

WHEN DOES OBSERVATIONAL DATA TEACH LATENT DYNAMICS? UNDERSTANDING CONTROL MISALIGNMENT WITH SYNTHETIC TASKS

Anonymous authors

Paper under double-blind review

ABSTRACT

Deep generative models are increasingly embedded in applications in robotics, simulation, and image/video/audio synthesis. In these settings, data likelihoods may depend on hidden “control parameters” not directly observed at training time (e.g., speed, energy, transition rule). Although standard loss functions do not enforce distributional alignment over such hidden variables, practitioners often assume that models generate samples with controls reflecting those priors. We identify *Control Misalignment (CM)*: model generations consistently violate distributional alignment with the control prior in patterned ways across tasks and architectures, posing significant safety and fairness concerns. We first catalogue the prevalence of CM in real-world vignettes: distribution shifts of movement speed in D4RL motion planning, total energy in double-pendulum physical simulation, and speaking rate in Tacotron2 speech synthesis. Next, we probe *when*, and *why*, such drifts emerge, testing confounds against carefully constructed synthetic tasks with known controls and tunable chaoticity. Through this, we characterize the mechanism behind CM: error signatures in data space are transported through ill-conditioned or ambiguous recovery procedures into coherent control-space malapportionment. We verify this mechanistic interpretation by constructing a minimal toy system that reproduces the defining characteristics of CM. Finally, we apply our learnings to propose mitigations, analyzing which ones are theoretically sound or empirically effective versus not. Overall, our work provides a precise mechanistic account as to why, and under what conditions, Control Misalignment arises in generative pipelines.

1 INTRODUCTION

Deep learning is now widely used to train neural networks taking on various shapes, sizes, and applications. For example, they are trusted as surrogates for complex real-world processes, from robotic motion planning (Janner et al., 2022) and dynamical simulation (Li et al., 2023) to speech synthesis (Shen et al., 2018; Kong et al., 2020). In these scientific and engineering domains, the observable data x is often governed by semantically meaningful *control parameters*: examples include the speed of a trajectory, rule of a physical system, or the speaking rate of an utterance. When training is unconditional, these controls remain latent; the model sees only the marginal data distribution $q(x) = \int p(x | r)\pi(r)dr$. Standard generative objectives (e.g., NLL, ELBO, Score Matching) incentivize the model to match this data marginal $q(x)$. Critically, however, approximating the data marginal does not guarantee that the implicit distribution over latent controls is preserved. We identify this phenomenon as *Control Misalignment (CM)*, encapsulating how the distribution of controls recovered from generated samples systematically deviates from the training prior $\pi(r)$, even when the generative model appears to achieve high fidelity on standard metrics.

Why does this occur? We argue for a mechanistic explanation based on density transport through control recovery: structured observation-space errors can be low-cost under marginal objectives yet become large recovered-control shifts when recovery is sensitive or ambiguous. We test this perspective by moving from real-world vignettes to controlled synthetic environments that isolate sensitivity (tent/logistic/sinusoid) and ambiguity (folded map), and by validating the mechanism in a minimal toy system. Our contributions are as follows:

- **Characterizing CM.** We show CM in Maze2D speed, pendulum energy, and Tacotron2 speaking rate (Sec. 2.1). We then formalize Control Misalignment as divergence between recovered-control pullbacks and the intended prior (Def. 1).
- **Mechanism + validation.** We explain CM as density transport through recovery and validate it on sensitivity/ambiguity benchmarks (Sec. 2).
- **Mitigations.** We analyze iterative prior reweighting and conditioning, showing limits under ambiguity and gains with control labels (Sec. 4).

Prior work. Many works document systematic sampling biases in diffusion and autoregressive models—mode interpolation, signal leak, spectral and exposure bias—and show that aggregate metrics can hide miscalibration (Aithal et al., 2024; Everaert et al., 2024; Kadkhodaie et al., 2023; Schmidt, 2019; Meister & Cotterell, 2021). CM also connects to inverse problem theory, where sensitivity and ambiguity turn small observation errors into large parameter errors (Tikhonov & Arsenin, 1977; Tarantola, 2005; Bellman & Åström, 1970).

2 EVIDENCE AND MECHANISM OF CM

2.1 Control Misalignment in Deployed Pipelines. We first examine three deployed pipelines and consistently observe shifts in semantically meaningful latent controls. We report numerical summaries in Appx. I.

- **Maze2D diffusion planning.** We use a pretrained Diffuser-style trajectory model to generate goal-conditioned plans in D4RL Maze2D U-maze (Fu et al., 2020; Janner et al., 2022), recover trajectory speed, and resample to enforce a uniform speed prior; a model trained on this set shifts toward higher speeds (Fig. 1a).
- **Physical simulation.** We simulate an energy-conserving double pendulum with initial energy E drawn uniformly, train an unconditional diffusion model on trajectories, and recover energy by finite-differencing positions and evaluating the energy functional. While the model’s samples look plausible, their recovered energy distribution deviates from the intended uniform prior, shifting towards higher energies (Fig. 1b).
- **Tacotron2 speech synthesis.** We evaluate Tacotron2 (Shen et al., 2018) on LJ Speech (Ito & Johnson, 2017), recover speaking rate using wav2vec 2.0–based transcription (Baevski et al., 2020) and syllable counts with a transcript-consistency filter. Appx. I.3 details the filtering and syllable-counting procedure. The recovered speaking-rate marginal for generations trends faster than that of the training data (Fig. 1c).

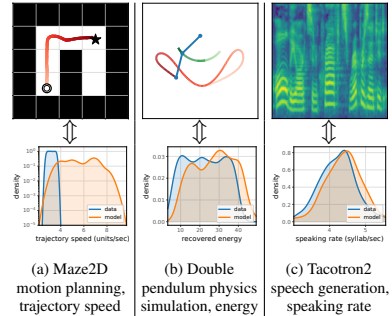


Figure 1: Perceptually reasonable generation can be significantly misaligned in control space.

2.2 Density Transport Through Recovery. The vignettes above share a common structure: the data are indexed by latent controls, but the model is trained unconditionally on the marginal over observations. Let $r \in \mathcal{R} \subseteq \mathbb{R}$ denote one such control with intended prior $\pi(r)$, and let x denote an observation. The data marginal is $q(x) = \int p(x | r)\pi(r) dr$, and an unconditional generative model learns $q_\theta(x)$. When a control can be recovered from observations, a recovery procedure maps each observation x to a distribution $\rho(r | x)$ over controls; for a source distribution S over observations, the induced recovered control marginal (pullback) is $\bar{\pi}_S(r) = \mathbb{E}_{x \sim S}[\rho(r | x)]$ (Eq. 1).

Definition 1 (Control Misalignment). Let $\bar{\pi}_{\text{data}} := \bar{\pi}_q$ and $\bar{\pi}_{\text{model}} := \bar{\pi}_{q_\theta}$ under a shared recovery procedure. We quantify CM of a model q_θ with respect to an intended prior π by the total-variation divergence $\text{TV}(\bar{\pi}_{q_\theta}, \pi) = \frac{1}{2} \int_{\mathbb{R}} |\bar{\pi}_{q_\theta}(r) - \pi(r)| dr$. We say a model exhibits CM when this divergence substantially exceeds the data baseline, i.e., when $\text{TV}(\bar{\pi}_{\text{model}}, \pi) \gg \text{TV}(\bar{\pi}_{\text{data}}, \pi)$. Even when recovery is certified on data (so $\bar{\pi}_q \approx \pi$), applying the same map to model samples can yield a different pullback. We unpack the mechanism in three steps: when CM can be large, what determines the misalignment signature, and why recovery is high-gain.

- **When can CM be large?** If ρ is L -Lipschitz in total variation, then $\text{TV}(\bar{\pi}_S, \bar{\pi}_T) \leq L W_1(S, T)$ (Eq. 5). The triangle inequality gives $\text{TV}(\bar{\pi}_{q_\theta}, \pi) \leq L W_1(q_\theta, q) + \text{TV}(\bar{\pi}_q, \pi)$, so large CM requires either a sizeable observation-space discrepancy or a high recovery gain.
- **What determines the misalignment signature?** Writing $\Delta_\theta(x) = q_\theta(x) - q(x)$ and $\tau_\theta(r) = \bar{\pi}_{q_\theta}(r) - \bar{\pi}_q(r)$, the transport identity $\tau_\theta(r) = \int \Delta_\theta(x) \rho(r | x) dx$ (Eq. 2) shows that signed model error is redistributed across controls by the recovery map, so low-cost deviations in x that align with recovery gain explain the coherent drift patterns in the vignettes.

- **What makes recovery high-gain?** Sensitivity means $\rho(\cdot | x)$ changes rapidly with x (large L), while ambiguity means multiple controls assign comparable mass to the same x , so small perturbations swap mass between competing explanations.

2.3 When and Where CM Appears. We now instantiate sensitivity and ambiguity in minimal synthetic benchmarks where the data generation process and recovery procedure are fully specified.

2.3.1 Dynamical sensitivity. We begin with dynamical systems where the forward map becomes increasingly sensitive as a function of the control. The tent map provides a minimal one-dimensional setting: $x_{t+1} = f_r(x_t) = r \cdot \min\{x_t, 1 - x_t\}$ with $r \in [0, 2]$ and $x_0 = 0.25$. Appx. D.2 defines the logistic and sinusoid comparison families. Unless stated otherwise, we use a uniform prior over r , train an unconditional diffusion model (Ho et al., 2020; Song et al., 2021) on the marginal over trajectories, and recover controls using the shared procedure in Eq. 3. The recovered control pullback on ground-truth data stays close to the baseline. By contrast, the unconditional model exhibits substantial misalignment, with the largest deviations in the chaotic regime. In the logistic-map comparison, misalignment likewise concentrates in sensitive regimes, while for the sinusoid family it is negligible; its controls remain well-separated in the MDS geometry under the same recovery interface. Quantitatively, in the baseline tent-map run, the model’s recovered-control discrepancy is $\text{TV}(\bar{\pi}_{q\theta}, \pi) \approx 0.13$ while the data baseline is $\text{TV}(\bar{\pi}_q, \pi) \approx 0.004$. Appx. F.1 and Appx. F.2 run ablations across estimators, model families, and horizons and show the qualitative pattern persists. We also supplement this understanding of sensitivity via Lyapunov exponents (rate of local trajectory separation) and MDS embeddings (where distinct controls collapse in observation space), defined in Appx. B. We show both in Fig. 2—the visualizations reveal that CM concentrates where the control-data relationship is particularly sensitive.

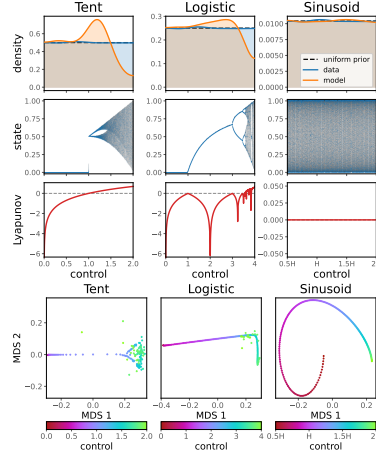


Figure 2: CM concentrates in regimes with chaotic and folded/clumped control, i.e., areas with high sensitivity.

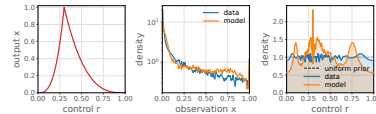


Figure 3: An asymmetric many-to-one simulator makes the inverse ambiguous, inducing CM.

2.3.2 Recovery ambiguity. To isolate ambiguity in the recovery interface, we construct a minimal one-dimensional inverse problem where r is intrinsically non-identifiable from x . Appx. H details the folded-map construction. Let $r \sim \text{Unif}[0, 1]$ and define a deterministic simulator $g : [0, 1] \rightarrow [0, 1]$ by an asymmetric fold at $c \in (0, 1)$, and observe $x = g(r) + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma_{\min}^2)$. For most x , there are two distinct controls $r \neq r'$ with $g(r) = g(r')$, so any recovery based only on x is necessarily ambiguous. Even though a single posterior $\rho(r | x)$ can have two sharp modes, the pullback on ground-truth data remains near the intended prior because recovery matches the observation model; ambiguity alone does not create CM. CM appears when a model perturbs the observation distribution, and recovery behaves like a soft competition between preimage branches, so small biases in where $q\theta$ concentrates probability can change which branch “wins” (Fig. 3).

Overall, these tasks support the mechanism from Sec. 2: recovery-side sensitivity and ambiguity localize where CM can be large, and the signature of misalignment depends on the interaction between the control-data map and the model’s preferred observation-space errors.

3 EXPLAINING THE VIGNETTES

With the mechanism and the synthetic benchmarks from Sec. 2 in hand, we can interpret each deployed vignette as *density transport through recovery*: the generator can make small, plausible observation-space deviations that become large recovered-control shifts when they align with high-gain (sensitive) or ambiguous directions of the recovery interface.

- Maze2D speed misalignment.** In Maze2D, the recovered control is a simple functional of the plan, $v(x) = \frac{1}{(H-1)\Delta t} \sum_t \|q_{t+1} - q_t\|_2$ (Appx. I.1). In Fig. 4a, we add feasible normal-direction perturbations scaled to a fixed trajectory MSE budget and vary only their roughness in time; increasingly jittery perturbations induce much larger shifts in recovered speed, illustrating how low-cost deviations can yield large control shifts. In this vignette, the resulting recovered-speed misalignment is large ($TV \approx 0.92$; Appx. I).
- Pendulum energy misalignment.** For the double pendulum, we recover energy from position sequences by finite differences and the standard energy functional (Appx. I.2). This recovery is sensitive (differentiation) and ambiguous off the simulator manifold: model samples drift in recovered energy over time, so there is no single energy value that an energy-conserving simulator could have generated. We visualize this in Fig. 4b, where a representative model trajectory violates conservation in a way that forces recovery to pick a single energy anyway; the recovered-energy discrepancy is $TV \approx 0.12$ (Appx. I).
- Tacotron2 speaking-rate misalignment.** Speaking rate is estimated as syllables-per-second after ASR-based filtering; Sec. 2.1 and Appx. I.3 detail the pipeline. This recovery is sensitive to time-axis distortions: time warps can change syllable rate without requiring large spectral changes. We construct iso-loss time-warp pairs for each utterance and find a consistent asymmetry: equal mel loss often permits substantially larger speed-ups than slow-downs (Fig. 4c), which provides a natural explanation for why an unconditional model can misalign toward faster recovered speaking rates ($TV \approx 0.13$; Appx. I).

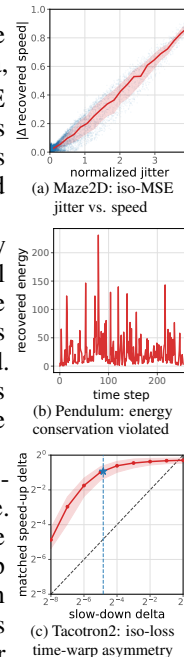


Figure 4

4 MITIGATIONS

Approximating the observation marginal can still admit small deviations that recovery amplifies into CM; so, mitigations must reduce such deviations, reshape the training mixture over controls, or make the relevant variation explicit during training. We explore several approaches below (accompanied by Fig. 5 and Appx. F).

- Improving fidelity and inductive biases.** Because Eq. 5 links observation-space error to pullback error, higher fidelity and inductive biases that suppress recovery-amplified artifacts are beneficial, but indirect; on the tent map, neither a wider model nor an autoregressive generator reduce CM.
- Iterative prior correction (IPC).** When we can sample data at chosen controls, IPC reweights the training mixture via $\pi_{\text{train}}^{(k+1)}(r) \propto \pi(r) / (\pi_{q_{\theta}^{(k)}}(r) + \varepsilon)$ and retrains. The folded-map benchmark shows IPC improves alignment but does not eliminate it, since ambiguity continues to distribute posterior mass across competing preimages.
- Conditional modeling.** If the control (or a reliable proxy) is known before training, we can train $p_{\theta}(x | r)$ and sample with $r \sim \pi$, making the variation explicit; conditioning largely restores alignment. In low-gain (near-unambiguous) settings such as the sinusoid family, unconditional models can already sit near the data baseline, so conditioning may offer little utility.

5 CONCLUSION

We identify CM, an omitted-variable failure mode of unconditional generative models: even when recovery validates on data, applying it to generations can expose shifts in the implied control mixture, so approximating $q(x)$ need not cover unobserved controls. Across deployed pipelines and synthetic benchmarks, we find a common mechanism: unconditional training makes certain structured deviations in observation space cheap, and sensitive or ambiguous recovery interfaces transport those deviations into coherent misalignment in recovered controls. Our mitigation results also suggest a practical lesson: when a salient control can be identified (or approximated by a reliable proxy), conditioning on it can make the model respect alignment while control-blind strategies cannot. More broadly, this motivates treating control discovery and measurement as a key focus of model development: without explicit controls to condition on or probe via recovery, observation-space fidelity (and capacity scaling) can mask systematic distortions and biases along salient control axes.

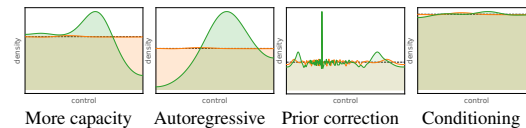


Figure 5: Mitigations; most fail to fully fix CM.

216 REFERENCES

- 217
218 Sumukh K Aithal, Pratyush Maini, Zachary C. Lipton, and J. Zico Kolter. Understanding hallucina-
219 tions in diffusion models through mode interpolation, 2024.
- 220 Alexandru Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A frame-
221 work for self-supervised learning of speech representations. In *Advances in Neural Information*
222 *Processing Systems*, volume 33, pp. 12449–12460, 2020.
- 223
224 Richard Bellman and Karl J. Åström. On structural identifiability. *Mathematical Biosciences*, 7(3–4):
225 329–339, 1970. doi: 10.1016/0025-5564(70)90132-X.
- 226
227 Martin Nicolas Everaert, Athanasios Fitsios, Marco Bocchio, Sami Arpa, Sabine Süssstrunk, and
228 Radhakrishna Achanta. Exploiting the signal-leak bias in diffusion models. In *Proceedings of the*
229 *IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 4025–4034, 2024.
- 230 Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep
231 data-driven reinforcement learning, 2020.
- 232
233 Shahin Hakemi, Naveed Akhtar, Ghulam Mubashar Hassan, and Ajmal Mian. Deeper diffusion
234 models amplify bias, 2025.
- 235
236 Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in*
237 *Neural Information Processing Systems*, 2020.
- 238
239 Keith Ito and Linda Johnson. The LJ speech dataset. [https://keithito.com/
LJ-Speech-Dataset/](https://keithito.com/LJ-Speech-Dataset/), 2017.
- 240
241 Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with diffusion for
242 flexible behavior synthesis. In *International Conference on Machine Learning*, 2022.
- 243
244 Zahra Kadkhodaie, Florentin Guth, Eero P. Simoncelli, and Stéphane Mallat. Generalization in
245 diffusion models arises from geometry-adaptive harmonic representation, 2023.
- 246
247 Jari P. Kaipio and Erkki Somersalo. *Statistical and Computational Inverse Problems*. Applied
248 Mathematical Sciences. Springer, 2005. doi: 10.1007/b138659.
- 249
250 Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. HiFi-GAN: Generative adversarial networks for
251 efficient and high fidelity speech synthesis. In *Advances in Neural Information Processing Systems*,
2020.
- 252
253 Lizao Li, Rob Carver, Ignacio Lopez-Gomez, Fei Sha, and John Anderson. Seeds: Emulation of
254 weather forecast ensembles with diffusion models. *arXiv preprint arXiv:2306.14066*, 2023.
- 255
256 Clara Meister and Ryan Cotterell. Language model evaluation beyond perplexity. In *Proceedings of*
257 *the 59th Annual Meeting of the Association for Computational Linguistics*, pp. 5328–5339, 2021.
- 258
259 Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable
260 fidelity and diversity metrics for generative models. In *International Conference on Machine*
261 *Learning*, pp. 7176–7185, 2020.
- 262
263 Nathan Roos, Ekaterina Iakovleva, Ani Gjergji, Vito Paolo Pastore, and Enzo Tartaglione. How i met
264 your bias: Investigating bias amplification in diffusion models, 2025.
- 265
266 Florian Schmidt. Generalization in generation: A closer look at exposure bias. (arXiv:1910.00292),
267 November 2019. doi: 10.48550/arXiv.1910.00292. URL [http://arxiv.org/abs/1910.](http://arxiv.org/abs/1910.00292)
268 [00292](http://arxiv.org/abs/1910.00292). arXiv:1910.00292 [cs].
- 269
Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng
Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. Natural tts synthesis by conditioning
wavenet on mel spectrogram predictions. In *2018 IEEE international conference on acoustics,*
speech and signal processing (ICASSP), pp. 4779–4783. IEEE, 2018.

270 Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben
271 Poole. Score-based generative modeling through stochastic differential equations. In *International*
272 *Conference on Learning Representations*, 2021.
273
274 Albert Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. Society for
275 Industrial and Applied Mathematics, 2005. doi: 10.1137/1.9780898717921.
276
277 Andrei N. Tikhonov and Vasilii Y. Arsenin. *Solutions of Ill-Posed Problems*. Winston, Washington,
278 D.C., 1977.
279
280 Warren S. Torgerson. Multidimensional scaling: I. Theory and method. *Psychometrika*, 17(4):
281 401–419, 1952. doi: 10.1007/BF02288916.
282
283 Soobin Um, Suhyeon Lee, and Jong Chul Ye. Don’t play favorites: Minority guidance for diffusion
284 models, 2023.
285
286 Binxu Wang and Cengiz Pehlevan. An analytical theory of spectral bias in the learning dynamics of
287 diffusion models, 2025.
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323

A ADDITIONAL RELATED WORK

A growing body of work reveals systematic sampling biases across model families. For example, diffusion models place excess probability mass along low-density bridges between modes (Aithal et al., 2024), exhibit signal-leak artifacts from prior mismatch (Everaert et al., 2024), and preferentially learn smooth structure due to spectral bias (Wang & Pehlevan, 2025; Kadkhodaie et al., 2023). Recent work also studies bias and its amplification in diffusion models, and proposes guidance mechanisms to improve coverage of underrepresented groups (Um et al., 2023; Hakemi et al., 2025; Roos et al., 2025). Autoregressive models suffer from exposure bias—a mismatch between teacher-forced training on ground-truth sequences and free-running inference on the model’s own predictions, causing errors to compound quadratically with sequence length (Schmidt, 2019). These biases often evade detection because standard metrics such as likelihood, inception score, perplexity, and FID measure aggregate fidelity rather than calibration across the full support of a latent variable. A model can therefore appear well-trained while harboring systematic distortions (Naeem et al., 2020; Meister & Cotterell, 2021). In some sense, these data-space artifacts can be viewed as each architecture’s preferred method of spending an error budget under a marginal objective.

Separately, the premise of CM—involving a hidden variable that dictates the data generation process—is rooted in classical inverse problem theory (Tikhonov & Arsenin, 1977; Tarantola, 2005; Kaipio & Somersalo, 2005). When an inverse map is sensitive or ambiguous, small observation-space errors propagate into large parameter-space errors. This instability is well-studied in dynamical systems parameter estimation (Bellman & Åström, 1970), where structural identifiability questions arise even with noiseless data. The crucial difference here is that the “perturbation” is not random noise, but the structured bias of a learned distribution of deep generative models, visible only through inversion.

B SUPPLEMENTARY DEFINITIONS

This appendix records definitions and the shared recovery interface used throughout the main text and appendices.

Throughout, a recovery procedure maps each observation x to a distribution $\rho(r | x)$ over controls. Recovery can be probabilistic (posterior-like) or deterministic, in which case $\rho(r | x)$ is a point mass at a recovered control $\hat{r}(x)$.

For any source distribution S over observations, the recovered control marginal (pullback) is

$$\bar{\pi}_S(r) = \mathbb{E}_{x \sim S}[\rho(r | x)]. \quad (1)$$

Writing $\Delta_\theta(x) = q_\theta(x) - q(x)$ and $\tau_\theta(r) = \bar{\pi}_{q_\theta}(r) - \bar{\pi}_q(r)$, the density-transport identity is

$$\tau_\theta(r) = \int \Delta_\theta(x) \rho(r | x) dx. \quad (2)$$

In most synthetic benchmarks, a forward simulator $g : \mathcal{R} \rightarrow \mathcal{X}$ is available and recovery is implemented by comparing observations to simulator references. In principle, one could operationalize recovery by Bayes' rule, $\rho(r | x) \propto p(x | r)\pi(r)$, but in many settings the forward map is (approximately) deterministic: $x = g(r)$. Without an explicit noise model, the corresponding likelihood is degenerate, $p(x | r) = \delta(x - g(r))$, so the posterior collapses to exact (or nearest-neighbor) matching and becomes brittle to discretization and to the fact that model samples are not exact simulator outputs. We therefore adopt an observation model that treats deviations from the simulator as noise, $x = g(r) + \varepsilon$, and use the resulting likelihood to define a stable, shared recovery procedure across data and model samples. Concretely, on a fixed control grid $\{r_j\}_{j=1}^J$ we compare x to simulated references $g(r_j)$ and convert discrepancies into an adaptive Gaussian likelihood:

$$\begin{aligned} e_j(x) &= \|x - g(r_j)\|_2^2, \\ \rho(r_j | x) &\propto \exp\left(-\frac{e_j(x)}{2\sigma^2(x)}\right) \pi(r_j), \\ \sigma^2(x) &= \max\left(\min_j e_j(x), \sigma_{\min}^2\right). \end{aligned} \quad (3)$$

Here $\sigma_{\min} > 0$ is a numerical floor and $g(r)$ denotes the simulator output used as a recovery reference (e.g., a fixed initial-condition rollout for dynamical systems, or a point on a geometric curve). Sec. D gives implementation details for the deployed vignettes and synthetic benchmarks.

We also record the stability assumption used in the mechanism section and the resulting bound:

$$\text{TV}(\rho(\cdot | x), \rho(\cdot | x')) \leq L \|x - x'\|_2, \quad (4)$$

$$\text{TV}(\bar{\pi}_S, \bar{\pi}_T) \leq L W_1(S, T). \quad (5)$$

Lyapunov exponent (sensitivity). The (maximal) Lyapunov exponent measures the exponential rate at which nearby trajectories separate under a dynamical system. Positive exponents indicate sensitivity to initial conditions; in the tent-map experiments, we estimate it to localize high-gain regimes where small control changes lead to large observation changes.

Multidimensional scaling (MDS). MDS (Torgerson, 1952) embeds points into a low-dimensional Euclidean space while approximately preserving pairwise distances from a given distance matrix. We use it to visualize the control-geometry induced by observation-space distances, revealing where distinct controls become close in x -space.

C TEMPLATE FOR MECHANISTIC ANALYSIS

Real pipelines mix together many effects such as learning dynamics, measurement noise, recovery heuristics, and inference-time procedures. To isolate the mechanism, we build synthetic benchmarks that keep the structure of interest—a control-indexed data family together with a recovery procedure—while stripping away most confounds. Concretely, the benchmark makes three ingredients explicit: an intended prior $\pi(r)$, a simulator that generates $x \sim p(x | r)$, and a fixed recovery procedure that can be validated on ground-truth data. We then apply the *same* recovery procedure to both data and model samples. Under this shared interface, differences in recovered-control pullbacks reflect the generator, recovery procedure, and the prior over r .

The evaluation template, depicted in Fig. 6, is:

1. Sample a control $r \sim \pi(r)$ and generate $x \sim p(x | r)$.
2. Train an unconditional model $q_\theta(x) \approx q(x) = \int p(x | r)\pi(r) dr$.
3. Recover a distribution $\rho(r | x)$ under the shared interface in Eq. 3.
4. Compute pullbacks $\bar{\pi}_q$ and $\bar{\pi}_{q_\theta}$ via Eq. 1 and report CM via Def. 1.

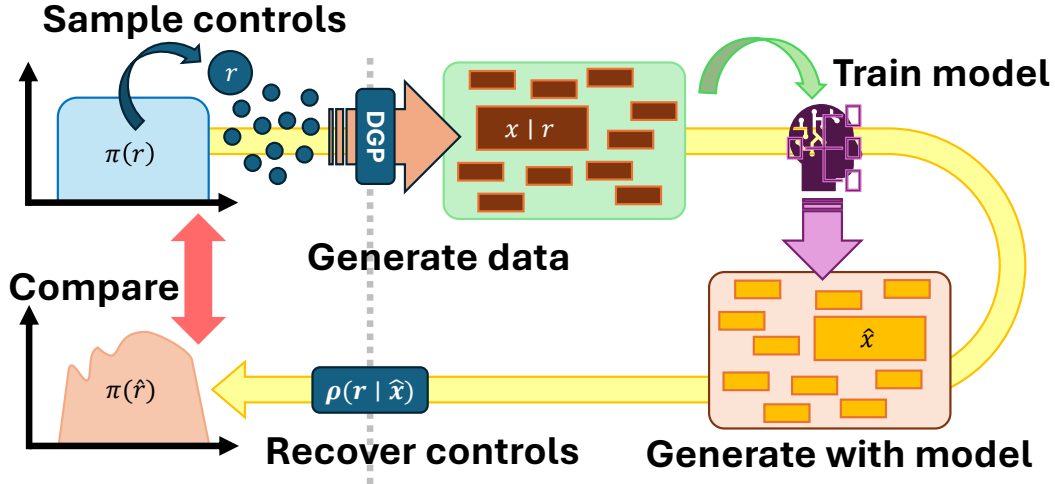


Figure 6: **Synthetic framework to study the mechanism of CM.** The steps are: sample r values from a prior distribution; use the data-generation process (DGP) to obtain observations following those r values; train a model only on observations; generate samples with the model; recover what controls correspond to each generated sample; and compare the implied control distribution to the intended prior. This template admits modularity of parts like the prior, simulator, model, and recovery procedure.

Note that for most experiments, we prevent the shape of the prior from becoming a separate confound by using a uniform prior over control parameters. In principle, this can be swapped for other priors; however, it is not necessary to test this directly, as such pipelines can be reparameterized by pushing the distribution into the generator definition to define a new generator with a uniform control prior. Concretely, if there is a transport map $T : [0, 1]^d \rightarrow \mathcal{R}$ such that $T_{\#}\text{Unif}([0, 1]^d) = \pi$, then writing $r = T(u)$ with $u \sim \text{Unif}([0, 1]^d)$ yields the same marginal

$$q(x) = \int p(x | r) \pi(r) dr = \int_{[0,1]^d} p(x | T(u)) du. \quad (6)$$

In the scalar case, $T(u) = F_\pi^{-1}(u)$ is the usual CDF transform.

486 D SETUP AND SHARED PROTOCOLS

487 D.1 MODEL ARCHITECTURES AND TRAINING DETAILS

488 **Overview.** This section documents the generator architectures and training hyperparameters used
 489 for results reported in the main text and appendix ablations. When multiple variants appear (e.g., a
 490 wider denoiser, a conditional model, or an autoregressive baseline), we list the concrete parameter
 491 choices in the tent-map hypothesis ladder (Sec. F) and in the deployed vignettes (Sec. I).

492 **Primary synthetic generator: JannerUNet1d diffusion.** For the tent/logistic/sinusoid dynamical
 493 families (and for most diffusion variants in the hypothesis ladder), we use a 1D U-Net denoiser
 494 adapted from the Diffuser-style architecture of Janner et al. (2022) via a lightweight fork of an
 495 open-source diffusion implementation. The denoiser takes a length- H trajectory and predicts the
 496 diffusion noise (or x_0 for the objective ablation), and diffusion sampling is performed with either
 497 DDPM or DDIM. Unless stated otherwise, the synthetic diffusion setup is:

- 501 • **Data representation:** trajectories $x_{0:H} \in [0, 1]^H$ are linearly mapped to $[-1, 1]$ for diffusion
 502 training; samples are mapped back to $[0, 1]$ for recovery and plotting.
- 503 • **Architecture (denoiser):** a Diffuser-style 1D U-Net with **input channels 1**, **base width 32**,
 504 **channel multipliers (1, 2, 2, 2)**, **kernel size 3**, **timestep embedding dim 32** (positional), **activa-**
 505 **tion Mish**, and **normalization** (groupnorm-style) throughout; no attention blocks are enabled in
 506 the baseline.
- 507 • **Optimizer:** AdamW with learning rate $3 \cdot 10^{-4}$, weight decay 10^{-5} , batch size 1024.
- 508 • **Training length:** 20 epochs in the full tent-map preset (and scaled proportionally in shorter
 509 presets used for quick runs).
- 510 • **Sampling:** DDPM with 256 sampling steps (unless an ablation specifies DDIM with fewer steps).

511 **Control-conditioned diffusion (conditioning mitigation).** For the control-conditioning mitigation,
 512 we keep the same diffusion backbone but inject a control embedding into the denoiser:

- 513 • **Control embedding:** r is mapped to $u \in [0, 1]$ using the known control range and encoded with
 514 a sinusoidal feature map of dimension 128.
- 515 • **Injection:** FiLM-style scale/shift modulation in residual blocks and feature-channel concatenation
 516 of a projected control embedding (concatenation width 64).
- 517 • **Backbone changes:** base width 64 (wider U-Net), trained for 60 epochs; sampling uses DDIM.

518 **Alternative generator families (hypothesis ladder baselines).** To test architecture dependence
 519 under the same recovery interface, we also report results for:

- 520 • **MLP diffusion denoiser:** a time-input MLP denoiser with hidden widths (1088, 1088, 1088)
 521 (GELU activations) trained with the same denoising objective but without convolutional locality.
 522 This backend uses a log-linear σ schedule with $N = 256$ noise levels over $\sigma \in [5 \cdot 10^{-3}, 3]$.
- 523 • **Autoregressive transformer:** sequences are discretized into 1024 bins and modeled token-by-
 524 token with a causal Transformer with $d_{\text{model}} = 216$, 4 layers, 6 heads, dropout 0.1, GELU MLP
 525 blocks, and tied input/output embeddings. We train with AdamW (learning rate $3 \cdot 10^{-4}$, weight
 526 decay 10^{-2}) with early stopping on a 2% validation split (patience 2); sampling uses temperature
 527 1.0 with top- k truncation ($k = 64$).
- 528 • **VAE:** a simple MLP VAE on the flattened trajectory with latent dimension 1, hidden widths
 529 (1088, 1088), and a β -VAE objective ($\beta = 1$ with KL warmup over 5 epochs). Samples are
 530 generated by drawing $z \sim \mathcal{N}(0, I)$ and decoding.

531 **Ambiguity benchmark generator (folded-map IPC).** For the ambiguity benchmark in Sec. 2.3.2
 532 and Sec. H, we use a deliberately minimal 1D diffusion generator trained on scalars $x \in [0, 1]$
 533 produced by the folded forward map:

- 534 • **Denoiser:** an MLP $\epsilon_{\theta}(x_t, t)$ with hidden size 128 and 3 hidden layers (SiLU activations).
- 535 • **Diffusion:** $T = 100$ steps with a linear β schedule from 10^{-4} to $2 \cdot 10^{-2}$; trained for 15,000
 536 optimization steps with Adam (learning rate $2 \cdot 10^{-4}$) and EMA decay 0.999.

540 **Deployed vignettes: Maze2D and double pendulum.** For Maze2D planning, we use a goal-
 541 conditioned diffusion planner for inpainting and train a separate diffusion model from scratch on a
 542 speed-uniform resampled dataset:

- 543 • **Pretrained proposal model:** a publicly available pretrained Maze2D trajectory diffusion model,
 544 used only to generate an initial pool for speed-uniform resampling.
- 545 • **Uniform-speed scratch model (used for CPM evaluation):** a 1D U-Net diffusion model (block
 546 channels (32, 128, 256); Mish; group normalization) trained with a 64-step diffusion schedule
 547 to predict the denoised sample. Training uses batch size 4096, learning rate 10^{-4} , 25,000
 548 optimization steps, and EMA decay 0.999.
- 549 • **Double pendulum diffusion:** a Diffuser-style 1D U-Net with input channels 2 (angles (q_1, q_2)),
 550 horizon 256, and the same baseline width/multipliers as above; trained for 50 epochs with batch
 551 size 64 and learning rate 10^{-3} .

552
 553 **Deployed vignette: Tacotron2.** For the speech vignette, we do not retrain the generator; we use a
 554 publicly available pretrained Tacotron2 model with a neural vocoder for synthesis, and we apply the
 555 speaking-rate recovery protocol described in Sec. 2.1 and Sec. I.3.

556 D.2 ADDITIONAL DETAILS

557
 558 **Synthetic data generation (defaults).** Unless stated otherwise (e.g., horizon-length or system-
 559 family ablations), we use:

- 560 • **Initial condition:** fixed $x_0 = 0.25$ for the dynamical families (tent/logistic/sinusoid), matching
 561 the main text.
- 562 • **Prior:** uniform $\pi(r)$ over the system’s stated control range.
- 563 • **Numerical stabilization:** for chaotic iterated maps, we discretize intermediate states onto a fixed
 564 grid (1024 bins on $[0, 1]$) to limit floating-point accumulation in long rollouts; this discretization
 565 is applied equally wherever the simulator is used (data generation and simulator references for
 566 recovery).

567
 568 We revisit how these defaults interact with dynamical sensitivity (e.g., high-chaos regimes) in Sec. F.

569
 570 **Recovery interface (simulator-backed posterior).** When a forward simulator is available, recovery
 571 is implemented on a fixed control grid $\{r_j\}_{j=1}^J$ spanning \mathcal{R} . For each r_j we precompute the
 572 simulator reference $g(r_j)$, and for a given observation x we define the squared discrepancy $e_j(x) =$
 573 $\|x - g(r_j)\|_2^2$ (trajectory ℓ_2 , unless an estimator ablation specifies a different norm). We then form a
 574 discrete posterior over the grid using the adaptive Gaussian likelihood from the main text (Eq. 3),
 575 and we report both: (i) *point estimates* (posterior mode or posterior mean, depending on the ablation
 576 tag), and (ii) *mixture pullbacks* obtained by averaging posteriors over samples (used for the primary
 577 pullback plots and TV numbers). This same recovery interface is the shared measurement layer for
 578 the hypothesis ladder (Sec. F) and is the mechanism by which ambiguity can amplify CM in the
 579 folded-map benchmark (Sec. H). A stability view of this mapping (and why sensitivity/ambiguity
 580 create high-gain regimes) is given in Sec. E.

581
 582 **Control misalignment metric.** We quantify CM by total variation between the recovered-control
 583 pullback and the intended prior, $\text{TV}(\bar{\pi}_S, \pi)$. For grid-based recovery this is computed on the same
 584 discretization used to represent $\bar{\pi}_S$ on $\{r_j\}$, and we report both the data baseline and the model value
 585 for each experiment. For deployed vignettes where recovery uses proxies (e.g., finite differencing,
 586 ASR-based heuristics), we follow the same shared-recovery convention and report the same family of
 587 summary statistics; see Sec. I.

588
 589 **Default synthetic generator and sampling.** Unless an ablation changes the model family, we use
 590 an unconditional diffusion generator (Sec. D.1) and sample with 256 DDPM steps; sampler ablations
 591 use DDIM with fewer steps. The concrete ablations (and the corresponding recovered-control
 592 pullbacks) are described in Sec. F.

E MECHANISM: PROOF DETAILS

This section provides proof details for the quantitative stability statement in Sec. 2. Let $\rho(\cdot | x)$ be a recovery distribution on \mathcal{R} given an observation x , and for any source distribution S over observations recall the pullback $\bar{\pi}_S = \int \rho(\cdot | x) S(dx)$.

A general Wasserstein-1 to total-variation bound. Assume there exists a constant L such that for all x, x' ,

$$\text{TV}(\rho(\cdot | x), \rho(\cdot | x')) \leq L \|x - x'\|_2,$$

as in Eq. 4. Equivalently, one may take $L := \sup_{x \neq x'} \text{TV}(\rho(\cdot | x), \rho(\cdot | x')) / \|x - x'\|_2$ over the region of interest (so L can be large when recovery is highly sensitive). Then for any two source distributions S, T over observations,

$$\text{TV}(\bar{\pi}_S, \bar{\pi}_T) \leq L W_1(S, T),$$

as in Eq. 5. *Proof.* Let γ be any coupling of S and T , with $(X, Y) \sim \gamma$. Using $\text{TV}(\mu, \nu) = \frac{1}{2} \|\mu - \nu\|_1$ and convexity of $\|\cdot\|_1$,

$$\begin{aligned} \text{TV}(\bar{\pi}_S, \bar{\pi}_T) &= \text{TV}(\mathbb{E}[\rho(\cdot | X)], \mathbb{E}[\rho(\cdot | Y)]) \\ &\leq \mathbb{E}[\text{TV}(\rho(\cdot | X), \rho(\cdot | Y))] \\ &\leq L \mathbb{E}\|X - Y\|_2. \end{aligned}$$

Taking the infimum over couplings γ yields $W_1(S, T)$.

Adaptive Gaussian recovery: sensitivity scaling. For the adaptive Gaussian recovery rule in Eq. 3, the map $x \mapsto \rho(\cdot | x)$ is locally TV-Lipschitz away from a set of measure-zero tie points where the nearest reference changes, with a bound scaling inversely with the local bandwidth $\sigma(x)$. This supports the main-text statement that recovery sensitivity increases as $\sigma(x)$ decreases.

Ambiguity factor in two-way competitions. When recovery is dominated by two competing explanations with log-weights $\ell_a(x), \ell_b(x)$, define $p(x) = \exp(\ell_a(x)) / (\exp(\ell_a(x)) + \exp(\ell_b(x)))$. Then

$$\nabla_x p(x) = p(x)(1 - p(x)) \nabla_x (\ell_a(x) - \ell_b(x)),$$

so near-ties ($p \approx \frac{1}{2}$) are precisely where recovered mass swaps most readily between the two explanations.

F SYNTHETIC BENCHMARKS: ABLATIONS

We test whether the qualitative claims in the main text persist under changes to recovery estimation, training construction, optimization/sampling, and model family/inductive bias, while holding the recovery interface fixed.

F.1 NUMERICAL SUMMARY AND PULLBACKS

Baseline tent-map setting. Unless stated otherwise, the hypothesis ladder uses the tent map family with control range $r \in [0, 2]$ and horizon $H = 64$, with an intended uniform control prior $\pi(r)$. Each run trains on $n_{\text{train}} = 200,000$ trajectories and evaluates on $n_{\text{eval}} = 25,000$ trajectories under a shared recovery interface: a fixed control grid of size $J = 16,384$, an adaptive Gaussian likelihood with an ℓ_2 discrepancy inside the exponent, and a posterior-mode point estimate used only for sample-level diagnostics. The baseline generator is an unconditional JannerUNet1d diffusion model with base width 32, kernel size 3, channel multipliers (1, 2, 2, 2), trained for 20 epochs with batch size 1024, AdamW learning rate $3 \cdot 10^{-4}$ and EMA 0.999, and sampled with DDPM using 256 diffusion steps.

Numerics (TV of recovered-control pullbacks). We report total variation between the recovered-control pullback and the intended prior for both the data baseline and the trained model. In the baseline tent-map run, $\text{TV}_{\text{data}} = 0.004$ while the unconditional model has $\text{TV}_{\text{model}} = 0.130$ for the posterior-mixture pullback (and $\text{TV}_{\text{model}} = 0.065$ under the point-estimator pullback).

F.2 INVESTIGATING POTENTIAL CAUSES OF CM ON THE TENT MAP

Hyp.	Experiment	System	H	TV data	TV model (mix)	TV model (pt)
H0	Baseline (tent map, diffusion)	tent	64	0.004	0.130	0.065
H1a	Estimator: ℓ_1 discrepancy, posterior mode (point)	tent	64	0.004	0.122	0.041
H1b	Estimator: ℓ_2 discrepancy, posterior mean (point)	tent	64	0.004	0.128	0.163
H1c	Estimator: ℓ_1 discrepancy, posterior mean (point)	tent	64	0.004	0.122	0.205
H2	Prior coverage: stratified control grid	tent	64	0.000	0.126	0.059
H3a	Optimization: short training	tent	64	0.004	0.129	0.074
H3b	Optimization: long training	tent	64	0.004	0.122	0.048
H4	Capacity: wider denoiser	tent	64	0.004	0.125	0.055
H5	Inductive bias: larger convolution kernel	tent	64	0.004	0.127	0.061
H6a	Model family: MLP denoiser	tent	64	0.004	0.309	0.377
H6b	Model family: autoregressive transformer	tent	64	0.004	0.261	0.275
H6c	Model family: VAE baseline	tent	64	0.004	0.253	0.321
H7	Sampler: DDIM (fewer steps)	tent	64	0.004	0.126	0.056
H8	Objective: x_0 -prediction	tent	64	0.004	0.126	0.060
H9a	Context: shorter horizon	tent	32	0.004	0.115	0.024
H9b	Context: longer horizon	tent	128	0.004	0.131	0.096
H10a	System: logistic map	logistic	64	0.004	0.053	0.022
H10b	System: sinusoid family	sinusoid	64	0.004	0.002	0.003
H11a	Sensitivity: low-chaos regime	tent	64	0.004	0.044	0.047
H11b	Sensitivity: high-chaos regime	tent	64	0.004	0.209	0.213
H12	Mitigation: control-label conditioning	tent	64	0.004	0.008	0.007
H13	Mitigation: importance reweighting (post-hoc)	tent	64	0.002	0.093	0.061

Table 1: **Tent-map hypothesis ladder (full numerics).** Each row is one ablation under a shared recovery interface. TV is computed from posterior-mixture pullbacks; “pt” reports TV for a point-estimator pullback when available.

× **Hypothesis H1:** CM is an artifact of a poorly specified recovery estimator.

We hold the generator fixed and vary recovery estimation choices that could plausibly inject systematic bias: we swap the ℓ_2 discrepancy inside the recovery likelihood for ℓ_1 (H1a), and for sample-level rollout diagnostics we swap the point estimate extracted from the posterior between its mode and posterior mean (H1b/H1c). These changes do not remove the recovered-control shift: H1a yields $\text{TV}_{\text{model}} = 0.122$ (mixture), H1b yields $\text{TV}_{\text{model}} = 0.128$, and H1c yields $\text{TV}_{\text{model}} = 0.122$, with the data baseline remaining at $\text{TV}_{\text{data}} = 0.004$. We therefore reject H1 (Fig. 7).

× **Hypotheses H2–H3:** CM is driven by insufficient exposure to some controls or by under-training.

We test whether CM disappears when we (i) rebalance control coverage explicitly by stratifying the training mixture over r (H2), or (ii) vary training duration (H3a/H3b). Neither change restores recovered-control coverage under the same recovery interface: $TV_{\text{model}} = 0.126$ for H2, $TV_{\text{model}} = 0.129$ for short training (H3a), and $TV_{\text{model}} = 0.122$ for long training (H3b), with $TV_{\text{data}} = 0.004$ throughout. We therefore reject H2–H3.

× **Hypothesis H4:** *CM is a capacity bottleneck.*

Increasing denoiser width (H4) does not remove CM under the same pipeline and recovery: $TV_{\text{model}} = 0.125$ (vs. $TV_{\text{data}} = 0.004$). While larger models often reduce observation-space error, these results suggest that higher capacity alone does not guarantee the *control* distribution implied by the fixed recovery procedure. We therefore reject H4 as a sufficient explanation.

× **Hypothesis H5:** *CM is a locality artifact of the convolutional denoiser.*

We increase the convolution kernel size (H5) to change the inductive bias toward local structure. This does not eliminate misalignment: $TV_{\text{model}} = 0.127$. We therefore reject H5.

× **Hypothesis H6:** *CM is diffusion-specific; other unconditional model families should avoid it.*

We replace the diffusion denoiser with alternative unconditional families under the same recovery interface: an MLP diffusion denoiser (H6a), an autoregressive Transformer (H6b), and a VAE baseline (H6c). None of these family changes reliably removes CM: $TV_{\text{model}} = 0.309$ (H6a), $TV_{\text{model}} = 0.261$ (H6b), and $TV_{\text{model}} = 0.253$ (H6c); the autoregressive baseline is particularly poor in this setting (Fig. 7). We therefore reject H6.

× **Hypotheses H7–H9:** *CM is an artifact of sampler choice, objective parameterization, or observation window length.*

Changing the sampler (DDIM; H7), switching to an x_0 -prediction parameterization (H8), or varying horizon length (H9a/H9b) does not restore recovered-control coverage under the same recovery: $TV_{\text{model}} = 0.126$ (H7), $TV_{\text{model}} = 0.126$ (H8), $TV_{\text{model}} = 0.115$ (H9a), and $TV_{\text{model}} = 0.131$ (H9b). We therefore reject H7–H9.

× **Hypothesis H10:** *CM is idiosyncratic to the tent map.*

We repeat the same pipeline on related one-dimensional dynamical families. Misalignment persists in the logistic map (H10a; $TV_{\text{model}} = 0.053$), but is substantially reduced in the sinusoid family (H10b; $TV_{\text{model}} = 0.002$), where controls remain well-separated in observation space under the same recovery interface. This indicates that CM depends on the *control geometry induced by recovery* rather than on any single system. We therefore reject H10 as stated.

✓ **Hypothesis H11:** *Sensitivity amplifies CM.*

To isolate sensitivity within the tent family, we compare a low-chaos regime (H11a) to a high-chaos regime (H11b). Recovered-control misalignment is noticeably larger in the high-chaos setting: $TV_{\text{model}} = 0.044$ (H11a) versus $TV_{\text{model}} = 0.209$ (H11b). This matches the main-text mechanism: when the forward map becomes more sensitive, small structured deviations in observation space are more easily transported into a coherent recovered-control shift. We therefore support H11.

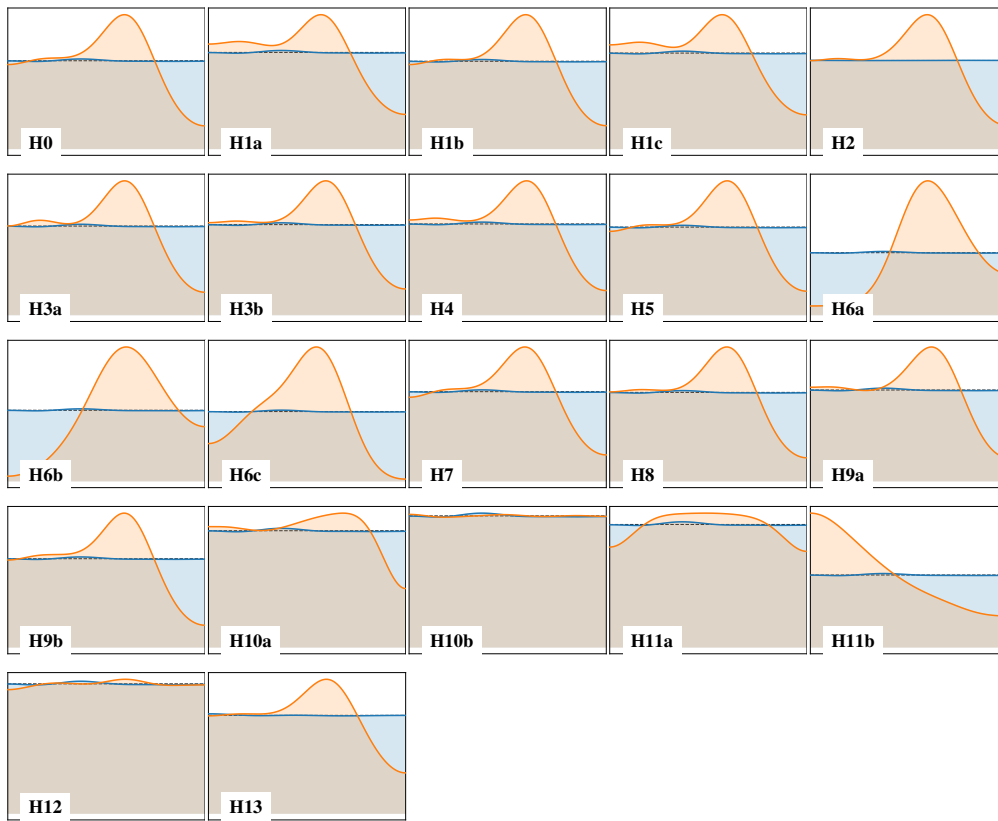
✓ **Hypothesis H12:** *When controls are known before training, conditioning restores coverage.*

Training the conditional model $p_{\theta}(x | r)$ and sampling $r \sim \pi$ at generation (H12) largely restores recovered-control alignment under the same recovery interface: $TV_{\text{model}} = 0.008$. This supports the main-text message that conditioning can bypass recovery-induced misalignment when the relevant control is available at training time.

× **Hypothesis H13:** *CM is mainly a misallocation effect that can be repaired by importance reweighting.*

We test post-hoc importance reweighting of generated samples using the recovered-control posterior (H13). This reduces but does not eliminate misalignment: $TV_{\text{model}} = 0.093$ (with $TV_{\text{data}} = 0.002$ under the same sampling estimator). This matches the fact that reweighting can only change the

756 frequency of existing outputs, not the recovery ambiguity structure induced by off-manifold samples
 757 under a fixed recovery interface. We therefore reject H13 as a general remedy.
 758



786 **Figure 7: Tent-map hypothesis ladder: recovered-control pullbacks.** Each mini is labeled by its
 787 hypothesis tag (see Tab. 1 for numerics).
 788

789
 790
 791
 792
 793
 794
 795
 796
 797
 798
 799
 800
 801
 802
 803
 804
 805
 806
 807
 808
 809

G ADDITIONAL VISUALIZATIONS: RECONSTRUCTION DIAGNOSTICS

To help interpret CM beyond aggregate pullbacks, we include reconstruction diagnostics for representative trajectories. For a trajectory x , we compute a point estimate $\hat{r}(x)$ using the experiment’s recovery estimator, and compare x to the forward rollout $g(\hat{r}(x))$ under the same horizon. For ground-truth trajectories, this serves as a sanity check that recovery produces simulator-consistent reconstructions. For model trajectories, the same reconstruction can look plausible even when the recovered-control pullback is misaligned, because a good point estimate for individual samples does not guarantee that the induced mixture over controls matches the intended prior.

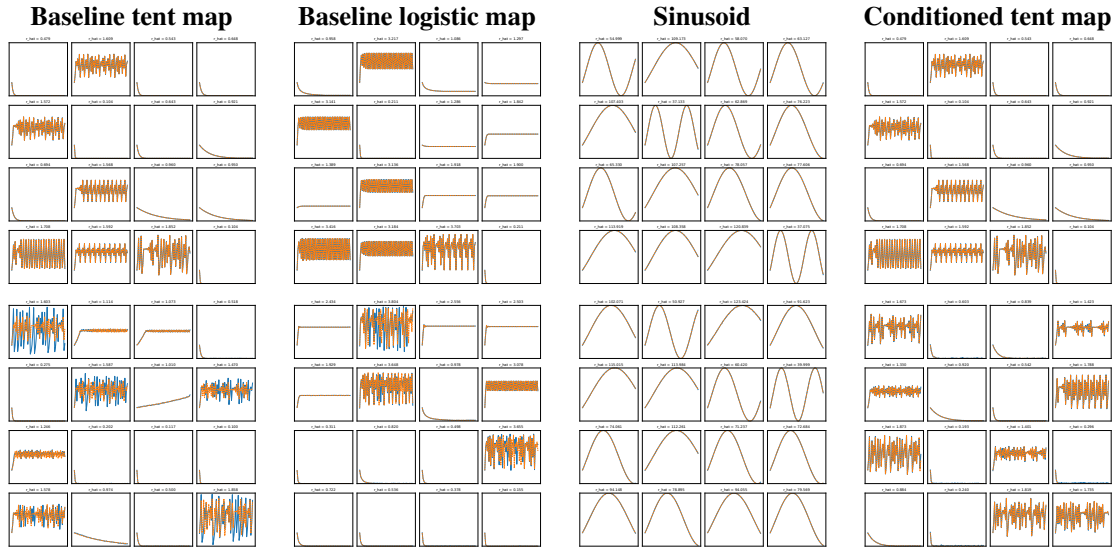


Figure 8: **Reconstruction diagnostics for synthetic benchmarks.** Top row: ground-truth trajectories and their reconstructions at $\hat{r}(x)$. Bottom row: model trajectories and their reconstructions at $\hat{r}(x)$. Even when sample-level reconstructions appear simulator-consistent, the recovered-control pullback can remain misaligned in sensitive regimes, while conditioning reduces CM.

H AMBIGUITY BENCHMARK AND ITERATIVE PRIOR CORRECTION

The tent-map ladder above largely isolates *sensitivity*. This section isolates a complementary amplifier: *ambiguity* in the recovery interface. Here, many observations are comparably compatible with multiple distant controls, so even a small amount of off-manifold observation-space mass can produce a coherent recovered-control shift.

Iterative prior correction (IPC). IPC is a training-time resampling procedure: at iteration k , train a model on a resampled training mixture, estimate its recovered-control pullback $\bar{\pi}^{(k)}$, and then reweight the next training mixture by $\pi(r)/\bar{\pi}^{(k)}(r)$ evaluated at each training example’s recovered control. In ambiguity-dominated settings, resampling can change how often certain *observations* appear, but it cannot remove the structural fact that many observations induce multi-modal recovery posteriors under the fixed recovery interface.

IPC iter	n_{train}	n_{eval}	TV (data)	TV (model)
0	60000	20000	0.022	0.119
1	60000	20000	0.022	0.084
2	60000	20000	0.022	0.069
3	60000	20000	0.022	0.062
4	60000	20000	0.022	0.059
5	60000	20000	0.022	0.057

Table 2: **IPC numerics for the ambiguity benchmark.** Per-iteration total variation of recovered-control pullbacks under the fixed recovery interface.

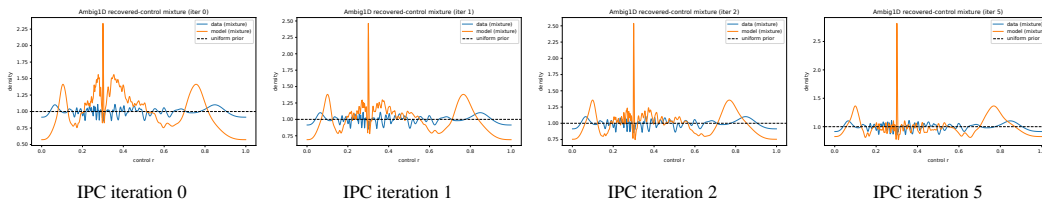


Figure 9: **IPC progression under ambiguity (folded-map benchmark).** Training-time resampling reduces recovered-control misalignment but can plateau above the data baseline, consistent with the fact that many observations remain multiply compatible with distant controls under the fixed recovery interface.

I DEPLOYED VIGNETTES: ADDITIONAL DETAILS

Numerical summary.

Vignette	Setting	N	Mean (data)	Mean (model)	TV (data vs model)
Maze2D (speed)	uniform_speed	1000	3.201	5.716	0.923
Double pendulum (energy)	energy-conserving simulator, position-only training	2000	22.599	27.467	0.118
Tacotron2 (speaking rate)	ASR-based recovery, paired WER filter	1000	4.232	4.386	0.126

Table 3: **Numerical summaries for the deployed vignettes.** We report mean recovered controls and total-variation discrepancies under the shared recovery procedures used in the main text.

I.1 MAZE2D: SETUP AND RECOVERY

Setup and recovery. We use the D4RL Maze2D U-maze dataset (Fu et al., 2020). We treat trajectory speed as an implicit control and recover it from a plan as the mean path length per unit time,

$$v(x) = \frac{1}{(H-1)\Delta t} \sum_{t=0}^{H-2} \|q_{t+1} - q_t\|_2,$$

where q_t is the position sequence and Δt is estimated from the dataset. We construct a binned-uniform speed training mixture by filtering feasible expert trajectories and resampling by recovered speed. Concretely, we use horizon $H = 128$, $\Delta t = 0.0100648$, and a uniform target speed range $v \in [2.7, 3.7]$ (chosen between dataset quantiles 30 and 75). For the recovery grid we use 20 uniform bins over the target range, map bins linearly to $r \in [0, 1]$, and use the adaptive Gaussian recovery with $\sigma_r = 0.05$ (clipped to $[0.01, 0.2]$). When reporting feasibility-based diagnostics, we use an occupancy grid of resolution 200, dilate obstacles for 2 iterations, and treat a rollout as valid if its maze-support fraction exceeds 0.95.

Iso-MSE jitter probe. Let $q_{0:H-1}$ be a valid trajectory segment and $q'_{0:H-1}$ a perturbed version with fixed endpoints ($q'_0 = q_0$ and $q'_{H-1} = q_{H-1}$). Define the displacement sequence $d_t := q'_t - q_t$ and its second difference

$$\Delta^2 d_t := d_{t+2} - 2d_{t+1} + d_t, \quad t = 0, \dots, H-3.$$

Our iso-MSE construction scales perturbations so that the trajectory MSE budget

$$\text{MSE}(q', q) := \frac{1}{H} \sum_{t=0}^{H-1} \|d_t\|_2^2$$

is fixed, and then varies only the temporal roughness of $d_{0:H-1}$. We summarize roughness by the dimensionless normalized jitter score

$$J(q', q) := \sqrt{\frac{\frac{1}{H-2} \sum_{t=0}^{H-3} \|\Delta^2 d_t\|_2^2}{\text{MSE}(q', q)}}.$$

I.2 DOUBLE PENDULUM: SETUP AND RECOVERY

Setup. We train a diffusion model on position-only trajectories and recover energy using finite differences and the standard energy functional. Model generations can violate energy conservation, so the recovered energy can drift substantially over time, and there is no unique invariant energy value for recovery to target. This contributes an ambiguity-like component in the pendulum vignette in addition to recovery sensitivity. Our diffusion model uses the JannerUNet1d backbone with base width 32, kernel size 3, channel multipliers (1, 2, 2, 2), and a 256-step diffusion schedule; we train for 50 epochs with batch size 64 and learning rate 10^{-3} on position-only trajectories, with angles normalized to a fixed range. Unless stated otherwise, quantitative summaries use $N = 2000$ samples under the same recovery.

972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025

I.3 TACOTRON2: SPEAKING-RATE RECOVERY

Recovery. We recover speaking rate by syllables-per-second after filtering based on ASR, as described in the main text (Sec. 2.1). We use the same recovery pipeline for data and model generations. Concretely, we use the LJSpeech dataset and synthesize speech with a pretrained Tacotron2 system and neural vocoder. Speaking rate is recovered using a pretrained wav2vec2 ASR model at 16kHz, over utterances with duration in $[0.3, 20]$ seconds and ≥ 2 words. We keep 1000 utterances and apply a paired word error rate filter; in the reported run, 757 paired utterances remain after filtering.