here the aggregation map is (implicitly) the identity on the augmented space and each agent is the disjoint block of coordinates, which corresponds to a coordinate-level product space construction. If $u_Y(Y_t,t) = \oplus_i u^i(X_t^u,t)$, the aggregation map corresponds to a stacking operation, $\varphi(\{X_t^{u,i}\}) = \oplus_{i=1}^N X_t^{u,i}$, with $X_t^{u,i} \in \mathbb{R}^{d/N}$. In this setting, the control naturally acts in the aggregated state space and the quadratic control cost satisfies

$$\|u_Y(Y_t,t)\|^2 = \big\|\oplus_{i=1}^N u^i\big(X_t^u,t;\varphi^i\big)\big\|^2 = \sum_{i=1}^N \big\|u^i\big(X_t^u,t;\phi^i\big)\big\|^2 \text{ (when blocks are disjoint)}$$

On the importance of the quadratic cost. The quadratic term $\sum_{i=1}^N \|u^i\|^2$ acts as a regularizer that penalizes deviations from the pretrained reverse-time dynamics. Since the uncontrolled drift for component i is b^i, minimizing $\|u^i\|$ keeps the effective drift $b^i + u^i$ close to the pretrained model.

Masked selection with non-overlapping regions. When ϕ corresponds to a non-overlapping masked concatenation, the aggregation is linear and given by

$$Y_t = MX_t^u, \qquad M \in \{0,1\}^{d\times Nd}, \qquad MM^\top = I_d.$$

In this case, the induced control satisfies

$$u_Y(Y_t,t) = Mu(X_t^u,t),$$

where $u(X_t^u,t) \in \mathbb{R}^{Nd}$ and the quadratic control cost is preserved under aggregation:

$$\|u_Y(Y_t,t)\|^2 = \sum_{i=1}^N \|u^i(X_t^u,t)\|^2.$$

Consequently, Eq. (2) coincides with a classical SOC problem posed on the aggregated state Y_t.

D Algorithmic Implementation

Here we detail the algorithmic implementation of the backpropagation-through-time (BPTT) formulation of our framework, together with two variants of iterative diffusion optimisation. The first is the joint optimisation scheme, in which all control policies are updated simultaneously. The second is a control-wise scheme inspired by fictitious-play dynamics, which enables improved scalability. In particular, we detail lines 6-7 in Algorithm 1.

The complete backpropagation-through-time update is also detailed in Algorithm 3, which computes Monte Carlo estimates of the control energy, path-wise cost, and terminal cost from sampled controlled trajectories and differentiates the resulting objective with respect to the control parameters. In the following paragraphs, we discuss key technical details and design choices.

Running cost via Tweedie look-ahead - ▷■ in Algorithm 3. A key design choice is the explicit incorporation of a non-trivial running state cost. Rather than evaluating the running criterion directly on the noisy states, we compute it on the Tweedie (denoised) look-ahead states

$$\widehat{X}_0^{u,i} = \frac{X_t^{u,i} + \sigma(t)^2 S(X_t^{u,i}, t; \theta)}{\alpha(t)}, \quad \text{as } \texttt{Tweedie} \text{ in Algorithm 3,}$$

and aggregate them into,

$$\widehat{Y}_0 := \varphi(\{\widehat{X}_0^{u,i}\}_{i=1}^N).$$

Here, $\sigma(t)$ and $\alpha(t)$ denote the noise standard deviation and signal scaling, respectively, induced by the variance-preserving (VP) diffusion process, such that $X_t = \alpha(t)X_0 + \sigma(t)\varepsilon$ with $\varepsilon \sim \mathcal{N}(0, I)$. The running cost is then accumulated along the diffusion trajectory and combined with the control energy and terminal cost to form the stochastic optimal control objective.

Control parametrisation - ▷■ in Algorithm 3. We parametrize each control using a *reward-informed inductive bias* given as

$$u^i(X_t^{u,i}, t, \{X_t^{u,i}\}_i; \phi^i) = \text{NN}_1(X_t^{u,i}, Y_t, t; \phi^i) + \text{NN}_2(t)\nabla_{\hat{X}_0^{u,i}} \Psi(\hat{Y}_0(\{X_t^{u,i}\}_i, t)), \tag{14}$$

where $\hat{X}_0^{u,i}$ is the Tweedie estimate given the pre-trained unconditional diffusion model, Ψ is the terminal loss function, $\text{NN}_1 : \mathbb{R}^d \times \times \mathbb{R}^d \times [0, T] \to \mathbb{R}^d$ is a vector-valued and $\text{NN}_2 : [0, T] \to \mathbb{R}^d$ a scalar-valued neural network. We initialise the last layer of NN_1 to be zero and NN_2 to be constant. Further, NN_1 gets both the current state $X_t^{u,i}$ as well as the current aggregated state Y_t as an input. This type of network parametrisation has recently been applied to many SOC problems, both for fine-tuning, sampling from energy function or solving inverse problems (Denker et al., 2025; Venkatraman et al., 2024; Zhang & Chen, 2021). A key distinction from the CDPS approximation in (6) is that the gradient of the loss Ψ is taken only with respect to the Tweedie estimate $\hat{X}_0^{u,i}$. As a result, we do not have to back-propagate through the unconditional score model, reducing both computation time and memory cost of this parametrization.

Running and state cost. This framework allows both c and Ψ to encode the reward, as the aggregator φ enters the SOC formulation indirectly through the loss terms. In contrast, we place the reward directly on the aggregated state, which induces visual alignment between agents; we refer to this design choice as *reward-driven image coherence*. In practice, we interpret the running cost as a surrogate of Ψ evaluated on the de-noised aggregated state $\hat{Y}_0$, obtained via Tweedie's update.

Both the running cost c (evaluated on the Tweedie estimate $\widehat{Y}_0$) and the terminal cost Ψ (evaluated on the final state $Y_{t_{k-1}}$) are defined as

$$c(\widehat{Y}_0, y^\star) := -\log p(y^\star|\widehat{Y}_0) + \ell_{\text{seam}}(\widehat{Y}_0), \qquad \Psi(Y_{t_{K-1}}, y^\star) := -\log p(y^\star|Y_{t_{k-1}}) + \ell_{\text{seam}}(Y_{t_{k-1}}).$$

The seam loss $\mathcal{L}_{\text{seam}}(Y)$ enforces continuity across predefined seams of Y and is defined as a weighted sum of intensity and vertical-gradient discrepancies between adjacent seam rows,

$$\mathcal{L}_{\text{seam}}(Y) := \sum_{(r_p, r_q) \in \mathcal{S}} \Big(\beta \, \| \, Y_{r_p} - Y_{r_q} \, \|_\rho \; + \; \gamma \, \| \, \nabla_y Y_{r_p} - \nabla_y Y_{r_q} \, \|_\rho \Big),$$

where $\mathcal{S}$ denotes the set of seam row pairs, ∇_y is the vertical finite-difference operator, $\| \cdot \|_\rho$ denotes the Charbonnier penalty $\rho(x) = \sqrt{x^2 + \varepsilon^2}$, and $\beta, \gamma \geq 0$ are weighting coefficients.

E Additional Experimental Evaluation for Section 3

These experiments are intended as proof-of-concept demonstrations of the framework rather than large-scale benchmarks.

Algorithm 3 `BPTT`: single BPTT step (shared score model) – lines 6-7 in Algorithm 1.

Require: shared score model $S(\cdot, \cdot; \theta)$ (frozen); control agents $\{u^i(\cdot, \cdot; \phi^i)\}_{i=1}^N$; aggregator φ; SDE with forward drift $f(\cdot, t)$, marginal std. $\sigma(t)$, diffusion coeff. $g(t)$; optimality criterion with running and terminal losses c, Ψ; target label $y^\star$; batch size B; number of steps $K \geq 2$; control regularisation λ; running-cost scaling $\alpha > 0$;

function `BPTT`($\{\phi^i\}_{i=1}^N$)
  Sample time grid $(t_0, \ldots, t_{K-1})$ linearly from 1 to ε
  $\ell_u \leftarrow 0, \quad \ell_c \leftarrow 0$
  $\sigma_0 \leftarrow \sigma(t_0)$
  for $i = 1$ **to** N **do** ▷ Initialization
    $X^i_{t_0} \sim \mathcal{N}(0, \sigma_0^2 I)$
  end for
  for $k = 0$ **to** $K - 2$ **do**
    $\Delta t \leftarrow t_k - t_{k+1}$ ▷ $\Delta t > 0$
    $g_k \leftarrow g(t_k)$
    $Y_{t_k} \leftarrow \varphi(\{X^i_{t_k}\}_{i=1}^N)$
    for $i = 1$ **to** N **do**
      $s^i_k \leftarrow S(X^i_{t_k}, t_k; \theta)$
      $\widehat{X}^i_{0,k} \leftarrow \texttt{Tweedie}(X^i_{t_k}, t_k, s^i_k)$ ▷ Tweedie look-ahead for running loss
    end for
    $\widehat{Y}_{0,k} \leftarrow \varphi(\{\widehat{X}^i_{0,k}\}_{i=1}^N)$
    $\ell_c \leftarrow \ell_c + c(\widehat{Y}_{0,k}, y^\star)\Delta t$
    for $i = 1$ **to** N **do**
      $\mathtt{g}^i_k \leftarrow \nabla_{\widehat{X}^i_{0,k}} c(\widehat{Y}_{0,k}, y^\star)$ ▷ Reward-informed inductive bias
      $\mathtt{g}^i_k \leftarrow \texttt{stopgrad}(\mathtt{g}^i_k)$
      Control input: $z^i_k \leftarrow [\, X^i_{t_k},\, Y_{t_k},\, \mathtt{g}^i_k \,]$
      $u^i_k \leftarrow u^i(z^i_k, t_k; \phi^i)$
      Sample $\xi^i_k \sim \mathcal{N}(0, I)$ ▷ Euler–Maruyama
      Reverse drift: $\mu^i_k \leftarrow -f(X^i_{t_k}, t_k) + g_k^2\, s^i_k$
      $X^i_{t_{k+1}} \leftarrow X^i_{t_k} + \big(\mu^i_k + g_k\, u^i_k\big)\Delta t + g_k \sqrt{\Delta t}\, \xi^i_k$
      $\ell_u \leftarrow \ell_u + \frac{1}{N}\|u^i_k\|_2^2\, \Delta t$
    end for
  end for
  $Y_{t_{K-1}} \leftarrow \varphi(\{X^i_{t_{K-1}}\}_{i=1}^N)$
  $\ell_\Psi \leftarrow \Psi(Y_{t_{K-1}}, y^\star)$
  $\widehat{\mathcal{J}} \leftarrow \lambda\, \ell_u \;+\; \ell_\Psi \;+\; \alpha\, \ell_c$
  return $\widehat{\mathcal{J}}$
end function

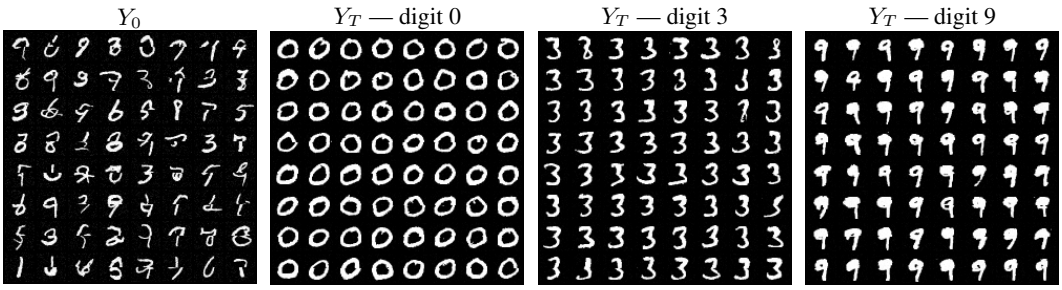

Figure 3: Two Agents (joint): Aggregated state in a two-agent compositional diffusion setup with non-overlapping masking. Agent 1 and Agent 2 control the upper and lower halves of the image, respectively (see Figure 2). The initial aggregated state, shown prior to optimisation, reveals the explicit split between the two components, highlighting how cooperative control progressively aligns independently generated trajectories into a unified global structure.

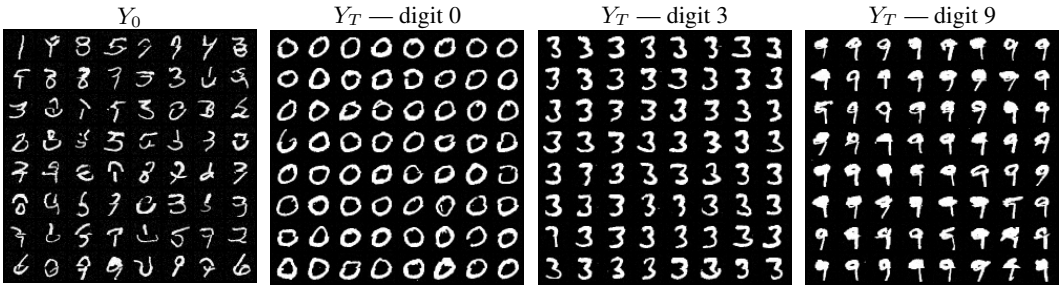

Figure 4: Two Agents (control-wise): Aggregated state in a two-agent compositional diffusion setup with non-overlapping masking. After 300 iterations of control-wise optimisation, the terminal reverse-time diffusion sample exhibits a semantically coherent digit emerging from coordinated agent dynamics.

The experimental evaluation is conducted as follows. We first train a score-based diffusion model on the MNIST dataset for 100 epochs, until convergence. The score network is parameterised by a resized U-Net architecture equipped with explicit time conditioning. The diffusion dynamics are formulated using a variance-preserving stochastic differential equation, and the model is trained to approximate the score of the perturbed data distribution at each diffusion time.

Joint optimisation of the control policies is performed using the Adam optimiser for 1000 gradient updates. In contrast, control-wise optimisation is carried out for 300 outer iterations, each comprising 5 inner gradient updates, also using the Adam optimiser with a learning rate of 1×10^{-4} in both settings.

Here we show 64 samples generated by the controlled diffusion dynamics for configurations with two (Figures 3 and 4) and three agents (Figures 5 and 6), under both joint and control-wise optimisation schemes, across three target digits (0, 3, and 9). We additionally report inference-time DPS samples in Figure 7, where the guidance term is scaled by a factor of 100. For all CMAD experiments, we use a control regularisation weight $\lambda = 10$ and a running cost scaling parameter $\alpha = 1$. All samples are generated using a 500-step Euler–Maruyama discretisation of the reverse-time diffusion process.

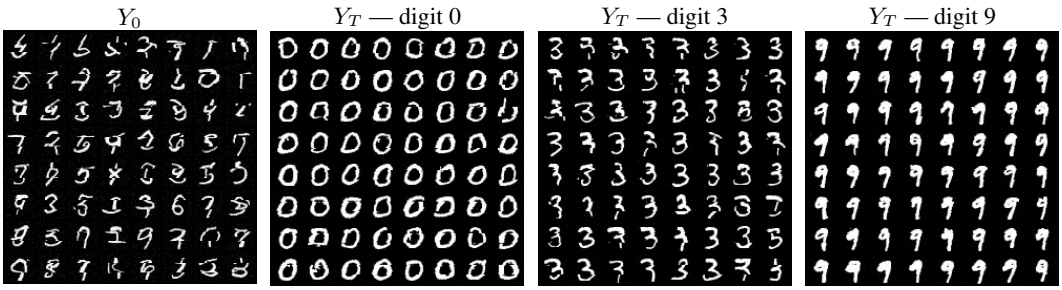

Figure 5: Three Agents (joint): Aggregated state in a three-agent compositional diffusion setup with non-overlapping masking. Agent 1, Agent 2, Agent 3 control the upper, middle, and lower halves of the image, respectively.

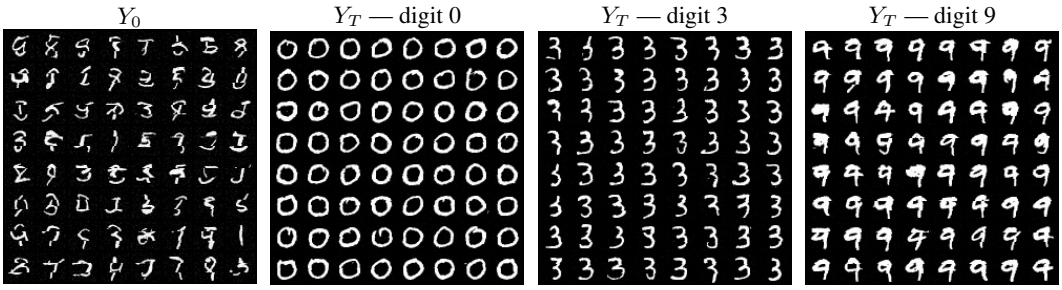

Figure 6: Three Agents (control-wise): Aggregated state in a three-agent compositional diffusion setup with non-overlapping masking. Agent 1, Agent 2 and Agent 3 control the upper, the middle, and lower halves of the image, respectively.

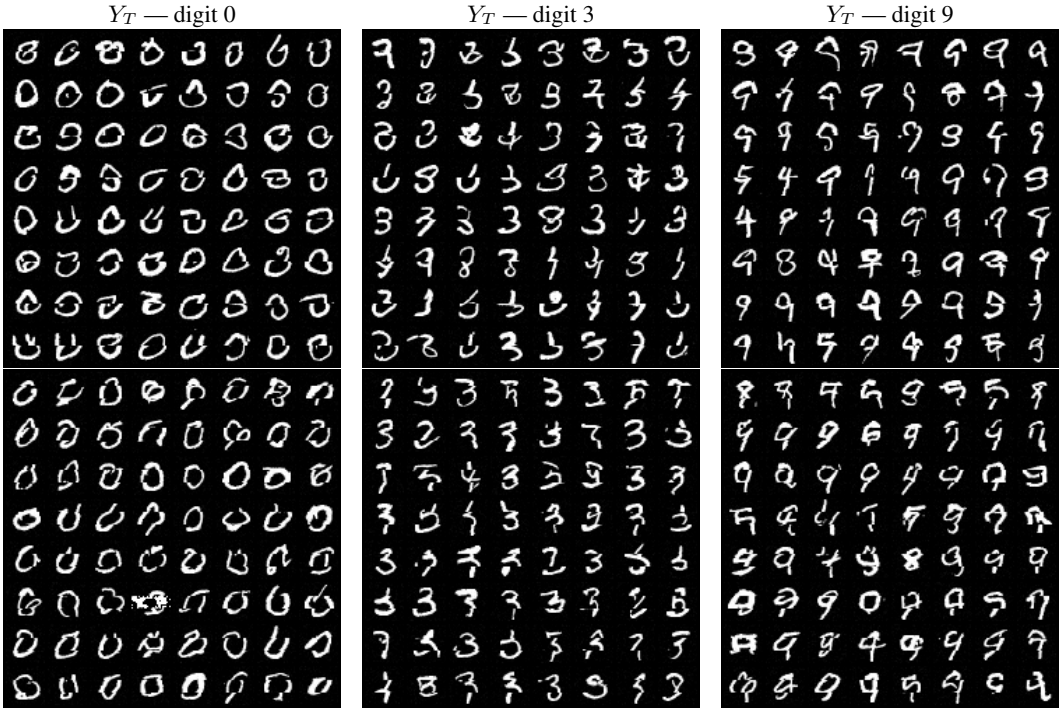

Figure 7: Inference-time CDPS composition on MNIST. Top: two-agent setup with non-overlapping aggregation. Bottom: three-agent setup. Each column corresponds to a target digit, showing how gradient guidance coordinates multiple pretrained diffusion models without learned control.

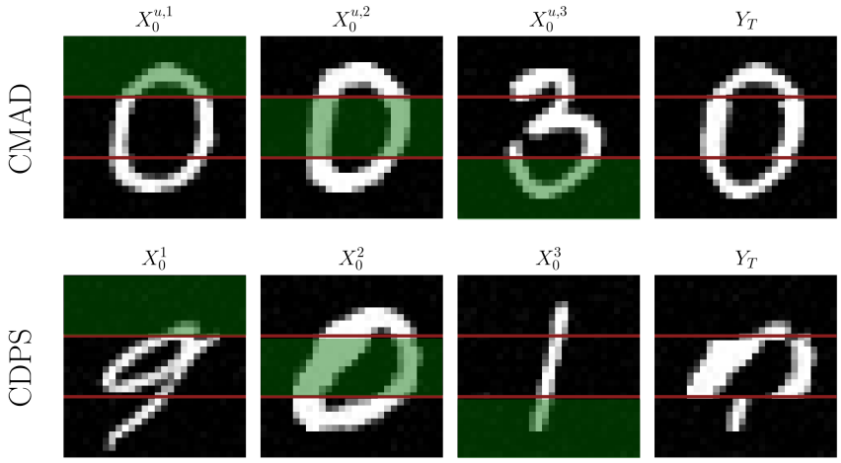

Figure 8: A single sample for CMAD and CDPS generated with 3 agents for the target 0. Every agent controls one horizontal stripe (■ coded) of the aggregated state Y_t. We show the state $X_0^{u,i}$ for every agent. Figure 6 shows multiple samples for this setting.

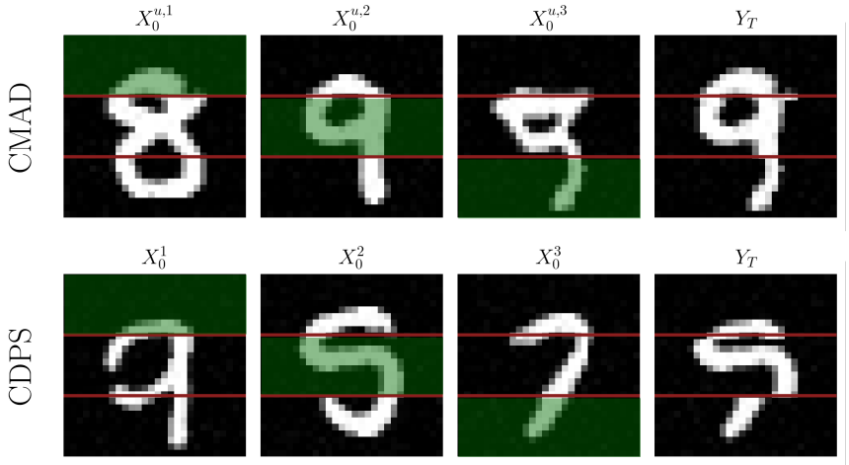

Figure 9: A single sample for CMAD and CDPS generated with 3 agents for the target 9. Every agent controls one horizontal stripe (■ coded) of the aggregated state Y_t. We show the state $X_0^{u,i}$ for every agent. Figure 6 shows multiple samples for this setting.