BELLMAN RECURSION DIFFUSION FOR OFF-POLICY EVALUATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Off-policy evaluation (OPE) estimates the value of an evaluation policy from data collected by a different behavior policy. A generative model of the policy's discounted future answers value queries without sequential rollouts, but training one by bootstrapping on its own predictions amplifies fitting error by up to the effective horizon $1/(1-\gamma)$. We introduce **Bellman Recursion Diffusion (BRD)**, which trains a conditional diffusion model on the discounted distribution of future transitions of a frozen one-step dynamics model. Each training target lets the frozen model simulate the first m transitions and bootstraps only the remaining mass γ^m, which reduces the amplification to $1/(1-\gamma^m)$. We prove that every probability-valued Bellman target buys contraction at the same rate per model call, so depth decides only where the horizon is paid for; that the Bellman structure holds at every noise level, so denoising score matching fits it; and that the estimator obeys an OPE bound separating generative from dynamics error. On five control benchmarks, deeper blocks bring the generator closer to the frozen model, and on Walker2d one-step bootstrapping remains far from it even after fifty times more sweeps. A released one-step occupancy model, TD^2-flow, has higher log-RMSE than BRD on all five tasks and lower rank correlation on four. BRD is more accurate on both metrics than model-based rollout on all four horizon-matched tasks, and than STITCH-OPE on three of them.

1 INTRODUCTION

In robotics (Kober et al., 2013), healthcare (Gottesman et al., 2019), and recommender systems (Li et al., 2011), a new policy often has to be evaluated before it can be deployed. Off-policy evaluation (OPE) estimates the value of an evaluation policy from data logged by a different behavior policy (Jiang & Li, 2016; Thomas & Brunskill, 2016). Its central difficulty is distribution shift: the evaluation policy reaches states and actions that the logged data cover poorly, and errors propagate over the whole horizon (Dudík et al., 2011; Kallus & Uehara, 2019).

Importance sampling (IS) reweights logged returns by products of policy ratios (Precup et al., 2000; Thomas et al., 2015). It requires the behavior policy or an estimate of it, and its variance typically grows rapidly with the horizon. Model-based OPE instead fits a one-step dynamics model and simulates the evaluation policy in it (Kidambi et al., 2020; Yu et al., 2020). Its accuracy is limited by what the data identify about the dynamics, and model errors propagate through the rollout (Talvitie, 2014). It also stores no long-horizon object: every value query rebuilds the long-horizon distribution with a fresh sequential chain of model calls whose length scales with the effective horizon $1/(1-\gamma)$.

A line of work stores that distribution in a generative model instead. γ-models, distributional successor measures, Bellman diffusion models, and temporal difference flows learn distributions over future states by bootstrapping: the training target places mass $1-\gamma$ on an observed next state and mass γ on the model's own prediction from that state (Janner et al., 2020; Wiltzer et al., 2024; Schramm & Boularias, 2024; Farebrother et al., 2025). Bootstrapping on a learned generator has a price. Every fitting step leaves an approximation error, the next target inherits it, and the standard fitted-iteration argument amplifies the per-step error by up to $1/(1-\gamma)$, a factor of 100 at $\gamma = 0.99$. When targets are formed from logged transitions, finite-sample error and fitting error also enter the same fitted object, so neither can be assessed apart from the final estimate. Diffusion-based OPE that stitches

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

sub-trajectories from a behavior-trained prior under target-policy guidance (Goli et al., 2025) adds a behavior model and a guidance approximation to the error budget.

This paper asks how to store the evaluation policy's long-horizon distribution in a diffusion model so that the generator's fitting error is not amplified by the full horizon, and so that the error coming from the data stays separate from the error coming from the fit. Our answer, **Bellman Recursion Diffusion (BRD)**, uses the one-step model that model-based OPE already fits, but only as a teacher. BRD fits a transition model on the logged data, freezes it, and trains a conditional diffusion model on the frozen model's discounted distribution of future transitions under the evaluation policy (Figure 1). Each training target is a block Bellman mixture: the frozen model simulates the first m transitions, and the previous generator supplies the remaining mass γ^m. The prefix is exact for the frozen model, so only the tail inherits fitting error, and the amplification drops from $1/(1-\gamma)$ to $1/(1-\gamma^m)$, about 2 at $\gamma = 0.99$ and $m = 64$. Because the target is defined by a frozen model, the generator's error is measured against that model alone, and the model's own error reaches the estimate as a separate term.

The construction fits diffusion training because every branch of the block mixture can be sampled, and because Gaussian corruption is linear in the clean law: the noised target is again a Bellman mixture at every noise level, so denoising score matching fits it without evaluating any branch weight. The block form is also not arbitrary: among probability-valued combinations of Bellman powers, contraction and frozen-model work obey an exact linear relation, so no target buys contraction more cheaply, and at a given depth the block target is the unique contraction-optimal one. Depth therefore decides only where the horizon is paid for during training: in the exact frozen model or in the generator's own predictions, which amplify its fitting error. At $m = 1$ the whole horizon is bootstrapped, as in generative temporal-difference methods; increasing m moves more of that horizon into the frozen model and reduces the tail mass geometrically. At evaluation the horizon is not paid at all, since each sample costs a fixed number of network evaluations. BRD does not remove model bias, because its target is the frozen model's occupancy at every depth. Our contributions are:

- **Method.** BRD trains a conditional diffusion model of the evaluation policy's discounted transition distribution on finite-depth block Bellman targets of a frozen one-step model, with no behavior model, density ratio, or guidance.
- **Theory.** An exact contraction–work relation for probability-valued Bellman targets, under which every target buys contraction at the same rate per model call and the block target is optimal at each depth (Theorem 3.2); a Bellman equation for the target's score at every noise level, with posterior branch weights (Theorem 3.3); and an OPE bound that separates generative, dynamics, reward, and Monte Carlo error, in which depth acts only on the generative term, certified by a residual under a uniquely optimal audit distribution (Proposition 3.4 and Theorem 3.5).
- **Experiments.** On five control benchmarks with the learned model held fixed, block depth consistently improves fidelity to the frozen model; on Walker2d, the one-step target remains far from it after fifty times more sweeps. The released TD^2-flow has higher log-RMSE than BRD on all five tasks and lower rank correlation on four. BRD is better on both metrics than model-based rollout on all four horizon-matched tasks, and than STITCH-OPE on three of them.

2 Preliminaries

Markov decision processes. We consider a discounted Markov decision process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P^\star, r, \gamma, \mu_0)$ with state and action spaces $\mathcal{S}$ and $\mathcal{A}$, true transition kernel $P^\star : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, reward $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ with $\|r\|_\infty \le R_{\max}$, discount $\gamma \in (0, 1)$, and initial-state distribution $\mu_0 \in \Delta(\mathcal{S})$. A stationary policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ maps states to action distributions. Under policy π and dynamics P, the discounted return is:

$$J(\pi, P) = \mathbb{E}_{\pi,P}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t, s_{t+1})\right], \qquad s_0 \sim \mu_0, \tag{1}$$

and we write $J(\pi) := J(\pi, P^\star)$ for the value under the true dynamics.

Off-policy evaluation. In the offline setting no further interaction with the environment is possible. We are given a static dataset $\mathcal{D}$ of transitions (s, a, r, s') collected by a behavior policy π_b under $P^\star$, and the goal is to estimate $J(\pi_e)$ for an evaluation policy π_e. In BRD the behavior policy enters only

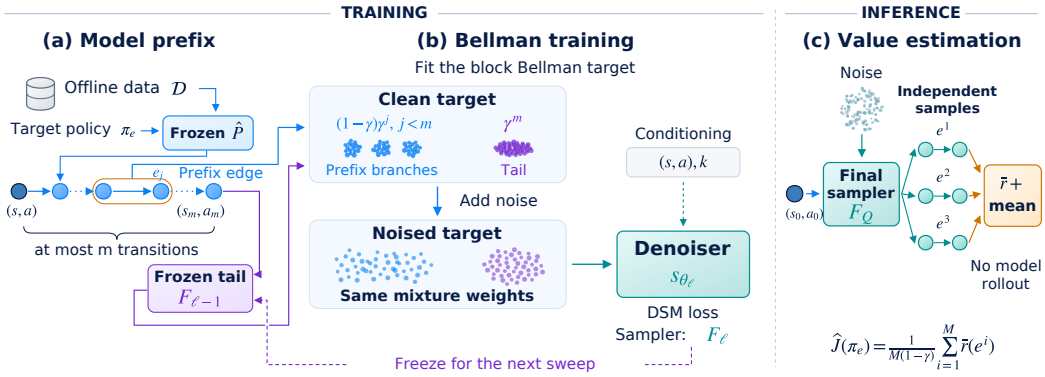

Figure 1: **Overview of BRD. (a)** We fit the dynamics model $\widehat{P}$ on $\mathcal{D}$, freeze it, and roll it out under π_e for at most m transitions (edges) from a training root (s, a). **(b)** At sweep ℓ, each clean target x_0 is drawn from the block Bellman mixture $\mathcal{B}^{\pi_e}_{\widehat{P},m} F_{\ell-1}$: the jth prefix transition with probability $(1-\gamma)\gamma^j$, or a sample of the frozen generator $F_{\ell-1}(\cdot|s_m, a_m)$ with probability γ^m. Noising keeps these weights on densities; on scores they become posterior weights (Theorem 3.3). The denoiser is trained on the noised target with the root (s, a) as conditioning, and its sampler F_ℓ becomes the next teacher. **(c)** At evaluation, each sample e^i is one discounted future transition drawn from $F_Q(\cdot|s_0, a_0)$ by reverse diffusion, not a step of a trajectory; we average their rewards and scale by $1/(1-\gamma)$.

through which transitions $\mathcal{D}$ contains: BRD never estimates π_b, forms a density ratio π_e/π_b, or fits a model of the behavior distribution. Coverage remains a constraint of the data rather than of the estimator: without it no estimator can identify $J(\pi_e)$ uniformly (Theorem C.4).

Diffusion models. Diffusion models (Ho et al., 2020; Song et al., 2020) generate data by reversing a noising process. For a clean sample $x_0 \in \mathbb{R}^d$ with conditioning variable c, the forward process uses Gaussian kernels $q_k(x_k \mid x_0) = \mathcal{N}(x_k; \alpha_k x_0, \sigma_k^2 I)$, $k \in \{1, \dots, K\}$, with schedules chosen so that x_K is approximately $\mathcal{N}(0, I)$. Generation simulates the reverse process using the conditional score $\nabla_{x_k} \log q_k(x_k \mid c)$, which is estimated by denoising score matching:

$$\min_\theta \mathbb{E}_{k,c,x_0,x_k} \left[\omega(k) \left\| s_\theta(x_k, k, c) - \nabla_{x_k} \log q_k(x_k \mid x_0) \right\|_2^2 \right], \tag{2}$$

where $x_k \sim q_k(\cdot \mid x_0)$ and $\omega(k) > 0$ is a weighting. Two properties of equation 2 matter below. It needs only samples of the clean law, never its density, and its minimizer is the score of whatever law generated those samples. Everything the model fits is therefore a property of the *noised* laws $q_k(\cdot \mid c)$. In BRD, the conditioning $c = (s, a)$ is a state-action pair and the clean sample $x_0 = (\widetilde{s}, \widetilde{a}, \widetilde{s}')$ is a transition drawn from the discounted future of (s, a).

3 Bellman Recursion Diffusion

We define the target (Section 3.1), turn its Bellman equation into sampleable block targets (Section 3.2), show that the structure survives noising (Section 3.3), and bound the OPE error (Section 3.4).

3.1 The Target: A Conditional Bellman Fixed Point

Discounted transition occupancy. For dynamics P, evaluation policy π_e, and measurable $A \subseteq \mathcal{S} \times \mathcal{A} \times \mathcal{S}$, define the *conditional transition occupancy*:

$$F_P^{\pi_e}(A|s, a) := (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \, \mathbb{P}_{\pi_e, P}\big((s_t, a_t, s_{t+1}) \in A \,\big|\, s_0{=}s, a_0{=}a\big). \tag{3}$$

We model transitions because the reward depends on (s, a, s'), and condition on (s, a) because bootstrapping queries the same object at continuation roots.

Bellman equation and value readout. Let $\mathcal{C}_P^{\pi_e}$ advance the conditioning root by one step, $(\mathcal{C}_P^{\pi_e} F)(A|s,a) := \mathbb{E}_{s'\sim P(\cdot|s,a),\, a'\sim\pi_e(\cdot|s')}[F(A|s',a')]$. Separating the current transition from the rest of equation 3 gives the Bellman equation:

$$F_P^{\pi_e}(A|s,a) = (1-\gamma)\,\mathbb{P}_{s'\sim P(\cdot|s,a)}\big((s,a,s')\in A\big) + \gamma\,(\mathcal{C}_P^{\pi_e} F_P^{\pi_e})(A|s,a). \tag{4}$$

We write $\mathcal{B}_{P,1}^{\pi_e} F$ for the right-hand side of equation 4 as a function of F. Since $\mathcal{C}_P^{\pi_e}$ averages F over the next root, it does not increase conditional total variation, so $\mathcal{B}_{P,1}^{\pi_e}$ is a γ-contraction with unique fixed point equation 3. The fixed point determines the value. With $d_0^{\pi_e}$ the law of (s_0,a_0) under $s_0\sim\mu_0$ and $a_0\sim\pi_e(\cdot|s_0)$, we have:

$$J(\pi_e,P) = \frac{1}{1-\gamma}\,\mathbb{E}_{(s_0,a_0)\sim d_0^{\pi_e},\,(\widetilde{s},\widetilde{a},\widetilde{s}')\sim F_P^{\pi_e}(\cdot|s_0,a_0)}\big[r(\widetilde{s},\widetilde{a},\widetilde{s}')\big]. \tag{5}$$

Frozen one-step models. In offline OPE, $P^\star$ is unknown. We fit $\widehat{P}(\cdot|s,a)$ and $\widehat{r}(s,a,s')$ on the logged transitions and freeze both before training the generator, so the target is $F_{\widehat{P}}^{\pi_e}$ rather than $F_{P^\star}^{\pi_e}$. Freezing makes errors attributable: a generator that could also adjust the dynamics would mix what the data failed to identify with what it failed to fit. The behavior policy appears nowhere in BRD, and π_e enters only through $\mathcal{C}_P^{\pi_e}$, so no density ratio or behavior prior is estimated.

3.2 Block Bellman Targets

Sampling $F_{\widehat{P}}^{\pi_e}$ directly reintroduces a rollout at every query. We instead fit it with conditional diffusion models $F_1,\dots,F_Q$, each trained on targets built from the frozen model and its predecessor.

Why one-step bootstrapping is not enough. The one-step target $\mathcal{B}_{\widehat{P},1}^{\pi_e} F_{\ell-1}$ can be sampled directly: with probability $1-\gamma$ return a transition of $\widehat{P}$ from the root, otherwise advance one step and query the frozen $F_{\ell-1}$. This is the target of generative temporal-difference methods (Janner et al., 2020; Farebrother et al., 2025). Its contraction factor is γ, so the fitting error of each sweep is amplified by up to $1/(1-\gamma)$ in the final generator (equation 15 with $m=1$). We let the frozen model resolve the first m steps and bootstrap the rest, which gives the *block Bellman target*:

$$(\mathcal{B}_{P,m}^{\pi_e} F)(A|s,a) := \mathbb{E}_{\pi_e,P}\left[(1-\gamma)\sum_{j=0}^{m-1}\gamma^j \mathbf{1}\{(s_j,a_j,s_{j+1})\in A\} + \gamma^m F(A|s_m,a_m)\right], \tag{6}$$

where the trajectory starts at $(s_0,a_0)=(s,a)$. The prefix materializes m transitions of the frozen model, and only the remaining mass γ^m is assigned to the generator. With $\mathrm{TV}(\mu,\nu) := \sup_A|\mu(A)-\nu(A)|$, we write $D_\infty(F,H) := \sup_{s,a}\mathrm{TV}(F(\cdot|s,a),H(\cdot|s,a))$ and $D_\beta(F,H) := \mathbb{E}_{(s,a)\sim\beta}\mathrm{TV}(F(\cdot|s,a),H(\cdot|s,a))$ for a distribution β on $\mathcal{S}\times\mathcal{A}$.

Proposition 3.1 (Block Bellman update)**.** *For every $m\ge 1$, the block operator is the m-fold composition of the one-step operator, fixes the conditional occupancy, and contracts at rate γ^m:*

$$\mathcal{B}_{P,m}^{\pi_e} = \big(\mathcal{B}_{P,1}^{\pi_e}\big)^{\circ m}, \qquad \mathcal{B}_{P,m}^{\pi_e} F_P^{\pi_e} = F_P^{\pi_e}, \qquad D_\infty\big(\mathcal{B}_{P,m}^{\pi_e}F, \mathcal{B}_{P,m}^{\pi_e}H\big) \le \gamma^m D_\infty(F,H). \tag{7}$$

The block target is the right mixture. Proposition 3.1 does not say whether some other combination of Bellman powers would trade model work for contraction more efficiently. None does.

Theorem 3.2 (Contraction–work relation)**.** *For a finitely supported weight vector $v=(v_0,v_1,\dots)$, let $\mathcal{B}_{P,v}^{\pi_e} := \sum_n v_n(\mathcal{B}_{P,1}^{\pi_e})^n$. This operator maps conditional probability laws to conditional probability laws on every finite MDP if and only if v is a probability vector. For such v, let $\kappa(v)$ be its smallest uniform contraction modulus in D_∞, and $\mathrm{cost}(v)$ the smallest expected number of frozen-model calls of an exact sampler of its targets when the model can only be queried along a trajectory. Then:*

$$1-\kappa(v) = (1-\gamma)\,\mathrm{cost}(v), \tag{8}$$

and if v is supported on $\{0,\dots,m\}$, then $\kappa(v)\ge\gamma^m$, with equality only for $\mathcal{B}_{P,m}^{\pi_e}$.

Every probability-valued target buys contraction at the same rate, $1-\gamma$ per expected model call. Deeper blocks are thus not cheaper per unit of contraction; they move the horizon out of the approximate generator, where each bootstrap step adds fitting error, and into the frozen model, where it is exact. Signed schemes such as Chebyshev acceleration leave the probability simplex and therefore yield no sampleable denoising target (Corollary C.13).

Algorithm 1 Bellman Recursion Diffusion (BRD)

Require: Logged data $\mathcal{D}$, evaluation policy π_e, block depth m, sweeps Q, pool size N, updates per sweep U, root weight $\lambda \in [0, 1)$, diffusion schedule $\{q_k\}_{k=1}^K$

Ensure: Final generator F_Q, value estimate $\widehat{J}(\pi_e)$

Fit $\widehat{P}$ and $\widehat{r}$ on $\mathcal{D}$ and freeze them
Build root samplers for $d_0^{\pi_e}$ and $\widehat{\beta}_m^\star$ by rolling out $\widehat{P}$ under π_e
Fit F_0, with parameters θ_0, by equation 2 to one-step transitions sampled from $\widehat{P}$
for $\ell = 1, \dots, Q$ **do**
    Draw N roots from $\lambda d_0^{\pi_e} + (1-\lambda)\widehat{\beta}_m^\star$ and one target per root via equation 9 with frozen $F_{\ell-1}$
    Initialize $\theta_\ell \leftarrow \theta_{\ell-1}$
    for $u = 1, \dots, U$ **do**
        Draw a minibatch of root–target pairs $((s_0, a_0), x_0)$ from the pool
        Sample $k \sim \mathrm{Unif}\{1, \dots, K\}$ and $x_k \sim q_k(\cdot|x_0)$
        Update θ_ℓ using equation 2 with $c = (s_0, a_0)$
    end for
    Let F_ℓ be the reverse-diffusion sampler induced by s_{θ_ℓ}
end for
Draw M evaluation samples from F_Q and compute $\widehat{J}(\pi_e)$ via equation 11

Recursive diffusion fitting. We initialize F_0 by fitting equation 2 to one-step transitions (s, a, s'), $s' \sim \widehat{P}(\cdot|s, a)$. At sweep $\ell \in \{1, \dots, Q\}$ we freeze $F_{\ell-1}$ and draw a pool of N clean targets from $\mathcal{B}^{\pi_e}_{\widehat{P},m} F_{\ell-1}$. For each target, starting from a root $(s_0, a_0) = (s, a)$, we draw a branch $\iota \in \{0, \dots, m\}$ and set:

$$\mathbb{P}(\iota{=}j) = \begin{cases} (1{-}\gamma)\gamma^j, & j < m, \\ \gamma^m, & j = m, \end{cases} \qquad \begin{cases} x_0 = (s_\iota, a_\iota, s_{\iota+1}), & \iota < m, \\ x_0 \sim F_{\ell-1}(\cdot|s_m, a_m), & \iota = m, \end{cases} \tag{9}$$

where the prefix is generated under $\widehat{P}$ and π_e and the frozen generator is queried only on the tail branch. Each x_0 is ordinary clean data: we draw minibatches from the pool, draw k and $x_k \sim q_k(\cdot|x_0)$, and minimize equation 2 with $c = (s, a)$. The evaluation policy enters through the target alone, with no guidance term and no prior trained on behavior data. Reverse diffusion defines F_ℓ, the teacher of sweep $\ell{+}1$. We warm-start θ_ℓ from $\theta_{\ell-1}$, where θ_0 parametrizes F_0, so training never holds more than two generators in memory at once.

Training and audit roots. Section 3.4 averages the Bellman residual under the root distribution:

$$\beta_m^\star(B) := (1{-}\gamma^m) \sum_{q=0}^{\infty} \gamma^{mq}\, \mathbb{P}_{\pi_e, \widehat{P}}\big((s_{mq}, a_{mq}) \in B\big), \qquad (s_0, a_0) \sim d_0^{\pi_e}, \tag{10}$$

which weights the root reached after q blocks by γ^{mq}. By Theorem C.20, $\beta_m^\star$ is the unique root distribution that converts the averaged residual into a bound on generator error with the smallest factor. It is an *audit* distribution: the certificate holds for any generator, however trained. Training roots must instead cover the roots where the next sweep queries the frozen generator, the m-step successors of the training roots. We draw them from $\lambda d_0^{\pi_e} + (1-\lambda)\beta_m^\star$: any $\lambda < 1$ keeps these queries within the training distribution up to a bounded density ratio, and initial roots alone ($\lambda = 1$) do not (Proposition C.22). We use $\lambda = 1/2$ throughout. Both distributions are sampled by rolling out the frozen model (Lemma C.21); Algorithm 1 summarizes the procedure.

Value estimation. The estimator is the Monte Carlo version of equation 5, with no rollout through $\widehat{P}$: we draw $(s_0^{(i)}, a_0^{(i)}) \sim d_0^{\pi_e}$ and $(\widetilde{s}_i, \widetilde{a}_i, \widetilde{s}'_i) \sim F_Q(\cdot|s_0^{(i)}, a_0^{(i)})$ by reverse diffusion, and compute:

$$\widehat{J}(\pi_e) = \frac{1}{(1-\gamma)M} \sum_{i=1}^{M} \bar{r}(\widetilde{s}_i, \widetilde{a}_i, \widetilde{s}'_i), \qquad \bar{r} := \mathrm{clip}(\widehat{r}, -R_{\max}, R_{\max}). \tag{11}$$

Clipping does not increase the reward error (Lemma C.30) and bounds the Monte Carlo summands.

3.3 Bellman Structure in Denoising Space

A diffusion model fits only the noised laws of its target (Section 2), so the Bellman structure of equation 6 is useful to it only if it survives noising. It does, exactly.

The mixture survives noising. Write $p_k^F(x_k|s,a) := \int q_k(x_k|x_0)F(dx_0|s,a)$ for the noised law of a conditional law F, $p_{j,k}^{\mathrm{pre}}$ for the noised law of the jth prefix transition, and $p_{m,k}^{\mathrm{tgt},F}$ for the noised law of $\mathcal{B}_{\widehat{P},m}^{\pi_e}F$. Forward corruption integrates against the clean law and is therefore linear in it:

$$p_{m,k}^{\mathrm{tgt},F}(x_k|s,a) = (1-\gamma)\sum_{j=0}^{m-1}\gamma^j p_{j,k}^{\mathrm{pre}}(x_k|s,a) + \gamma^m\big((\mathcal{C}_{\widehat{P}}^{\pi_e})^m p_k^F(x_k|\cdot)\big)(s,a). \tag{12}$$

Differentiating the logarithm turns this mixture of densities into a mixture of scores, weighted by the posterior over the latent branch rather than by the prior coefficients.

Theorem 3.3 (Bellman score equation)**.** *Under the Gaussian corruption of Section 2, for every k and every conditional law F, let $\varpi_{j,k}^F(x_k,s,a)$ be the posterior probability that the branch in equation 9 was j, and $\varpi_{\mathrm{tail},k}^F(ds',da'|x_k,s,a)$ the posterior law of the continuation root on the tail branch, both given x_k and (s,a). These weights are nonnegative and sum to one, and:*

$$s_{m,k}^{\mathrm{tgt},F}(x_k,s,a) = \sum_{j=0}^{m-1}\varpi_{j,k}^F(x_k,s,a)\, s_{j,k}^{\mathrm{pre}}(x_k,s,a) + \int s_k^F(x_k,s',a')\,\varpi_{\mathrm{tail},k}^F(ds',da'|x_k,s,a), \tag{13}$$

where each s is the gradient in x_k of the log of the corresponding noised density. At $F = F_{\widehat{P}}^{\pi_e}$ both sides are scores of the same law, so the score of the target occupancy satisfies a Bellman equation at every noise level. The same weights relate the posterior means (denoisers).

Two consequences matter for implementation. The weights depend on x_k, so combining branch scores with the clean coefficients $(1-\gamma)\gamma^j$, as a naive score-level bootstrap would, is biased except in degenerate cases (Corollary C.16; Figure 2d). And the weights require normalized densities, which a score network does not provide; sampling the mixture in clean space as in equation 9 and applying denoising score matching recovers the posterior-weighted score without evaluating any weight. Noising also regularizes the $j{=}0$ branch, which fixes the first two coordinates of x_0 at the conditioning pair and so has no density (Remark C.18).

3.4 Theoretical Guarantees

We bound the OPE error of Algorithm 1. The analysis separates what the generator fails to fit from what the data fail to identify, and shows that block depth acts only on the former.

Assumption 1 (Diffusion fitting error)**.** *At every sweep $\ell = 1,\dots,Q$, $D_\infty\big(F_\ell, \mathcal{B}_{\widehat{P},m}^{\pi_e}F_{\ell-1}\big) \le \varepsilon_{\mathrm{proj}}$.*

Assumption 2 (One-step model error)**.** $\sup_{s,a}\mathrm{TV}\big(\widehat{P}(\cdot|s,a), P^\star(\cdot|s,a)\big) \le \varepsilon_{\mathrm{dyn}}$ *and* $\sup_{s,a,s'}|\widehat{r}(s,a,s') - r(s,a,s')| \le \varepsilon_r$.

$\varepsilon_{\mathrm{proj}}$ is the analogue of the projection error in fitted value evaluation; $\varepsilon_{\mathrm{dyn}}$ and ε_r measure what the data identified. Assumption 1 is uniform over (s,a) while training covers only the root distribution, so we use it only for the a priori bound equation 15; for a trained generator, the guarantee is the residual certificate below, which needs no uniform assumption.

Block depth and dynamics error. A deeper block places less weight on the bootstrapped generator, so one might expect it to also reduce the influence of a misspecified model. It does not.

Proposition 3.4 (Block depth does not change the dynamics error)**.** *For every dynamics P and every $m \ge 1$, $\mathcal{B}_{P,m}^{\pi_e}$ has the unique fixed point $F_P^{\pi_e}$, so the target and its dependence on the dynamics are the same for all block depths. Under Assumption 2:*

$$D_\infty\big(F_{\widehat{P}}^{\pi_e}, F_{P^\star}^{\pi_e}\big) \le \frac{\varepsilon_{\mathrm{dyn}}}{1-\gamma}. \tag{14}$$

Block depth thus changes how fitting error accumulates and nothing else; no choice of m substitutes for a better one-step model, and the dynamics term of Theorem 3.5 is the same for every m.

The audited residual. For a conditional generator F, let $\mathrm{Res}_F(s,a) := \mathrm{TV}\big(F(\cdot|s,a), (\mathcal{B}^{\pi_e}_{\widehat{P},m}F)(\cdot|s,a)\big)$ be its block Bellman residual, which involves only the frozen model and the generator. Averaging under $\beta^\star_m$ gives the *audited residual* $\varepsilon_{\mathrm{res}} := \mathbb{E}_{(s,a)\sim\beta^\star_m}[\mathrm{Res}_{F_Q}(s,a)]$, and the optimal conversion factor of Theorem C.20 turns it into the *generative certificate* $\varepsilon_{\mathrm{gen}} := \varepsilon_{\mathrm{res}}/(1-\gamma^m)$, which bounds the generator's error at the initial roots.

Theorem 3.5 (OPE guarantee)**.** *Under Assumption 1, after Q sweeps:*

$$D_\infty\big(F_Q, F^{\pi_e}_{\widehat{P}}\big) \le \gamma^{mQ} D_\infty\big(F_0, F^{\pi_e}_{\widehat{P}}\big) + \frac{1-\gamma^{mQ}}{1-\gamma^m}\,\varepsilon_{\mathrm{proj}}. \tag{15}$$

With no assumption, the final generator satisfies the residual certificate:

$$D_{d_0^{\pi_e}}\big(F_Q, F^{\pi_e}_{\widehat{P}}\big) \le \varepsilon_{\mathrm{gen}}. \tag{16}$$

Under Assumption 2, with probability at least $1-\delta$ over the M evaluation samples:

$$\big|\widehat{J}(\pi_e) - J(\pi_e)\big| \le \underbrace{\frac{2R_{\max}\varepsilon_{\mathrm{gen}}}{1-\gamma}}_{\text{(a) generative}} + \underbrace{\frac{2R_{\max}\varepsilon_{\mathrm{dyn}}}{(1-\gamma)^2}}_{\text{(b) dynamics}} + \underbrace{\frac{\varepsilon_r}{1-\gamma}}_{\text{(c) reward}} + \underbrace{\frac{2R_{\max}}{1-\gamma}\sqrt{\frac{\log(2/\delta)}{2M}}}_{\text{(d) Monte Carlo}}. \tag{17}$$

In equation 15 and in $\varepsilon_{\mathrm{gen}}$ alike, fitting error is amplified by $1/(1-\gamma^m)$ rather than $1/(1-\gamma)$, and no other audit distribution gives the certificate a smaller factor. Term (a) involves $\widehat{P}$ and the generator but not $P^\star$; it is controlled by the excess denoising loss of F_Q on its own block targets $\mathcal{B}^{\pi_e}_{\widehat{P},m}F_Q$, the objective of one further sweep (Appendix C.9). Because it does not involve $P^\star$, the generator can be audited against the frozen model alone. Term (b) is inherited from $\widehat{P}$, has the $O(\varepsilon_{\mathrm{dyn}}/(1-\gamma)^2)$ dependence standard in model-based evaluation, and does not depend on m. Term (c) is fixed once $\widehat{r}$ is fitted, and term (d) decays at the rate $M^{-1/2}$. No term involves the behavior policy. Total variation is conservative here: because the generator samples the $j{=}0$ branch by diffusion, its residual is at least $1-\gamma$ at every root (Remark C.18). Section 4 therefore measures the generative error by transport and value distances to the frozen model, and Appendix C.12 gives Wasserstein analogues.

4 Experiments

We follow the order of the theory: we first hold the learned model fixed and vary only the Bellman target, then test whether block depth closes the gap between generator and frozen model, and finally compare BRD with existing OPE estimators.

Setup. We use D4RL (Fu et al., 2020) Hopper, Walker2d, and HalfCheetah (`medium-v2`) with the ten DOPE target policies (Fu et al., 2021), and Gym (Brockman et al., 2016) Acrobot and Pendulum with the target policies of the STITCH-OPE release. Reference returns come from simulator rollouts. Unless noted, BRD uses $m = 64$, $\gamma = 0.99$, $\lambda = 1/2$, and eight sweeps (Appendix D.2). The comparators are model-based rollout (MBR), which fits a one-step dynamics and reward model and rolls it out under π_e, STITCH-OPE (Goli et al., 2025), FQE (Le et al., 2019), IS (Precup et al., 2000), and WIS (Voloshin et al., 2019), all run from released code (Appendix D.6). We report the log-RMSE of min–max normalized values and Spearman's ρ (Appendix D.5). A task is horizon-matched when its reference is scored at the comparators' configured horizon. This holds for all tasks except Pendulum, whose log-RMSE against the comparators is therefore descriptive (Appendix D.4).

Changing only the Bellman target. Table 1 holds the frozen model fixed and varies only the Bellman target. At the same eight-sweep budget, $m = 16$ improves both metrics over $m = 1$ on all five tasks. Training the one-step target longer does not close the gap: on Walker2d (training seed 0), its sliced Wasserstein distance to the frozen model stays between 0.61 and 0.69 from 32 to 400 sweeps, whereas $m = 16$ reaches 0.14 after eight (Appendix D.10). Its OPE error does improve with training, from -0.18 to -1.68 in log-RMSE at 400 sweeps, close to the -1.74 of BRD on the same seed, but its fidelity to the frozen model does not. The released model-free TD^2-flow (Farebrother et al., 2025; Zheng et al., 2026), trained for 10^6 steps, has higher log-RMSE than BRD at $m = 16$ on all five tasks and lower rank correlation on four (Appendix D.7).

Table 1: **Changing only the Bellman target.** Within each row, the two m columns share the frozen model and reward readout and differ only in block depth (arms of Appendix D.10). TD^2-flow is the released model-free one-step implementation. Means over training seeds; TD^2-flow uses training seed 0. †Every column targets the same infinite-horizon value, so this row is comparable within the table. Standard deviations: Table 12.

	Log-RMSE↓			**Spearman ρ ↑**		
Environment	m=1	TD^2-flow	m=16	m=1	TD^2-flow	m=16
Hopper	−0.06	−0.22	**−1.30**	+0.03	+0.32	**+0.53**
Walker2d	−0.20	−0.59	**−1.86**	−0.14	+0.04	**+0.72**
HalfCheetah	−0.37	−0.57	**−0.85**	+0.58	+0.27	**+0.84**
Acrobot	−0.71	+1.61	**−0.93**	−0.05	**+0.71**	+0.69
Pendulum†	−0.38	−0.21	**−1.57**	+0.60	+0.21	**+1.00**

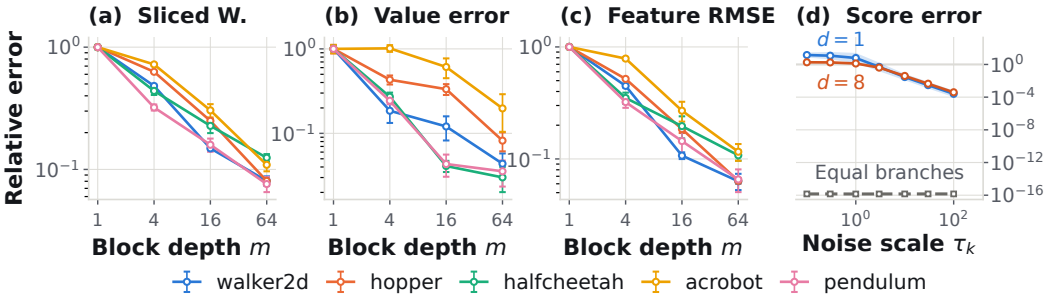

Figure 2: **Block depth and score weights. (a)–(c)** Sliced Wasserstein distance, value error, and feature RMSE between generator and frozen model, normalized by the $m = 1$ mean; mean ± one sd over five training seeds. **(d)** Relative score error when clean coefficients replace the posterior weights of Theorem 3.3, against noise scale, in dimension $d \in \{1, 8\}$; coincident branches give 10^{-16}.

Table 2: **Off-policy evaluation.** MBR is direct rollout of a learned one-step model. Entries are means where repeated runs exist. Bold marks the best mean, not significance (Tables 13 and 14). †Not horizon-matched, so its log-RMSE is descriptive (Appendix D.4).

	Environment	IS	WIS	FQE	MBR	STITCH-OPE	**BRD**
Log-RMSE↓	Hopper	0.594	−1.191	−1.018	0.099	**−2.278**	−2.139
	Walker2d	−0.187	−1.033	−0.091	−0.829	−1.255	**−1.710**
	HalfCheetah	−0.248	−0.744	−0.240	−0.369	−0.441	**−0.854**
	Acrobot	−0.168	−0.976	0.093	−1.416	−1.509	**−2.085**
	Pendulum†	−0.180	−0.232	−2.045	−0.482	−2.087	**−2.281**
Spearman ρ ↑	Hopper	0.321	−0.497	0.321	−0.176	**0.812**	0.751
	Walker2d	−0.733	0.455	0.273	0.503	0.504	**0.697**
	HalfCheetah	−0.042	0.879	−0.491	0.527	0.376	**0.886**
	Acrobot	−0.543	0.543	0.943	0.943	0.751	**0.962**
	Pendulum†	−0.667	−0.154	0.975	0.872	0.975	**1.000**

Depth closes the gap to the frozen model. Across $m \in \{1, 4, 16, 64\}$ and five training seeds, all three fidelity measures in Figure 2(a)–(c) decrease monotonically with depth in 14 of 15 task–metric pairs, and 88 of 90 pairwise depth orderings exceed seed noise (Table 15). From $m = 1$ to $m = 64$ the value error to the frozen model falls by a factor between 5 (Acrobot) and 33 (HalfCheetah). This is the finite-budget behavior that equation 15 predicts: deeper blocks spend more frozen-model calls per target, but they contract faster and amplify fitting error less. OPE accuracy need not follow fidelity: on Walker2d it peaks at $m = 16$, which beats $m = 64$ by 0.148 in log-RMSE at a seed standard deviation of about 0.05 (Table 16). Depth controls how faithfully the generator reproduces the frozen model, bias included, so a closer fit need not give a closer value.

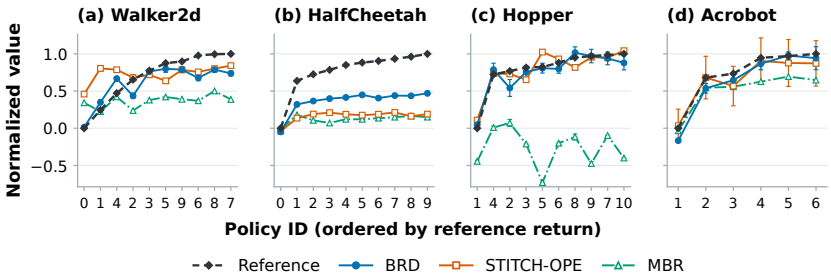

Figure 3: **Policy-level values.** Per-environment normalized values ordered by reference return; black: reference, blue: BRD, orange: STITCH-OPE. Error bars: one sd; Pendulum omitted (horizon mismatch).

Full OPE. On the four horizon-matched tasks, BRD has lower log-RMSE and higher ρ than MBR. MBR fits its own one-step model, so this is a pipeline-level comparison; the effect of the Bellman target on a fixed model is isolated in Table 1. Against STITCH-OPE, BRD wins both metrics on Walker2d, HalfCheetah, and Acrobot and loses on Hopper, where the mean over the first six of fifteen training seeds (-2.271) would read as a tie; only HalfCheetah also survives policy resampling (Table 13 and Appendix D.8). At the policy level (Figure 3), STITCH-OPE overestimates the lowest-return Walker2d policies, which BRD tracks, and both methods underestimate the HalfCheetah policies above the worst one, BRD by less. WIS nearly ties BRD in ρ on HalfCheetah (0.879 vs. 0.886), but it is deterministic and admits no permutation test. The released doubly robust estimator is also less accurate than BRD on both metrics on Walker2d, HalfCheetah, and Acrobot (Table 11); on Hopper it needs about 20 CPU-hours per target policy and is not reported. On Walker2d and HalfCheetah the reference itself is resolved to about 2 units of return, the scale of some gaps in Table 2 (Appendix D.4). Pendulum is excluded from these claims: at its configured 200-step horizon, BRD trails STITCH-OPE by 0.855 in log-RMSE (Appendix D.4).

Mechanism checks. In Figure 2(d), replacing the posterior weights of Theorem 3.3 by the clean coefficients gives relative score error of order one at low noise; the error shrinks as the noised branches merge and reaches machine precision when they coincide, so the mismatch is structural rather than numerical. Training on initial roots alone ($\lambda = 1$) instead of $\lambda = 1/2$ increases the sliced Wasserstein distance to the same frozen model on all ten Walker2d policies, by 142.8% on average, as Proposition C.22 predicts. Its effect on OPE is mixed: log-RMSE worsens from -1.741 to -1.125 while ρ rises from 0.685 to 0.758 (Table 18). The two arms match the training schedule, not total computation, since initial roots avoid the rollouts that build continuation roots, and the ablation does not show that $\lambda = 1/2$ is optimal.

Inference cost. The denoiser has 2.6 times the forward-pass FLOPs of the dynamics model. A BRD sample takes 32 denoiser calls, about 83 dynamics-model calls in FLOPs and about 65 in forward-pass time at batch size $2{,}048$, independent of the horizon; a geometrically stopped rollout takes 100 dynamics calls per discounted transition in expectation at $\gamma = 0.99$. Training the generator costs more than one rollout evaluation, so it pays off only over repeated queries, and we claim no end-to-end speedup (Appendix D.12).

5 Conclusion

Block depth moves the horizon from the generator into the frozen model without changing the fixed point, and reproduces that occupancy far more faithfully than one-step bootstrapping at the same budget. BRD is more accurate than model-based rollout on every horizon-matched task, but it inherits the bias of its own frozen model.

AI USE STATEMENT

In this work, we used generative AI tools for language editing and code assistance. We have not used generative AI tools to generate experimental results or fabricate data, and other tasks requiring disclosure are not applicable to this work. Additionally, we used generative AI tools to improve the clarity and presentation of the manuscript. We have reviewed all AI-assisted work. AI-assisted code was inspected and tested by the authors for correctness, and all AI-assisted text was reviewed and revised by the authors. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? *arXiv preprint arXiv:2211.15657*, 2022.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

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

Peter Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural Computation*, 5(4):613–624, 1993.

Miroslav Dudík, John Langford, and Lihong Li. Doubly robust policy evaluation and learning. *arXiv preprint arXiv:1103.4601*, 2011.

Jesse Farebrother, Matteo Pirotta, Andrea Tirinzoni, Remi Munos, Alessandro Lazaric, and Ahmed Touati. Temporal difference flows. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 16102–16143, 2025.

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

Justin Fu, Mohammad Norouzi, Ofir Nachum, George Tucker, Ziyu Wang, Alexander Novikov, Mengjiao Yang, Michael R Zhang, Yutian Chen, Aviral Kumar, et al. Benchmarks for deep off-policy evaluation. *arXiv preprint arXiv:2103.16596*, 2021.

Hossein Goli, Michael Gimelfarb, Nathan Samuel de Lara, Haruki Nishimura, Masha Itkina, and Florian Shkurti. Stitch-ope: Trajectory stitching with guided diffusion for off-policy evaluation. *arXiv preprint arXiv:2505.20781*, 2025.

Omer Gottesman, Fredrik Johansson, Matthieu Komorowski, Aldo Faisal, David Sontag, Finale Doshi-Velez, and Leo Anthony Celi. Guidelines for reinforcement learning in healthcare. *Nature medicine*, 25(1):16–18, 2019.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

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

Matthew Thomas Jackson, Michael Tryfan Matthews, Cong Lu, Benjamin Ellis, Shimon Whiteson, and Jakob Foerster. Policy-guided diffusion. *arXiv preprint arXiv:2404.06356*, 2024.

Michael Janner, Igor Mordatch, and Sergey Levine. γ-models: Generative temporal difference learning for infinite-horizon prediction. In *Advances in Neural Information Processing Systems*, 2020.

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

Nan Jiang and Lihong Li. Doubly robust off-policy value evaluation for reinforcement learning. In *International conference on machine learning*, pp. 652–661. PMLR, 2016.

Nathan Kallus and Masatoshi Uehara. Intrinsically efficient, stable, and bounded off-policy evaluation for reinforcement learning. *Advances in neural information processing systems*, 32, 2019.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. *Advances in neural information processing systems*, 33: 21810–21823, 2020.

Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

Ilya Kostrikov, Kumar Krishna Agrawal, Debidatta Dwibedi, Sergey Levine, and Jonathan Tompson. Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. In *International Conference on Learning Representations*, 2019.

Hoang Le, Cameron Voloshin, and Yisong Yue. Batch policy learning under constraints. In *International Conference on Machine Learning*, pp. 3703–3712. PMLR, 2019.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pp. 297–306, 2011.

Fan-Ming Luo, Tian Xu, Hang Lai, Xiong-Hui Chen, Weinan Zhang, and Yang Yu. A survey on model-based reinforcement learning. *Science China Information Sciences*, 67(2):121101, 2024.

Thomas M. Moerland, Joost Broekens, Aske Plaat, and Catholijn M. Jonker. Model-based reinforcement learning: A survey. *Foundations and Trends in Machine Learning*, 16(1):1–118, 2023.

Ofir Nachum, Yinlam Chow, Bo Dai, and Lihong Li. Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. *Advances in neural information processing systems*, 32, 2019.

Tim Pearce, Tabish Rashid, Anssi Kanervisto, Dave Bignell, Mingfei Sun, Raluca Georgescu, Sergio Valcarcel Macua, Shan Zheng Tan, Ida Momennejad, Katja Hofmann, et al. Imitating human behaviour with diffusion models. *arXiv preprint arXiv:2301.10677*, 2023.

Doina Precup, Richard S Sutton, and Satinder Singh. Eligibility traces for off-policy policy evaluation. 2000.

Marc Rigter, Bruno Lacerda, and Nick Hawes. Rambo-rl: Robust adversarial model-based offline reinforcement learning. *Advances in neural information processing systems*, 35:16082–16097, 2022.

Liam Schramm and Abdeslam Boularias. Bellman diffusion models. *arXiv preprint arXiv:2407.12163*, 2024.

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.

Erik Talvitie. Model regularization for stable sample rollouts. In *UAI*, pp. 780–789, 2014.

Philip Thomas and Emma Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. In *International conference on machine learning*, pp. 2139–2148. PMLR, 2016.

Philip Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. High-confidence off-policy evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

Masatoshi Uehara, Jiawei Huang, and Nan Jiang. Minimax weight and q-function learning for off-policy evaluation. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Cameron Voloshin, Hoang M Le, Nan Jiang, and Yisong Yue. Empirical study of off-policy policy evaluation for reinforcement learning. *arXiv preprint arXiv:1911.06854*, 2019.

Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. *arXiv preprint arXiv:2208.06193*, 2022.

Harley Wiltzer, Jesse Farebrother, Arthur Gretton, Yunhao Tang, Andre Barreto, Will Dabney, Marc G. Bellemare, and Mark Rowland. A distributional analogue to the successor representation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 52994–53016, 2024.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. *Advances in neural information processing systems*, 33:14129–14142, 2020.

Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. Combo: Conservative offline model-based policy optimization. *Advances in neural information processing systems*, 34:28954–28967, 2021.

Chongyi Zheng, Seohong Park, Sergey Levine, and Benjamin Eysenbach. Intention-conditioned flow occupancy models. In *International Conference on Learning Representations*, 2026.

Supplementary Materials

A Limitations and Scope

Model bias. BRD is model-based. Its target is the occupancy of the frozen model, so it cannot recover dynamics under the evaluation policy that the logged data do not identify (Theorem C.4), and no block depth reduces the resulting error (Proposition 3.4).

Cost. The generator is trained per evaluation policy, and its training cost exceeds that of a single rollout-based evaluation; training the generator pays off only when the same occupancy is queried repeatedly, and we do not measure end-to-end cost (Appendix D.12). Theorem C.32 shows that some horizon-dependent cost is unavoidable: BRD pays it offline, through the rollouts that build its targets and root samplers.

Certificates. The guarantees are stated in total variation, which is conservative for the near-deterministic kernels used on Hopper and HalfCheetah. Because our generator samples every coordinate by diffusion, including the singular $j{=}0$ branch, its total-variation residual is at least $1-\gamma$ at every root (Remark C.18), a floor that transport distances do not share; the Wasserstein variant of Appendix C.12 requires $\gamma L_T < 1$, which contact-rich dynamics need not satisfy. The generative certificate is controlled by the excess denoising loss, which training reduces but which cannot be evaluated without the entropy of the target (Remark C.26); we do not report numerical values of the certificate.

Experimental scope. The root weight λ and the block depth were not tuned, the training-root ablation uses one training seed and one environment, and Pendulum is scored at a truncation of 256 steps while its configured horizon is 200 (Appendix D.4).

Theory. The contraction–work relation of Theorem 3.2 holds over probability-valued targets and does not rule out acceleration through signed measures, value functions, analytic multi-step operators, or problem-specific structure.

B Related Work

Off-policy evaluation. Importance sampling corrects the mismatch between behavior and evaluation policies with products of policy ratios (Precup et al., 2000; Thomas et al., 2015); its variance grows with the horizon, and it requires the behavior policy or an estimate of it. Fitted Q evaluation estimates the evaluation policy's value function by Bellman regression on logged transitions (Le et al., 2019; Levine et al., 2020), and doubly robust estimators combine the two (Jiang & Li, 2016). Distribution-correction methods estimate the ratio between the evaluation policy's occupancy and the data distribution (Nachum et al., 2019; Uehara et al., 2020); they avoid the behavior policy but require the data to cover the evaluation policy's occupancy. Model-based OPE fits one-step dynamics and simulates the evaluation policy (Yu et al., 2020; Rigter et al., 2022; Kidambi et al., 2020), a paradigm studied extensively in offline reinforcement learning (Yu et al., 2021; M. Moerland et al., 2023; Luo et al., 2024). BRD belongs to this last family and shares its dependence on the one-step model; it differs in what is stored after training, namely a generator of the model's occupancy rather than the model alone, and in keeping the error of that generator separate from the model's in the OPE bound.

Generative models of future states. The successor representation expresses discounted future state occupancy (Dayan, 1993). γ-models learn a generative model of the discounted distribution of future states by temporal-difference bootstrapping on logged transitions (Janner et al., 2020). Distributional successor measures learn a distribution over such occupancies (Wiltzer et al., 2024), Bellman diffusion models represent the successor state measure with a diffusion model (Schramm & Boularias, 2024), and temporal difference flows train geometric horizon models by flow matching with bootstrapped probability paths (Farebrother et al., 2025). These methods bootstrap one step at a time from logged transitions and need no dynamics model. They were proposed as models for long-horizon prediction, planning, and policy improvement rather than as off-policy estimators, so we

position BRD relative to them here and compare it empirically with OPE methods. BRD differs in three respects (Table 3). It bootstraps through a frozen one-step model, which lets the model resolve m steps of each target and reduces the amplification of fitting error from $1/(1-\gamma)$ to $1/(1-\gamma^m)$ (Theorem 3.5); Theorem 3.2 shows that this block form is optimal among probability-valued targets. It generates transitions rather than states, so rewards of the form $r(s,a,s')$ can be read out directly. And it targets OPE, with a bound in which generative and dynamics error are separate terms.

Table 3: **Methods that learn long-horizon distributions with generative models.** "Bootstrap" describes how the training target reaches beyond the observed data.

Method	Generated object	Bootstrap	Behavior model
γ-models (Janner et al., 2020)	future states	one step, logged data	no
Distributional SM (Wiltzer et al., 2024)	distributions of occupancies	one step, logged data	no
Bellman diffusion (Schramm & Boularias, 2024)	future states	one step, logged data	no
TD flows (Farebrother et al., 2025)	future states	one step, logged data	no
STITCH-OPE (Goli et al., 2025)	sub-trajectories	none (stitching)	yes
BRD (ours)	future transitions	m steps, frozen model	no

Diffusion models for decision making. Diffusion models (Ho et al., 2020; Song et al., 2020) have been used to generate trajectories for planning (Janner et al., 2022; Ajay et al., 2022), to represent policies (Wang et al., 2022; Chi et al., 2025; Pearce et al., 2023), and, with guidance (Ho & Salimans, 2022), to steer generated trajectories toward a target policy (Jackson et al., 2024). STITCH-OPE applies guided diffusion to OPE: it trains a trajectory prior on behavior data, steers generated sub-trajectories toward the evaluation policy, and stitches them into long-horizon rollouts (Goli et al., 2025). BRD uses no guidance and no behavior prior; the evaluation policy enters only through the Bellman target, and the diffusion model is fitted to that target by ordinary denoising score matching.

C Detailed Theoretical Guarantees

Organization. Appendix C.1 fixes notation and Appendix C.2 proves three measure-theoretic lemmas used throughout. Appendix C.3 shows that coverage is necessary for any estimator. Appendix C.4 proves Proposition 3.1 and the a priori bound equation 15, Appendix C.5 proves Proposition 3.4, and Appendix C.6 proves Theorem 3.2. Appendix C.7 proves Theorem 3.3 and shows that its hypotheses hold automatically under Gaussian corruption. Appendix C.8 proves the residual certificate equation 16, the optimality of $\beta_m^\star$, and the coverage property of the training roots. Appendix C.9 relates the certificate to the denoising loss, and Appendix C.10 proves equation 17. Appendices C.11 and C.12 contain supplementary results.

C.1 Notation

All symbols of Section 3 are retained: $P^\star, \widehat{P}, \pi_b, \pi_e, \mu_0$; $F_P^{\pi_e}$ Equation (3); $\mathcal{C}_P^{\pi_e}$ (Section 3.1); $\mathcal{B}_{P,m}^{\pi_e}$ Equation (6); TV, D_∞, D_β (Section 3.2); Res_F; $\beta_m^\star$ Equation (10); $\varepsilon_{\mathrm{res}}, \varepsilon_{\mathrm{gen}}, \varepsilon_{\mathrm{proj}}, \varepsilon_{\mathrm{dyn}}, \varepsilon_r$ (Section 3.4). Additionally:

$$\mathcal{X} := \mathcal{S} \times \mathcal{A}, \qquad \mathcal{E} := \mathcal{S} \times \mathcal{A} \times \mathcal{S}, \qquad x = (s,a) \in \mathcal{X}, \qquad e = (s,a,s') \in \mathcal{E}, \tag{18}$$

$d_0^{\pi_e} := \mathrm{Law}(s_0, a_0)$ with $s_0 \sim \mu_0$, $a_0 \sim \pi_e(\cdot \mid s_0)$; $\mathbf{1}_A$ is the indicator of A and δ_z the Dirac mass at z. An unsubscripted x always denotes a root; subscripted x_0 and x_k denote clean and noised diffusion samples, as in Section 2, and roots along a fixed chain are written $y_0, y_1, \ldots$.

Root kernel. $T_P^{\pi_e}(\,\mathrm{d}x' \mid x) := P(\,\mathrm{d}s' \mid s,a)\,\pi_e(\,\mathrm{d}a' \mid s')$ is a Markov kernel on $\mathcal{X}$, and for every conditional law F, measurable $A \subseteq \mathcal{E}$ and $n \geq 0$:

$$\begin{aligned}(\mathcal{C}_P^{\pi_e} F)(A \mid x) &= \int_{\mathcal{X}} F(A \mid x')\, T_P^{\pi_e}(\,\mathrm{d}x' \mid x),\\ \big((\mathcal{C}_P^{\pi_e})^n F\big)(A \mid x) &= \int_{\mathcal{X}} F(A \mid x')\, (T_P^{\pi_e})^n(\,\mathrm{d}x' \mid x).\end{aligned} \tag{19}$$

Transition laws and affine form. For $j \geq 0$:

$$G^{\pi_e}_{P,j}(A \mid x) := \Pr_{\pi_e,P}\big((s_j, a_j, s_{j+1}) \in A \mid (s_0, a_0) = x\big), \qquad A \subseteq \mathcal{E}, \tag{20}$$

and Equation (6) is equivalently:

$$\mathcal{B}^{\pi_e}_{P,m} F = \underbrace{(1-\gamma) \sum_{j=0}^{m-1} \gamma^j G^{\pi_e}_{P,j}}_{\text{constant part}} + \underbrace{\gamma^m (\mathcal{C}^{\pi_e}_P)^m F}_{\text{linear part}}. \tag{21}$$

Total variation. For a finite signed measure σ with $\sigma(\mathcal{X}) = 0$, $\mathrm{TV}(\sigma) := \sup_A |\sigma(A)|$; thus $\mathrm{TV}(\mu, \nu) = \mathrm{TV}(\mu - \nu)$ and $D_\infty(F, H) = \sup_x \mathrm{TV}\big(F(\cdot \mid x), H(\cdot \mid x)\big)$.

C.2 Three elementary facts

Lemma C.1 (Markov kernels do not increase total variation)**.** *Let* K *be a Markov kernel and* $(\mu\mathsf{K})(A) := \int \mathsf{K}(A \mid x)\, \mu(\mathrm{d}x)$. *For probability measures* μ, ν*:*

$$\mathrm{TV}(\mu\mathsf{K}, \nu\mathsf{K}) \leq \mathrm{TV}(\mu, \nu), \tag{22}$$

and the same holds with $\mu - \nu$ *replaced by any finite signed measure of zero total mass.*

Proof. Let $\sigma := \mu - \nu$, so $\sigma(\mathcal{X}) = 0$. For measurable A:

$$(\mu\mathsf{K})(A) - (\nu\mathsf{K})(A) = \int \mathsf{K}(A \mid x)\, \sigma(\mathrm{d}x). \tag{23}$$

Let $\mathcal{X} = B^+ \sqcup B^-$ be a Hahn decomposition of σ. For measurable $0 \leq f \leq 1$, $-\sigma^-(\mathcal{X}) \leq \int f \, \mathrm{d}\sigma \leq \sigma^+(\mathcal{X})$, and $\sigma^+(\mathcal{X}) = \sigma(B^+) = -\sigma(B^-) = \sigma^-(\mathcal{X})$ since $\sigma(\mathcal{X}) = 0$. Hence:

$$\sup_{0 \leq f \leq 1} \Big| \int f \, \mathrm{d}\sigma \Big| = \sigma(B^+) = \sup_B |\sigma(B)| = \mathrm{TV}(\mu, \nu), \tag{24}$$

with equality at $f = \mathbf{1}_{B^+}$. Applying Equation (24) with $f = \mathsf{K}(A \mid \cdot) \in [0, 1]$ in Equation (23):

$$\big|(\mu\mathsf{K})(A) - (\nu\mathsf{K})(A)\big| \leq \mathrm{TV}(\mu, \nu). \tag{25}$$

Take $\sup_A$. □

Lemma C.2 (Mixtures against a common measure)**.** *Let* ξ *be a probability measure and* $x \mapsto F_x, H_x$ *measurable families of probability measures. Then:*

$$\mathrm{TV}\Big(\int F_x \, \xi(\mathrm{d}x), \int H_x \, \xi(\mathrm{d}x)\Big) \leq \int \mathrm{TV}(F_x, H_x)\, \xi(\mathrm{d}x). \tag{26}$$

Proof. For measurable A:

$$\begin{aligned} \Big| \int F_x(A)\, \xi(\mathrm{d}x) - \int H_x(A)\, \xi(\mathrm{d}x) \Big| &= \Big| \int \big(F_x(A) - H_x(A)\big)\, \xi(\mathrm{d}x) \Big| \\ &\leq \int \big|F_x(A) - H_x(A)\big|\, \xi(\mathrm{d}x) \leq \int \mathrm{TV}(F_x, H_x)\, \xi(\mathrm{d}x). \end{aligned} \tag{27}$$

The right side is independent of A; take $\sup_A$. □

Lemma C.3 (Bounded readout)**.** *Let* $-\infty < a < b < \infty$*,* f *measurable with* $f \in [a, b]$*, and* μ, ν *probability measures. Then:*

$$\Big| \int f \, \mathrm{d}\mu - \int f \, \mathrm{d}\nu \Big| \leq (b - a)\, \mathrm{TV}(\mu, \nu). \tag{28}$$

Proof. Put $g := (f - a)/(b - a) \in [0, 1]$. Since $(\mu - \nu)(\mathcal{X}) = 0$:

$$\int f \, \mathrm{d}(\mu - \nu) = a\,(\mu - \nu)(\mathcal{X}) + (b - a) \int g \, \mathrm{d}(\mu - \nu) = (b - a) \int g \, \mathrm{d}(\mu - \nu), \tag{29}$$

and $\big|\int g \, \mathrm{d}(\mu - \nu)\big| \leq \mathrm{TV}(\mu, \nu)$ by Equation (24). □

C.3 Coverage and nonidentification

Theorem C.4 (Coverage is necessary for uniform identification)**.** *There exist two MDPs with common* $(\mathcal{S}, \mathcal{A}, r, \mu_0, \pi_b, \pi_e)$ *such that the law of* $\mathcal{D}$ *is identical under both and* $|J(\pi_e, P_1) - J(\pi_e, P_0)| = \gamma/(1-\gamma)$*. Consequently, for every estimator* $\widehat{J} = \widehat{J}(\mathcal{D})$*:*

$$\max_{i \in \{0,1\}} \mathbb{E}^{\text{data}} \big| \widehat{J} - J(\pi_e, P_i) \big| \geq \frac{\gamma}{2(1-\gamma)}. \tag{30}$$

Proof. Step 1 (models). Both MDPs use the spaces and initial distribution:

$$\mathcal{S} = \{s_0, s_{\text{safe}}, s_{\text{good}}, s_{\text{bad}}\}, \qquad \mathcal{A} = \{a_b, a_\star\}, \qquad \mu_0 = \delta_{s_0}, \tag{31}$$

$\pi_b(a_b \mid s_0) = 1$, $\pi_e(a_\star \mid s_0) = 1$, $\pi_b = \pi_e$ on $\mathcal{S} \setminus \{s_0\}$, and:

$$P_0(s_{\text{safe}} \mid s_0, a_b) = P_1(s_{\text{safe}} \mid s_0, a_b) = 1, \qquad P_0(s_{\text{bad}} \mid s_0, a_\star) = 1, \qquad P_1(s_{\text{good}} \mid s_0, a_\star) = 1, \tag{32}$$

with $s_{\text{safe}}, s_{\text{good}}, s_{\text{bad}}$ absorbing under both, and $r(s, a, s') = \mathbf{1}\{s = s_{\text{good}}\}$.

Step 2 (identical data). $\pi_b(a_\star \mid s_0) = 0$, so the behavior chain visits only (s_0, a_b) and $\{s_{\text{safe}}\} \times \mathcal{A}$, on which $P_0 = P_1$. Hence $\text{Law}_{P_0}(\mathcal{D}) = \text{Law}_{P_1}(\mathcal{D})$ and so $\text{Law}_{P_0}(\widehat{J}) = \text{Law}_{P_1}(\widehat{J})$.

Step 3 (values). Under P_0, $s_t = s_{\text{bad}}$ for $t \geq 1$, so $J(\pi_e, P_0) = 0$. Under P_1, $s_t = s_{\text{good}}$ for $t \geq 1$, so:

$$J(\pi_e, P_1) = \sum_{t=1}^{\infty} \gamma^t = \frac{\gamma}{1-\gamma}. \tag{33}$$

Step 4 (bound). For $z \in \mathbb{R}$, $|z - J(\pi_e, P_0)| + |z - J(\pi_e, P_1)| \geq \gamma/(1-\gamma)$. Setting $z = \widehat{J}$ and taking $\mathbb{E}^{\text{data}}$, well defined by Step 2:

$$\mathbb{E}^{\text{data}} \big| \widehat{J} - J(\pi_e, P_0) \big| + \mathbb{E}^{\text{data}} \big| \widehat{J} - J(\pi_e, P_1) \big| \geq \frac{\gamma}{1-\gamma}, \tag{34}$$

so the larger summand is at least $\gamma/(2(1-\gamma))$. □

Remark C.5. This is the standard two-point argument. It shows that the term ε_{dyn} in Equation (17) is bounded below for every estimator on this class and is not an artifact of BRD.

C.4 Block Bellman algebra and projected iteration

Lemma C.6 (Bellman power expansion)**.** *For every* $n \geq 1$*, conditional law* F *and measurable* $A \subseteq \mathcal{E}$*:*

$$\big((\mathcal{B}_{P,1}^{\pi_e})^n F\big)(A \mid x) = (1-\gamma) \sum_{j=0}^{n-1} \gamma^j G_{P,j}^{\pi_e}(A \mid x) + \gamma^n \big((\mathcal{C}_P^{\pi_e})^n F\big)(A \mid x). \tag{35}$$

Proof. Induction on n. For $n = 1$, Equation (4) with continuation argument F reads $(\mathcal{B}_{P,1}^{\pi_e} F)(A \mid x) = (1-\gamma) G_{P,0}^{\pi_e}(A \mid x) + \gamma (\mathcal{C}_P^{\pi_e} F)(A \mid x)$. Assume Equation (35) at n. By the case $n = 1$:

$$\big((\mathcal{B}_{P,1}^{\pi_e})^{n+1} F\big)(A \mid x) = (1-\gamma) G_{P,0}^{\pi_e}(A \mid x) + \gamma \big(\mathcal{C}_P^{\pi_e} (\mathcal{B}_{P,1}^{\pi_e})^n F\big)(A \mid x). \tag{36}$$

By the induction hypothesis and linearity of $\mathcal{C}_P^{\pi_e}$:

$$\big(\mathcal{C}_P^{\pi_e} (\mathcal{B}_{P,1}^{\pi_e})^n F\big)(A \mid x) = (1-\gamma) \sum_{j=0}^{n-1} \gamma^j \big(\mathcal{C}_P^{\pi_e} G_{P,j}^{\pi_e}\big)(A \mid x) + \gamma^n \big((\mathcal{C}_P^{\pi_e})^{n+1} F\big)(A \mid x). \tag{37}$$

By the Markov property:

$$\big(\mathcal{C}_P^{\pi_e} G_{P,j}^{\pi_e}\big)(A \mid x) = \int G_{P,j}^{\pi_e}(A \mid x') \, T_P^{\pi_e}(\, \mathrm{d}x' \mid x) = G_{P,j+1}^{\pi_e}(A \mid x). \tag{38}$$

Substituting Equation (38) into Equation (37) and then into Equation (36):

$$\big((\mathcal{B}_{P,1}^{\pi_e})^{n+1} F\big)(A \mid x) = (1-\gamma) \sum_{i=0}^{n} \gamma^i G_{P,i}^{\pi_e}(A \mid x) + \gamma^{n+1} \big((\mathcal{C}_P^{\pi_e})^{n+1} F\big)(A \mid x), \tag{39}$$

where $i = j + 1$. This is Equation (35) at $n + 1$. □

Proof of Proposition 3.1. *(i)* $\mathcal{B}^{\pi_e}_{P,m} = (\mathcal{B}^{\pi_e}_{P,1})^{\circ m}$. Equation (35) at $n = m$ equals Equation (21) term by term.

(ii) $\mathcal{B}^{\pi_e}_{P,m} F^{\pi_e}_P = F^{\pi_e}_P$. Splitting Equation (3) at $t = m$:

$$F^{\pi_e}_P(A \mid x) = (1-\gamma)\sum_{t=0}^{m-1} \gamma^t G^{\pi_e}_{P,t}(A \mid x) + (1-\gamma)\sum_{t=m}^{\infty} \gamma^t G^{\pi_e}_{P,t}(A \mid x). \tag{40}$$

With $t = m + i$ and $G^{\pi_e}_{P,m+i}(A \mid x) = \int G^{\pi_e}_{P,i}(A \mid x')(T^{\pi_e}_P)^m(\,\mathrm{d}x' \mid x)$:

$$\begin{aligned}(1-\gamma)\sum_{t=m}^{\infty} \gamma^t G^{\pi_e}_{P,t}(A \mid x) &= \gamma^m \int \Big[(1-\gamma)\sum_{i=0}^{\infty} \gamma^i G^{\pi_e}_{P,i}(A \mid x')\Big](T^{\pi_e}_P)^m(\,\mathrm{d}x' \mid x) \\ &= \gamma^m \int F^{\pi_e}_P(A \mid x')(T^{\pi_e}_P)^m(\,\mathrm{d}x' \mid x) = \gamma^m \big((\mathcal{C}^{\pi_e}_P)^m F^{\pi_e}_P\big)(A \mid x).\end{aligned} \tag{41}$$

Insert Equation (41) into Equation (40) and compare with Equation (21).

(iii) Contraction. By Equation (21) the constant parts cancel:

$$\mathrm{TV}\big((\mathcal{B}^{\pi_e}_{P,m}F)(\cdot \mid x), (\mathcal{B}^{\pi_e}_{P,m}H)(\cdot \mid x)\big) = \gamma^m\ \mathrm{TV}\big(((\mathcal{C}^{\pi_e}_P)^m F)(\cdot \mid x), ((\mathcal{C}^{\pi_e}_P)^m H)(\cdot \mid x)\big). \tag{42}$$

By Equation (19) both arguments are mixtures over $(T^{\pi_e}_P)^m(\cdot \mid x)$; by Lemma C.2:

$$\begin{aligned}\mathrm{TV}\big(((\mathcal{C}^{\pi_e}_P)^m F)(\cdot \mid x), ((\mathcal{C}^{\pi_e}_P)^m H)(\cdot \mid x)\big) &\le \int \mathrm{TV}\big(F(\cdot \mid x'), H(\cdot \mid x')\big)(T^{\pi_e}_P)^m(\,\mathrm{d}x' \mid x) \\ &\le D_\infty(F, H).\end{aligned} \tag{43}$$

Combine and take $\sup_x$. □

Corollary C.7 (Uniqueness of the fixed point)**.** *$F^{\pi_e}_P$ is the unique conditional probability law fixed by $\mathcal{B}^{\pi_e}_{P,m}$.*

Proof. If $F = \mathcal{B}^{\pi_e}_{P,m}F$ and $H = \mathcal{B}^{\pi_e}_{P,m}H$, then $D_\infty(F, H) \le \gamma^m D_\infty(F, H)$, and $\gamma^m < 1$ gives $D_\infty(F, H) = 0$. □

Proposition C.8 (Projected iteration)**.** *Let $F_0, \dots, F_Q$ satisfy $D_\infty(F_\ell, \mathcal{B}^{\pi_e}_{P,m}F_{\ell-1}) \le \varepsilon_{\mathrm{proj},\ell}$, $\ell = 1, \dots, Q$. Then:*

$$D_\infty(F_Q, F^{\pi_e}_P) \le \gamma^{mQ} D_\infty(F_0, F^{\pi_e}_P) + \sum_{\ell=1}^{Q} \gamma^{m(Q-\ell)} \varepsilon_{\mathrm{proj},\ell}. \tag{44}$$

Proof. Let $\epsilon_\ell := D_\infty(F_\ell, F^{\pi_e}_P)$. Using $F^{\pi_e}_P = \mathcal{B}^{\pi_e}_{P,m}F^{\pi_e}_P$ and Proposition 3.1:

$$\epsilon_\ell \le D_\infty(F_\ell, \mathcal{B}^{\pi_e}_{P,m}F_{\ell-1}) + D_\infty(\mathcal{B}^{\pi_e}_{P,m}F_{\ell-1}, \mathcal{B}^{\pi_e}_{P,m}F^{\pi_e}_P) \le \varepsilon_{\mathrm{proj},\ell} + \gamma^m \epsilon_{\ell-1}. \tag{45}$$

Unrolling Equation (45) for $\ell = Q, Q-1, \dots, 1$ gives Equation (44). □

With $\varepsilon_{\mathrm{proj},\ell} \equiv \varepsilon_{\mathrm{proj}}$ and $\sum_{\ell=1}^{Q} \gamma^{m(Q-\ell)} = (1-\gamma^{mQ})/(1-\gamma^m)$, Equation (44) is Equation (15), the first claim of Theorem 3.5.

C.5 Dynamics sensitivity is unaffected by block depth

Proposition 3.4 consists of (i) the statement that every block operator has the same fixed point, and (ii) the bound $D_\infty(F^{\pi_e}_{\widehat{P}}, F^{\pi_e}_{P^\star}) \le \varepsilon_{\mathrm{dyn}}/(1-\gamma)$. Remark C.9 shows how the dependence on m cancels in the first-order sensitivity of the fixed point.

Proof of the first claim of Proposition 3.4. Fix P and $m \geq 1$. By Proposition 3.1(ii), $\mathcal{B}^{\pi_e}_{P,m} F^{\pi_e}_P = F^{\pi_e}_P$, and by Corollary C.7 no other conditional probability law is fixed. Hence:

$$\mathrm{Fix}\big(\mathcal{B}^{\pi_e}_{P,m}\big) = \{F^{\pi_e}_P\} \qquad \forall\, m \geq 1, \tag{46}$$

and the right side does not depend on m. Applied to $P = \widehat{P}$ and $P = P^\star$, the target and the dynamics-induced difference $F^{\pi_e}_{\widehat{P}} - F^{\pi_e}_{P^\star}$ are the same for every block depth. □

Remark C.9 (Where the block depth cancels). Let $P_t = P^\star + t\Delta P + o(t)$ be a smooth family of kernels and let $\dot{F}_m$ be the derivative at $t = 0$ of the fixed point of $\mathcal{B}^{\pi_e}_{P_t,m}$; by equation 46 it does not depend on m, and the computation below shows how the m-step equation reduces to the one-step one. Write $\mathcal{C} := \mathcal{C}^{\pi_e}_{P^\star}$ and $\Delta_m := \partial_t|_{t=0}\, \mathcal{B}^{\pi_e}_{P_t,m} F^{\pi_e}_{P^\star}$, the derivative with the argument $F^{\pi_e}_{P^\star}$ held fixed; it collects the variation of both the constant and the linear part of Equation (21). The map $(t, F) \mapsto F - \mathcal{B}^{\pi_e}_{P_t,m} F$ has F-derivative $\mathrm{I} - \gamma^m \mathcal{C}^m$ at the fixed point, which is invertible since $\|\gamma^m \mathcal{C}^m\| \leq \gamma^m < 1$ by Lemma C.1; the implicit function theorem gives differentiability and:

$$(\mathrm{I} - \gamma^m \mathcal{C}^m)\dot{F}_m = \Delta_m, \qquad \Delta_m = \sum_{j=0}^{m-1} \gamma^j \mathcal{C}^j \Delta_1, \tag{47}$$

where the second identity follows from $\mathcal{B}^{\pi_e}_{P_t,m} = (\mathcal{B}^{\pi_e}_{P_t,1})^{\circ m}$, the chain rule, and $\mathcal{B}^{\pi_e}_{P^\star,1} F^{\pi_e}_{P^\star} = F^{\pi_e}_{P^\star}$. Since:

$$\mathrm{I} - \gamma^m \mathcal{C}^m = (\mathrm{I} - \gamma\mathcal{C}) \sum_{j=0}^{m-1} \gamma^j \mathcal{C}^j, \tag{48}$$

the m-step system reduces to $(\mathrm{I} - \gamma\mathcal{C})\dot{F}_1 = \Delta_1$.

Lemma C.10 (One-step dynamics error controls the one-step Bellman gap)**.** *Under Assumption 2, for every conditional probability law F:*

$$D_\infty\big(\mathcal{B}^{\pi_e}_{\widehat{P},1} F, \mathcal{B}^{\pi_e}_{P^\star,1} F\big) \leq \varepsilon_{\mathrm{dyn}}. \tag{49}$$

Proof. Fix $x = (s, a)$ and define the Markov kernel from $\mathcal{S}$ to $\mathcal{E}$:

$$\mathsf{K}(\,\mathrm{d}e \mid s') := (1-\gamma)\,\delta_{(s,a,s')}(\,\mathrm{d}e) + \gamma \int \pi_e(\,\mathrm{d}a' \mid s')\, F(\,\mathrm{d}e \mid s', a'), \tag{50}$$

which does not depend on the dynamics. By Equation (4), for $P \in \{\widehat{P}, P^\star\}$:

$$(\mathcal{B}^{\pi_e}_{P,1} F)(\cdot \mid x) = P(\cdot \mid s, a)\,\mathsf{K}. \tag{51}$$

By Lemma C.1 and Assumption 2:

$$\mathrm{TV}\big((\mathcal{B}^{\pi_e}_{\widehat{P},1} F)(\cdot \mid x), (\mathcal{B}^{\pi_e}_{P^\star,1} F)(\cdot \mid x)\big) \leq \mathrm{TV}\big(\widehat{P}(\cdot \mid s, a), P^\star(\cdot \mid s, a)\big) \leq \varepsilon_{\mathrm{dyn}}. \tag{52}$$

Take $\sup_x$. □

Lemma C.11 (Block Bellman gap)**.** *Under Assumption 2, for every $m \geq 1$ and every conditional probability law F:*

$$D_\infty\big(\mathcal{B}^{\pi_e}_{\widehat{P},m} F, \mathcal{B}^{\pi_e}_{P^\star,m} F\big) \leq \varepsilon_{\mathrm{dyn}} \frac{1-\gamma^m}{1-\gamma}. \tag{53}$$

Proof. Let $d_n := \sup_F D_\infty\big((\mathcal{B}^{\pi_e}_{\widehat{P},1})^n F, (\mathcal{B}^{\pi_e}_{P^\star,1})^n F\big)$; $d_1 \leq \varepsilon_{\mathrm{dyn}}$ by Lemma C.10. Inserting $\mathcal{B}^{\pi_e}_{\widehat{P},1} (\mathcal{B}^{\pi_e}_{P^\star,1})^n F$:

$$\begin{aligned} D_\infty\big((\mathcal{B}^{\pi_e}_{\widehat{P},1})^{n+1} F, (\mathcal{B}^{\pi_e}_{P^\star,1})^{n+1} F\big) &\leq D_\infty\big(\mathcal{B}^{\pi_e}_{\widehat{P},1} (\mathcal{B}^{\pi_e}_{\widehat{P},1})^n F, \mathcal{B}^{\pi_e}_{\widehat{P},1} (\mathcal{B}^{\pi_e}_{P^\star,1})^n F\big) \\ &\quad + D_\infty\big(\mathcal{B}^{\pi_e}_{\widehat{P},1} (\mathcal{B}^{\pi_e}_{P^\star,1})^n F, \mathcal{B}^{\pi_e}_{P^\star,1} (\mathcal{B}^{\pi_e}_{P^\star,1})^n F\big) \leq \gamma d_n + \varepsilon_{\mathrm{dyn}}, \end{aligned} \tag{54}$$

by Proposition 3.1 (first term) and Lemma C.10 (second term). Hence $d_m \leq \varepsilon_{\mathrm{dyn}} \sum_{j=0}^{m-1} \gamma^j$, and $\mathcal{B}^{\pi_e}_{P,m} = (\mathcal{B}^{\pi_e}_{P,1})^{\circ m}$ gives Equation (53). □

Proof of the second claim of Proposition 3.4. Let $\Delta := D_\infty(F^{\pi_e}_{\widehat{P}}, F^{\pi_e}_{P^\star}) \le 1$. By the fixed-point property:

$$\begin{aligned}\Delta &= D_\infty\big(\mathcal{B}^{\pi_e}_{\widehat{P},m}F^{\pi_e}_{\widehat{P}}, \mathcal{B}^{\pi_e}_{P^\star,m}F^{\pi_e}_{P^\star}\big)\\ &\le D_\infty\big(\mathcal{B}^{\pi_e}_{\widehat{P},m}F^{\pi_e}_{\widehat{P}}, \mathcal{B}^{\pi_e}_{\widehat{P},m}F^{\pi_e}_{P^\star}\big) + D_\infty\big(\mathcal{B}^{\pi_e}_{\widehat{P},m}F^{\pi_e}_{P^\star}, \mathcal{B}^{\pi_e}_{P^\star,m}F^{\pi_e}_{P^\star}\big) \le \gamma^m\Delta + \varepsilon_{\mathrm{dyn}}\frac{1-\gamma^m}{1-\gamma}, \end{aligned} \tag{55}$$

by Proposition 3.1 and Lemma C.11. Thus $(1-\gamma^m)\Delta \le \varepsilon_{\mathrm{dyn}}(1-\gamma^m)/(1-\gamma)$, i.e. $\Delta \le \varepsilon_{\mathrm{dyn}}/(1-\gamma)$ for every $m \ge 1$. □

C.6 The contraction–work relation

This subsection proves Theorem 3.2. We write $m_{\max}$ for the largest index in the support of v.

For $m_{\max} \in \mathbb{N}$ and $v = (v_0, \dots, v_{m_{\max}}) \in \mathbb{R}^{m_{\max}+1}$, define:

$$\mathcal{B}^{\pi_e}_{P,v}F := \sum_{n=0}^{m_{\max}} v_n(\mathcal{B}^{\pi_e}_{P,1})^n F, \tag{56}$$

so that $v_n = \mathbf{1}\{n = m\}$ recovers $\mathcal{B}^{\pi_e}_{P,m}$.

Theorem C.12 (Full form of Theorem 3.2)**.** *$\mathcal{B}^{\pi_e}_{P,v}$ maps conditional probability laws to conditional probability laws on every finite MDP iff $v_n \ge 0$ for all n and $\sum_n v_n = 1$. For such v, set:*

$$\kappa(v) := \sum_{n=0}^{m_{\max}} v_n\gamma^n, \qquad \mathrm{cost}(v) := \sum_{n=0}^{m_{\max}} v_n\frac{1-\gamma^n}{1-\gamma}. \tag{57}$$

Then $\kappa(v)$ is the smallest uniform contraction modulus of $\mathcal{B}^{\pi_e}_{P,v}$ in D_∞; $\mathrm{cost}(v)$ is the smallest expected number of frozen-model calls of an exact sampler of $\mathcal{B}^{\pi_e}_{P,v}F$ in the opaque model of Appendix C.11; both are attained; and:

$$1 - \kappa(v) = (1-\gamma)\,\mathrm{cost}(v). \tag{58}$$

Proof. Step 1 (sufficiency). Each $(\mathcal{B}^{\pi_e}_{P,1})^n F$ is a conditional probability law; a convex combination of such is one.

Step 2 ($\sum_n v_n = 1$). $\big((\mathcal{B}^{\pi_e}_{P,1})^n F\big)(\mathcal{E} \mid x) = 1$ for all n, x, so:

$$(\mathcal{B}^{\pi_e}_{P,v}F)(\mathcal{E} \mid x) = \sum_{n=0}^{m_{\max}} v_n, \tag{59}$$

which must equal 1.

Step 3 ($v_n \ge 0$). Take roots $y_0, \dots, y_{m_{\max}}$, one action, deterministic transitions $y_0 \to y_1 \to \cdots \to y_{m_{\max}}$, distinct atoms $e_j \in \mathcal{E}$ emitted at y_j, distinct atoms $z_j \notin \{e_i\}$, and $F(\cdot \mid y_j) := \delta_{z_j}$. Then $(\mathcal{B}^{\pi_e}_{P,1})^0 F(\cdot \mid y_0) = \delta_{z_0}$ and, by Lemma C.6:

$$(\mathcal{B}^{\pi_e}_{P,1})^n F(\cdot \mid y_0) = (1-\gamma)\sum_{j=0}^{n-1}\gamma^j\delta_{e_j} + \gamma^n\delta_{z_n}, \qquad n \ge 1. \tag{60}$$

The atom z_n appears only in the nth power, so:

$$(\mathcal{B}^{\pi_e}_{P,v}F)(\{z_0\} \mid y_0) = v_0, \qquad (\mathcal{B}^{\pi_e}_{P,v}F)(\{z_n\} \mid y_0) = v_n\gamma^n \quad (1 \le n \le m_{\max}). \tag{61}$$

Nonnegativity of these masses and $\gamma^n > 0$ give $v_n \ge 0$.

Step 4 ($\kappa(v)$ is a modulus). By Lemma C.2 and Proposition 3.1:

$$D_\infty(\mathcal{B}^{\pi_e}_{P,v}F, \mathcal{B}^{\pi_e}_{P,v}H) \le \sum_{n=0}^{m_{\max}} v_n D_\infty\big((\mathcal{B}^{\pi_e}_{P,1})^n F, (\mathcal{B}^{\pi_e}_{P,1})^n H\big) \le \kappa(v)\, D_\infty(F, H). \tag{62}$$

Step 5 (attained). On the chain of Step 3 let $F(\cdot \mid y_n) = \delta_{u_n}$, $H(\cdot \mid y_n) = \delta_{w_n}$ with all $2(m_{\max}+1)$ atoms distinct and distinct from $\{e_j\}$. Then $D_\infty(F, H) = 1$ and, by Equation (60) (also at $n = 0$):

$$(\mathcal{B}^{\pi_e}_{P,1})^n F(\cdot \mid y_0) - (\mathcal{B}^{\pi_e}_{P,1})^n H(\cdot \mid y_0) = \gamma^n(\delta_{u_n} - \delta_{w_n}), \tag{63}$$

$$\mathcal{B}^{\pi_e}_{P,v} F(\cdot \mid y_0) - \mathcal{B}^{\pi_e}_{P,v} H(\cdot \mid y_0) = \sum_{n=0}^{m_{\max}} v_n \gamma^n(\delta_{u_n} - \delta_{w_n}). \tag{64}$$

The supremum defining TV is attained at $\{u_0, \dots, u_{m_{\max}}\}$ and equals $\kappa(v)$.

Step 6 (sampling one power). Let $\Pr(\tau = j) = (1-\gamma)\gamma^j$, $j \geq 0$. If $\tau < n$, generate $\tau+1$ transitions and return $(s_\tau, a_\tau, s_{\tau+1})$; otherwise generate n transitions and return a draw of $F(\cdot \mid s_n, a_n)$. By Equation (35) the output law is $(\mathcal{B}^{\pi_e}_{P,1})^n F(\cdot \mid x)$, with $\min(\tau+1, n)$ calls and:

$$\mathbb{E}[\min(\tau+1, n)] = \sum_{i=1}^{n} \Pr(\min(\tau+1, n) \geq i) = \sum_{i=1}^{n} \Pr(\tau \geq i-1) = \sum_{j=0}^{n-1} \gamma^j = \frac{1-\gamma^n}{1-\gamma}, \tag{65}$$

which is 0 for $n = 0$.

Step 7 (sampling a mixture). Draw $n \sim v$, then apply Step 6. The expected cost is:

$$\sum_{n=0}^{m_{\max}} v_n \frac{1-\gamma^n}{1-\gamma} = \frac{1 - \sum_n v_n \gamma^n}{1-\gamma} = \frac{1-\kappa(v)}{1-\gamma} = \mathrm{cost}(v), \tag{66}$$

using $\sum_n v_n = 1$; rearranging gives Equation (58).

Step 8 (optimality). On the chain of Step 3, opacity implies that e_j cannot be output before query $j+1$ and z_n not before query n. For an exact sampler, let N denote its number of queries. For $1 \leq i \leq m_{\max}$, by Equation (60):

$$\Pr(N \geq i) \geq \sum_{j=i-1}^{m_{\max}-1} (1-\gamma)\gamma^j \sum_{n=j+1}^{m_{\max}} v_n + \sum_{n=i}^{m_{\max}} v_n \gamma^n. \tag{67}$$

Exchanging sums:

$$\sum_{j=i-1}^{m_{\max}-1} (1-\gamma)\gamma^j \sum_{n=j+1}^{m_{\max}} v_n = \sum_{n=i}^{m_{\max}} v_n \sum_{j=i-1}^{n-1} (1-\gamma)\gamma^j = \sum_{n=i}^{m_{\max}} v_n(\gamma^{i-1} - \gamma^n), \tag{68}$$

hence:

$$\Pr(N \geq i) \geq \gamma^{i-1} \sum_{n=i}^{m_{\max}} v_n, \tag{69}$$

and:

$$\mathbb{E}N = \sum_{i \geq 1} \Pr(N \geq i) \geq \sum_{i=1}^{m_{\max}} \gamma^{i-1} \sum_{n=i}^{m_{\max}} v_n = \sum_{n=0}^{m_{\max}} v_n \sum_{i=1}^{n} \gamma^{i-1} = \mathrm{cost}(v). \tag{70}$$

Step 7 attains this value.

Step 9 (hard depth budget). If $v_n = 0$ for $n > m$, then $\gamma^n \geq \gamma^m$ for $n \leq m$ and:

$$\kappa(v) = \sum_{n=0}^{m} v_n \gamma^n \geq \gamma^m \sum_{n=0}^{m} v_n = \gamma^m, \tag{71}$$

with equality iff $v_n = \mathbf{1}\{n = m\}$. Hence $\mathcal{B}^{\pi_e}_{P,m}$ is contraction-optimal at depth budget m. □

Corollary C.13 (Signed coefficients leave the class)**.** *If $v_n < 0$ for some n, there exist a finite MDP and F such that $\mathcal{B}^{\pi_e}_{P,v} F$ is not a probability law. In particular, signed accelerations (Chebyshev, extrapolation) do not yield clean denoising targets without a representation of signed measures.*

Proof. In Step 3 of Theorem C.12, $(\mathcal{B}^{\pi_e}_{P,v} F)(\{z_n\} \mid y_0) = v_n \gamma^n < 0$. □

C.7 The Bellman score equation

Fix $\widehat{P}$, m, F and $k \in \{1, \dots, K\}$; let $q_k(x_k \mid x_0)$ be the forward density of Equation (2). Define:

$$p_k^F(x_k \mid x) := \int q_k(x_k \mid x_0)\, F(\mathrm{d}x_0 \mid x), \tag{72}$$

$$p_{j,k}^{\mathrm{pre}}(x_k \mid x) := \int q_k(x_k \mid x_0)\, G_{\widehat{P},j}^{\pi_e}(\mathrm{d}x_0 \mid x), \qquad j = 0, \dots, m-1. \tag{73}$$

Lemma C.14 (Noising preserves the block mixture)**.** *The noised law of $\mathcal{B}_{\widehat{P},m}^{\pi_e} F$ is:*

$$p_{m,k}^{\mathrm{tgt},F}(x_k \mid x) = (1-\gamma) \sum_{j=0}^{m-1} \gamma^j p_{j,k}^{\mathrm{pre}}(x_k \mid x) + \gamma^m \int p_k^F(x_k \mid x')\, (T_{\widehat{P}}^{\pi_e})^m(\mathrm{d}x' \mid x). \tag{74}$$

Proof. By Equations (19) and (21):

$$(\mathcal{B}_{\widehat{P},m}^{\pi_e} F)(\mathrm{d}x_0 \mid x) = (1-\gamma) \sum_{j=0}^{m-1} \gamma^j G_{\widehat{P},j}^{\pi_e}(\mathrm{d}x_0 \mid x) + \gamma^m \int F(\mathrm{d}x_0 \mid x')\, (T_{\widehat{P}}^{\pi_e})^m(\mathrm{d}x' \mid x). \tag{75}$$

Integrate $q_k(x_k \mid x_0)$ against this measure; linearity and Tonelli's theorem distribute the integral over both terms. Apply Equations (72) and (73). □

Lemma C.14 is Equation (12). Where the densities are positive and differentiable in x_k, set:

$$s_k^F := \nabla_{x_k} \log p_k^F, \qquad s_{j,k}^{\mathrm{pre}} := \nabla_{x_k} \log p_{j,k}^{\mathrm{pre}}, \qquad s_{m,k}^{\mathrm{tgt},F} := \nabla_{x_k} \log p_{m,k}^{\mathrm{tgt},F}, \tag{76}$$

and define the posterior branch weights:

$$\varpi_{j,k}^F(x_k, x) := \frac{(1-\gamma)\gamma^j p_{j,k}^{\mathrm{pre}}(x_k \mid x)}{p_{m,k}^{\mathrm{tgt},F}(x_k \mid x)}, \qquad j = 0, \dots, m-1, \tag{77}$$

$$\varpi_{\mathrm{tail},k}^F(\mathrm{d}x' \mid x_k, x) := \frac{\gamma^m p_k^F(x_k \mid x')\, (T_{\widehat{P}}^{\pi_e})^m(\mathrm{d}x' \mid x)}{p_{m,k}^{\mathrm{tgt},F}(x_k \mid x)}. \tag{78}$$

For the branch variable ι of Equation (9), $\varpi_{j,k}^F = \Pr(\iota = j \mid x_k, x)$ and $\varpi_{\mathrm{tail},k}^F(\mathrm{d}x') = \Pr(\iota = m, (s_m, a_m) \in \mathrm{d}x' \mid x_k, x)$.

Theorem C.15 (Bellman score and denoiser equations; full form of Theorem 3.3)**.** *Assume $p_{j,k}^{\mathrm{pre}}(\cdot \mid x)$ $(j < m)$ and $p_k^F(\cdot \mid x')$ ($(T_{\widehat{P}}^{\pi_e})^m$-a.e. x') are positive and differentiable in x_k, $p_{m,k}^{\mathrm{tgt},F}(\cdot \mid x) > 0$, and ∇_{x_k} commutes with the sum and integral in Equation (74). Then the weights Equations (77) and (78) are nonnegative, sum to one, and:*

$$s_{m,k}^{\mathrm{tgt},F}(x_k, x) = \sum_{j=0}^{m-1} \varpi_{j,k}^F(x_k, x)\, s_{j,k}^{\mathrm{pre}}(x_k, x) + \int s_k^F(x_k, x')\, \varpi_{\mathrm{tail},k}^F(\mathrm{d}x' \mid x_k, x). \tag{79}$$

If $F = F_{\widehat{P}}^{\pi_e}$, then $s_{m,k}^{\mathrm{tgt},F} = s_k^F$. If moreover $x_k = x_0 + \sigma_k \varepsilon$, $\varepsilon \sim \mathcal{N}(0, I_d)$, $\sigma_k > 0$, and first moments are finite, then with $\eta_k^F(x_k, x) := \mathbb{E}[x_0 \mid x_k]$ under $x_0 \sim F(\cdot \mid x)$ and $\eta_{j,k}, \eta_{m,k}^{\mathrm{tgt},F}$ defined analogously:

$$\eta_{m,k}^{\mathrm{tgt},F}(x_k, x) = \sum_{j=0}^{m-1} \varpi_{j,k}^F(x_k, x)\, \eta_{j,k}(x_k, x) + \int \eta_k^F(x_k, x')\, \varpi_{\mathrm{tail},k}^F(\mathrm{d}x' \mid x_k, x). \tag{80}$$

Proof. Step 1 (weights). Nonnegativity is immediate. By Equation (74), the numerator below equals the denominator:

$$\sum_{j=0}^{m-1} \varpi_{j,k}^F + \int \varpi_{\mathrm{tail},k}^F(\mathrm{d}x') = \frac{(1-\gamma) \sum_{j<m} \gamma^j p_{j,k}^{\mathrm{pre}}(x_k \mid x) + \gamma^m \big((\mathcal{C}_{\widehat{P}}^{\pi_e})^m p_k^F(x_k \mid \cdot)\big)(x)}{p_{m,k}^{\mathrm{tgt},F}(x_k \mid x)} = 1. \tag{81}$$

Step 2 (gradient). Using $\nabla p = p \nabla \log p$ on each component of Equation (74):

$$\nabla_{x_k} p^{\mathrm{tgt},F}_{m,k}(x_k \mid x) = (1-\gamma) \sum_{j=0}^{m-1} \gamma^j p^{\mathrm{pre}}_{j,k}(x_k \mid x)\, s^{\mathrm{pre}}_{j,k}(x_k, x) + \gamma^m \int p^F_k(x_k \mid x')\, s^F_k(x_k, x')\, (T^{\pi_e}_{\widehat{P}})^m(\,\mathrm{d}x' \mid x). \tag{82}$$

Step 3 (normalize). Divide Equation (82) by $p^{\mathrm{tgt},F}_{m,k} > 0$; the coefficients become Equations (77) and (78), which is Equation (79).

Step 4 (fixed point). If $F = F^{\pi_e}_{\widehat{P}}$, then $\mathcal{B}^{\pi_e}_{\widehat{P},m} F = F$ by Proposition 3.1; equal clean laws have equal noised laws and scores.

Step 5 (denoiser). Tweedie's identity gives $\eta = x_k + \sigma_k^2 s$ for each of $\eta^{\mathrm{tgt},F}_{m,k}, \eta_{j,k}, \eta^F_k$. Substituting into Equation (79):

$$\eta^{\mathrm{tgt},F}_{m,k} - x_k = \sum_{j=0}^{m-1} \varpi^F_{j,k}(\eta_{j,k} - x_k) + \int \big(\eta^F_k(x_k, x') - x_k\big) \varpi^F_{\mathrm{tail},k}(\,\mathrm{d}x' \mid x_k, x), \tag{83}$$

and by Equation (81) the x_k terms cancel, giving Equation (80). □

Corollary C.16 (The clean coefficients are not the score coefficients)**.** *Let* $w_j := (1-\gamma)\gamma^j$. *The rule* $\sum_{j<m} w_j s^{\mathrm{pre}}_{j,k} + \gamma^m \int s^F_k(\cdot, x')\, (T^{\pi_e}_{\widehat{P}})^m(\,\mathrm{d}x' \mid x)$ *equals* $s^{\mathrm{tgt},F}_{m,k}$ *at* (x_k, x) *iff:*

$$\sum_{j=0}^{m-1} \big(\varpi^F_{j,k}(x_k, x) - w_j\big) s^{\mathrm{pre}}_{j,k}(x_k, x) + \int s^F_k(x_k, x') \Big[\varpi^F_{\mathrm{tail},k}(\,\mathrm{d}x' \mid x_k, x) - \gamma^m (T^{\pi_e}_{\widehat{P}})^m(\,\mathrm{d}x' \mid x)\Big] = 0. \tag{84}$$

Since $\varpi^F_{j,k}$ *depends on* x_k *and* w_j *does not, Equation (84) fails generically.*

Proof. Subtract the rule from Equation (79). □

Remark C.17 (Why BRD samples the target instead of evaluating it). Equation (79) requires the normalized densities $p^{\mathrm{pre}}_{j,k}, p^F_k$, which a score network does not provide. The minimizer of Equation (2) over samples drawn by Equation (9) is the score of the sampled law, namely $s^{\mathrm{tgt},F}_{m,k}$; the weights are accounted for implicitly.

Remark C.18 (Gaussian corruption makes the hypotheses automatic). Let $q_k(x_k \mid x_0) = \mathcal{N}(x_k; \alpha_k x_0, \sigma_k^2 I)$ with $\sigma_k > 0$, and let μ be any probability measure on $\mathcal{E}$. Then $p(x_k) := \int q_k(x_k \mid x_0)\, \mu(\,\mathrm{d}x_0)$ is positive everywhere, because q_k is. Moreover $\nabla_{x_k} q_k(x_k \mid x_0) = -\sigma_k^{-2}(x_k - \alpha_k x_0)\, q_k(x_k \mid x_0)$, and since $\sup_{u \geq 0} u\, e^{-u^2/(2\sigma_k^2)} = \sigma_k e^{-1/2}$:

$$\big\|\nabla_{x_k} q_k(x_k \mid x_0)\big\|_2 \leq (2\pi)^{-d/2} \sigma_k^{-d-1} e^{-1/2} \qquad \text{for all } x_k, x_0. \tag{85}$$

The gradient of the integrand is therefore bounded uniformly, and dominated convergence allows differentiation under the integral, both for μ and for the nested integrals in Equation (74). All hypotheses of Theorem C.15 thus hold for every F, which is why Theorem 3.3 carries no regularity conditions. No density of the clean law is required. In particular, $G^{\pi_e}_{\widehat{P},0}(\cdot \mid x)$ is concentrated on the slice $N_x := \{e = (\tilde{s}, \tilde{a}, \tilde{s}') : (\tilde{s}, \tilde{a}) = x\}$, yet its noised law is smooth and positive.

The singular branch does matter for total-variation bounds. In Algorithm 1 the training targets contain exact samples of every prefix branch, including $j = 0$, but the trained generator produces every coordinate of every output transition by reverse diffusion, so $F_Q(\cdot \mid x)$ has a Lebesgue density on $\mathcal{E}$ when $\mathcal{S} \times \mathcal{A}$ is continuous. The slice N_x is then Lebesgue-null, so $F_Q(N_x \mid x) = 0$, while the target places mass at least $1 - \gamma$ on it: $(\mathcal{B}^{\pi_e}_{\widehat{P},m} F_Q)(N_x \mid x) \geq 1 - \gamma$. Taking $A = N_x$ in the definition of total variation:

$$\mathrm{Res}_{F_Q}(x) \geq \big|(\mathcal{B}^{\pi_e}_{\widehat{P},m} F_Q)(N_x \mid x) - F_Q(N_x \mid x)\big| \geq 1 - \gamma \qquad \text{for every } x. \tag{86}$$

This lower bound holds however well the generator is trained. It is not a statement that the generator is inaccurate: the part of F_Q that corresponds to the $j{=}0$ branch sits near N_x, not on it, and total variation between a density and a measure concentrated on a null set equals one no matter how close the two are in transport distance. As a consequence $\varepsilon_{\text{gen}} \geq (1-\gamma)/(1-\gamma^m)$, and term (a) of Equation (17) is at least $2R_{\max}/(1-\gamma^m)$, compared with the value scale $R_{\max}/(1-\gamma)$. The floor is an artifact of total variation and does not appear in the Wasserstein analysis of Appendix C.12. It could be removed within total variation by emitting the $j{=}0$ branch in closed form, i.e. by parametrizing $F(\cdot \mid x) = (1-\gamma)G^{\pi_e}_{\widehat{P},0}(\cdot \mid x) + \gamma H(\cdot \mid x)$, which has the form of the fixed point by Equation (4), and fitting the diffusion model H to the remaining mass; our experiments do not do this.

C.8 The residual certificate and the optimal audit law

Let $R_F := F - \mathcal{B}^{\pi_e}_{\widehat{P},m} F$, so $\mathrm{Res}_F(x) = \mathrm{TV}(R_F(\cdot \mid x))$, and:

$$\nu_m := \sum_{i=0}^{\infty} \gamma^{mi}\, d_0^{\pi_e} (T^{\pi_e}_{\widehat{P}})^{mi}, \qquad \beta^\star_m = (1-\gamma^m)\nu_m \ \text{ by Equation (10).} \tag{87}$$

Lemma C.19 (Residual resolvent)**.** *For every conditional probability law F, with convergence in* $\sup_x \mathrm{TV}$*:*

$$F - F^{\pi_e}_{\widehat{P}} = \sum_{i=0}^{\infty} \gamma^{mi} (\mathcal{C}^{\pi_e}_{\widehat{P}})^{mi} R_F. \tag{88}$$

Proof. Subtract $F^{\pi_e}_{\widehat{P}} = \mathcal{B}^{\pi_e}_{\widehat{P},m} F^{\pi_e}_{\widehat{P}}$ from $F = R_F + \mathcal{B}^{\pi_e}_{\widehat{P},m} F$; the constant parts of Equation (21) cancel:

$$F - F^{\pi_e}_{\widehat{P}} = R_F + \gamma^m (\mathcal{C}^{\pi_e}_{\widehat{P}})^m (F - F^{\pi_e}_{\widehat{P}}). \tag{89}$$

Iterating n times:

$$F - F^{\pi_e}_{\widehat{P}} = \sum_{i=0}^{n-1} \gamma^{mi} (\mathcal{C}^{\pi_e}_{\widehat{P}})^{mi} R_F + \gamma^{mn} (\mathcal{C}^{\pi_e}_{\widehat{P}})^{mn} (F - F^{\pi_e}_{\widehat{P}}). \tag{90}$$

By Lemma C.1 and Equation (19) and $\mathrm{TV}(F - F^{\pi_e}_{\widehat{P}}) \leq 1$:

$$\sup_x \mathrm{TV}\Big(\gamma^{mn} (\mathcal{C}^{\pi_e}_{\widehat{P}})^{mn} (F - F^{\pi_e}_{\widehat{P}})(\cdot \mid x)\Big) \leq \gamma^{mn} \xrightarrow[n\to\infty]{} 0. \tag{91}$$

□

Theorem C.20 (Residual certificate and minimax-optimal audit law)**.** *For every conditional probability law F:*

$$D_{d_0^{\pi_e}}(F, F^{\pi_e}_{\widehat{P}}) \leq \frac{1}{1-\gamma^m} \int \mathrm{Res}_F(x)\, \beta^\star_m(\,\mathrm{d}x). \tag{92}$$

For probability measures β on $\mathcal{X}$, let:

$$C(\beta) := \inf\Big\{ C \geq 0 : \int \psi \,\mathrm{d}\nu_m \leq C \int \psi \,\mathrm{d}\beta \ \ \forall\, \psi \geq 0 \text{ measurable} \Big\}. \tag{93}$$

Then $\beta^\star_m$ is the unique minimizer of C, $C(\beta^\star_m) = (1-\gamma^m)^{-1}$, and Equation (92) holds with equality on some instance.

Proof. Step 1 (bound). Take TV in Equation (88) at x; by the triangle inequality and Lemma C.2 with mixing measures $(T^{\pi_e}_{\widehat{P}})^{mi}(\cdot \mid x)$:

$$\mathrm{TV}\big(F(\cdot \mid x), F^{\pi_e}_{\widehat{P}}(\cdot \mid x)\big) \leq \sum_{i=0}^{\infty} \gamma^{mi} \int \mathrm{Res}_F(x')\, (T^{\pi_e}_{\widehat{P}})^{mi}(\,\mathrm{d}x' \mid x). \tag{94}$$

Integrating against $d_0^{\pi_e}$ and using Equation (87):

$$D_{d_0^{\pi_e}}(F, F^{\pi_e}_{\widehat{P}}) \leq \sum_{i=0}^{\infty} \gamma^{mi} \int \mathrm{Res}_F \ \mathrm{d}\big(d_0^{\pi_e} (T^{\pi_e}_{\widehat{P}})^{mi}\big) = \int \mathrm{Res}_F \ \mathrm{d}\nu_m. \tag{95}$$

Since $\nu_m(\mathcal{X}) = \sum_i \gamma^{mi} = (1-\gamma^m)^{-1}$, $\beta_m^\star$ is a probability measure and Equation (95) is Equation (92).

Step 2 (C as a density ratio). If $\nu_m \not\ll \beta$, choose A with $\beta(A) = 0 < \nu_m(A)$ and $\psi = \mathbf{1}_A$: $C(\beta) = \infty$. Otherwise let $\chi := \mathrm{d}\nu_m/\,\mathrm{d}\beta$. For $\psi \geq 0$, $\int \psi \,\mathrm{d}\nu_m = \int \psi\chi \,\mathrm{d}\beta \leq \|\chi\|_{L_\infty(\beta)} \int \psi \,\mathrm{d}\beta$, so $C(\beta) \leq \|\chi\|_{L_\infty(\beta)}$. With $\chi_{\max} := \operatorname{ess\,sup}_\beta \chi$, $\epsilon > 0$ and $A_\epsilon := \{\chi > \chi_{\max} - \epsilon\}$, $\beta(A_\epsilon) > 0$ and:

$$\frac{\int \mathbf{1}_{A_\epsilon} \,\mathrm{d}\nu_m}{\int \mathbf{1}_{A_\epsilon} \,\mathrm{d}\beta} = \frac{1}{\beta(A_\epsilon)} \int_{A_\epsilon} \chi \,\mathrm{d}\beta \geq \chi_{\max} - \epsilon. \tag{96}$$

Letting $\epsilon \downarrow 0$:

$$C(\beta) = \left\| \frac{\mathrm{d}\nu_m}{\mathrm{d}\beta} \right\|_{L_\infty(\beta)}. \tag{97}$$

Step 3 (lower bound). $\int \chi \,\mathrm{d}\beta = \nu_m(\mathcal{X}) = (1-\gamma^m)^{-1}$, and an essential supremum dominates the mean:

$$C(\beta) \geq \frac{1}{1-\gamma^m} \qquad \forall \beta. \tag{98}$$

Step 4 (attainment). For $\beta = \beta_m^\star$, $\chi \equiv (1-\gamma^m)^{-1}$, so $C(\beta_m^\star) = (1-\gamma^m)^{-1}$.

Step 5 (uniqueness). If $C(\beta) = (1-\gamma^m)^{-1}$, then $\chi \leq (1-\gamma^m)^{-1} = \mathbb{E}_\beta \chi$ β-a.s., hence $\chi = (1-\gamma^m)^{-1}$ β-a.s., i.e. $\nu_m = (1-\gamma^m)^{-1}\beta$ and $\beta = \beta_m^\star$.

Step 6 (tightness). Let $\mathcal{X}$ be a single root, so $(T^{\pi_e}_{\widehat{P}})^m = \mathrm{I}$ and $d_0^{\pi_e} = \beta_m^\star$. For a fixed law Λ let $\mathcal{B}^{\pi_e}_{\widehat{P},m} F = (1-\gamma^m)\Lambda + \gamma^m F$. Then $F^{\pi_e}_{\widehat{P}} = \Lambda$, $R_F = (1-\gamma^m)(F-\Lambda)$, $\mathrm{Res}_F = (1-\gamma^m)\,\mathrm{TV}(F,\Lambda)$, and $\mathrm{TV}(F, F^{\pi_e}_{\widehat{P}}) = (1-\gamma^m)^{-1}\,\mathrm{Res}_F$. □

With $F = F_Q$, $\varepsilon_{\mathrm{res}} = \mathbb{E}_{\beta_m^\star}[\mathrm{Res}_{F_Q}]$ and $\varepsilon_{\mathrm{gen}} = \varepsilon_{\mathrm{res}}/(1-\gamma^m)$ (Section 3.4), Equation (92) is Equation (16), the second claim of Theorem 3.5.

Lemma C.21 (Sampling $\beta_m^\star$)**.** *Let $\tau \sim \mathrm{Geom}(1-\gamma^m)$ on $\{0,1,\ldots\}$ be independent, draw $(s_0,a_0) \sim d_0^{\pi_e}$, and roll out under $\widehat{P}, \pi_e$ for $m\tau$ steps. Then $(s_{m\tau}, a_{m\tau}) \sim \beta_m^\star$.*

Proof. $\Pr(\tau = i) = (1-\gamma^m)\gamma^{mi}$ and $\mathrm{Law}(s_{mi}, a_{mi}) = d_0^{\pi_e}(T^{\pi_e}_{\widehat{P}})^{mi}$, so:

$$\mathrm{Law}(s_{m\tau}, a_{m\tau}) = \sum_{i=0}^{\infty} (1-\gamma^m)\gamma^{mi}\, d_0^{\pi_e}(T^{\pi_e}_{\widehat{P}})^{mi} = (1-\gamma^m)\nu_m = \beta_m^\star. \tag{99}$$

□

The audit distribution $\beta_m^\star$ is optimal for converting residuals into error, but Algorithm 1 trains on $\beta_\lambda := \lambda d_0^{\pi_e} + (1-\lambda)\beta_m^\star$. The next result explains what the training distribution must provide. At sweep ℓ, the target at a root $x \sim \beta_\lambda$ queries the frozen generator $F_{\ell-1}$ at $x' \sim (T^{\pi_e}_{\widehat{P}})^m(\cdot \mid x)$ on the tail branch, while $F_{\ell-1}$ was trained on roots drawn from β_λ. The error inherited through the tail is controlled only if the query distribution $\beta_\lambda(T^{\pi_e}_{\widehat{P}})^m$ has a bounded density ratio with respect to β_λ.

Proposition C.22 (Training roots cover bootstrap queries)**.** *Write $T := T^{\pi_e}_{\widehat{P}}$. For every $\lambda \in [0,1)$ and every measurable $B \subseteq \mathcal{X}$:*

$$\big(\beta_\lambda T^m\big)(B) \leq c_\lambda\, \beta_\lambda(B), \qquad c_\lambda := \frac{1}{\gamma^m}\Big(1 + \frac{\lambda}{(1-\lambda)(1-\gamma^m)}\Big). \tag{100}$$

Consequently, for every measurable $\psi \geq 0$ on $\mathcal{X}$, such as the pointwise error of $F_{\ell-1}$:

$$\mathbb{E}_{x\sim\beta_\lambda} \int \psi(x')\, T^m(\,\mathrm{d}x' \mid x) \leq c_\lambda\, \mathbb{E}_{x\sim\beta_\lambda} \psi(x). \tag{101}$$

For $\lambda = 1$, no finite constant exists in general.

Proof. Step 1 (continuation roots). By Equation (87):

$$\beta^\star_m T^m = (1-\gamma^m)\sum_{i=0}^{\infty}\gamma^{mi} d_0^{\pi_e} T^{m(i+1)} = \frac{1-\gamma^m}{\gamma^m}\sum_{i=1}^{\infty}\gamma^{mi} d_0^{\pi_e} T^{mi}$$
$$= \frac{1}{\gamma^m}\big(\beta^\star_m - (1-\gamma^m) d_0^{\pi_e}\big) \le \frac{1}{\gamma^m}\beta^\star_m. \tag{102}$$

Step 2 (initial roots). The $i=1$ term of Equation (87) gives $\beta^\star_m \ge (1-\gamma^m)\gamma^m\, d_0^{\pi_e} T^m$, so:

$$d_0^{\pi_e} T^m \le \frac{1}{(1-\gamma^m)\gamma^m}\beta^\star_m. \tag{103}$$

Step 3 (combine). By Equations (102) and (103):

$$\beta_\lambda T^m = \lambda\, d_0^{\pi_e} T^m + (1-\lambda)\beta^\star_m T^m \le \Big(\frac{\lambda}{(1-\gamma^m)\gamma^m} + \frac{1-\lambda}{\gamma^m}\Big)\beta^\star_m, \tag{104}$$

and $\beta^\star_m \le \beta_\lambda/(1-\lambda)$ gives Equation (100). The left side of Equation (101) is $\int \psi\ \mathrm{d}(\beta_\lambda T^m)$, so Equation (101) follows from Equation (100) for indicator functions, then for simple functions by linearity, and for general $\psi \ge 0$ by monotone convergence.

Step 4 ($\lambda = 1$). Take a deterministic chain $y_0 \to y_1 \to \cdots$ with distinct roots and $d_0^{\pi_e} = \delta_{y_0}$. Then $\beta_1 T^m = \delta_{y_m}$ while $\beta_1(\{y_m\}) = 0$. □

For $\gamma = 0.99$ and $m = 64$, $c_0 \approx 1.9$ and $c_{1/2} \approx 5.9$: every $\lambda < 1$ keeps the bootstrap queries within the training distribution, and λ trades this constant against the extra weight on the initial roots at which equation 5 evaluates the generator.

Corollary C.23 (Auditing under the training distribution)**.** *For every conditional probability law F and every $\lambda \in [0,1)$:*

$$D_{d_0^{\pi_e}}(F, F^{\pi_e}_{\widehat{P}}) \le \frac{1}{(1-\lambda)(1-\gamma^m)}\int \mathrm{Res}_F(x)\,\beta_\lambda(\mathrm{d}x). \tag{105}$$

Proof. By Equation (95), $D_{d_0^{\pi_e}}(F, F^{\pi_e}_{\widehat{P}}) \le \int \mathrm{Res}_F\ \mathrm{d}\nu_m$, and $\nu_m = \beta^\star_m/(1-\gamma^m) \le \beta_\lambda/\big((1-\lambda)(1-\gamma^m)\big)$. □

A residual measured on training roots therefore also certifies the generator, at the price of an extra factor $1/(1-\lambda)$: in the notation of Theorem C.20, $C(\beta_\lambda) \le 1/\big((1-\lambda)(1-\gamma^m)\big)$.

C.9 The denoising loss bounds the certificate

Theorem C.20 reduces $\varepsilon_{\mathrm{gen}}$ to $\mathbb{E}_{\beta^\star_m}[\mathrm{Res}_{F_Q}]$. We bound it by the training objective for a general forward chain, covering continuous, categorical and mixed data.

Fix x and F, set $\mu_x := (\mathcal{B}^{\pi_e}_{\widehat{P},m}F)(\cdot \mid x)$, and let $q_k(\mathrm{d}z_k \mid z_{k-1})$, $k = 1,\dots,K$, be forward kernels with:

$$z_0 \sim \mu_x, \qquad z_k \sim q_k(\cdot \mid z_{k-1}), \qquad \mu_{x,k} := \mathrm{Law}(z_k). \tag{106}$$

The forward path law is:

$$\mathbb{P}_x(\mathrm{d}z_{0:K}) := \mu_x(\mathrm{d}z_0)\prod_{k=1}^{K} q_k(\mathrm{d}z_k \mid z_{k-1}), \tag{107}$$

which on standard Borel spaces factorizes backwards through the exact reverse kernels R_k:

$$\mathbb{P}_x(\mathrm{d}z_{0:K}) = \mu_{x,K}(\mathrm{d}z_K)\prod_{k=K}^{1}\mathsf{R}_k(\mathrm{d}z_{k-1} \mid z_k, x). \tag{108}$$

The model path law is:

$$\mathbb{Q}_x(\,\mathrm{d}z_{0:K}) := p_K(\,\mathrm{d}z_K \mid x) \prod_{k=K}^{1} \mathsf{R}_{\theta,k}(\,\mathrm{d}z_{k-1} \mid z_k, x), \tag{109}$$

whose z_0-marginal is $F(\cdot \mid x)$.

Proposition C.24 (Reverse-path certificate)**.** *Assume $\mathbb{P}_x \ll \mathbb{Q}_x$ and let:*

$$\mathrm{KL}_F^{\mathrm{path}}(x) := \mathrm{KL}\big(\mu_{x,K} \,\|\, p_K(\cdot \mid x)\big) + \sum_{k=1}^{K} \mathbb{E}_{z_k \sim \mu_{x,k}} \mathrm{KL}\big(\mathsf{R}_k(\cdot \mid z_k, x) \,\|\, \mathsf{R}_{\theta,k}(\cdot \mid z_k, x)\big). \tag{110}$$

Then $\mathrm{Res}_F(x) \le \sqrt{\frac{1}{2}\,\mathrm{KL}_F^{\mathrm{path}}(x)}$ *and:*

$$D_{d_0^{\pi_e}}(F, F_{\widehat{P}}^{\pi_e}) \le \frac{1}{1-\gamma^m} \sqrt{\frac{1}{2} \int \mathrm{KL}_F^{\mathrm{path}}(x)\, \beta_m^\star(\,\mathrm{d}x)}. \tag{111}$$

Proof. Step 1 (path KL). By Equations (108) and (109):

$$\log \frac{\mathrm{d}\mathbb{P}_x}{\mathrm{d}\mathbb{Q}_x}(z_{0:K}) = \log \frac{\mathrm{d}\mu_{x,K}}{\mathrm{d}p_K}(z_K) + \sum_{k=1}^{K} \log \frac{\mathrm{d}\mathsf{R}_k(\cdot \mid z_k, x)}{\mathrm{d}\mathsf{R}_{\theta,k}(\cdot \mid z_k, x)}(z_{k-1}). \tag{112}$$

Taking $\mathbb{E}_{\mathbb{P}_x}$ and conditioning the kth term on $z_k \sim \mu_{x,k}$ gives $\mathrm{KL}(\mathbb{P}_x \| \mathbb{Q}_x) = \mathrm{KL}_F^{\mathrm{path}}(x)$.

Step 2 (endpoint). $z_{0:K} \mapsto z_0$ pushes $\mathbb{P}_x \mapsto \mu_x$ and $\mathbb{Q}_x \mapsto F(\cdot \mid x)$; by data processing:

$$\mathrm{KL}\big(\mu_x \,\|\, F(\cdot \mid x)\big) \le \mathrm{KL}(\mathbb{P}_x \| \mathbb{Q}_x) = \mathrm{KL}_F^{\mathrm{path}}(x). \tag{113}$$

Step 3 (Pinsker). With $\mathrm{TV}(\mu, \nu) = \sup_A |\mu(A) - \nu(A)|$:

$$\mathrm{Res}_F(x) = \mathrm{TV}\big(F(\cdot \mid x), \mu_x\big) \le \sqrt{\tfrac{1}{2}\,\mathrm{KL}\big(\mu_x \| F(\cdot \mid x)\big)} \le \sqrt{\tfrac{1}{2}\,\mathrm{KL}_F^{\mathrm{path}}(x)}. \tag{114}$$

Step 4 (average). By Jensen for $\sqrt{\cdot}$:

$$\int \mathrm{Res}_F \;\mathrm{d}\beta_m^\star \le \int \sqrt{\tfrac{1}{2}\,\mathrm{KL}_F^{\mathrm{path}}}\;\mathrm{d}\beta_m^\star \le \sqrt{\frac{1}{2}\int \mathrm{KL}_F^{\mathrm{path}}\;\mathrm{d}\beta_m^\star}. \tag{115}$$

Substitute into Equation (92). □

Lemma C.25 (The denoising objective dominates the reverse-kernel error)**.** *For $k \ge 2$, let $q_k^{\mathrm{post}}(\,\mathrm{d}z_{k-1} \mid z_k, z_0)$ be the forward posterior under $\mathbb{P}_x$. Then:*

$$\begin{aligned}\mathbb{E}_{\mathbb{P}_x} \mathrm{KL}\big(q_k^{\mathrm{post}}(\cdot \mid z_k, z_0) \,\|\, \mathsf{R}_{\theta,k}(\cdot \mid z_k, x)\big) &= \mathbb{E}_{z_k \sim \mu_{x,k}} \mathrm{KL}\big(\mathsf{R}_k(\cdot \mid z_k, x) \,\|\, \mathsf{R}_{\theta,k}(\cdot \mid z_k, x)\big) \\ &\quad + I_{\mathbb{P}_x}(z_0; z_{k-1} \mid z_k).\end{aligned} \tag{116}$$

In particular the left side dominates the reverse-kernel KL, with a θ-independent gap.

Proof. Under the joint law of (z_0, z_{k-1}, z_k):

$$\mathbb{E} \log \frac{q_k^{\mathrm{post}}(z_{k-1} \mid z_k, z_0)}{\mathsf{R}_{\theta,k}(z_{k-1} \mid z_k, x)} = \mathbb{E} \log \frac{q_k^{\mathrm{post}}(z_{k-1} \mid z_k, z_0)}{\mathsf{R}_k(z_{k-1} \mid z_k, x)} + \mathbb{E} \log \frac{\mathsf{R}_k(z_{k-1} \mid z_k, x)}{\mathsf{R}_{\theta,k}(z_{k-1} \mid z_k, x)}. \tag{117}$$

Since $\mathsf{R}_k(\cdot \mid z_k, x) = \mathrm{Law}(z_{k-1} \mid z_k)$ and $q_k^{\mathrm{post}} = \mathrm{Law}(z_{k-1} \mid z_k, z_0)$, the first term is $I(z_0; z_{k-1} \mid z_k) \ge 0$; the second is the expected reverse-kernel KL. □

Remark C.26 (The raw variational loss is not a usable certificate). The left side of Equation (116), summed over k, is the variational denoising loss. Its θ-independent excess over the reverse-kernel divergence is $\sum_{k=2}^{K} I(z_0; z_{k-1} \mid z_k)$. Because $z_0 \to z_1 \to \cdots \to z_K$ is a Markov chain, $I(z_0; z_{k-1}, z_k) = I(z_0; z_{k-1})$, so by the chain rule $I(z_0; z_{k-1} \mid z_k) = I(z_0; z_{k-1}) - I(z_0; z_k)$ and the sum telescopes to $I(z_0; z_1) - I(z_0; z_K)$. Under Gaussian corruption this is not small: if z_0 has a density with finite entropy, then $I(z_0; z_1) = h(z_1) - \frac{d}{2}\log(2\pi e\sigma_1^2)$ grows like $\frac{d}{2}\log(1/\sigma_1^2)$ as $\sigma_1 \to 0$. Substituting the raw loss for the path divergence in Equation (111) is therefore valid but typically vacuous. The certificate requires the *excess* loss, which training reduces but whose value cannot be computed without the entropy of the target.

Corollary C.27 (The excess loss certifies the generator). *Let $\mathcal{L}_{\mathrm{exc}}(F_Q) := \int \mathrm{KL}^{\mathrm{path}}_{F_Q}(x)\, \beta^\star_m(\mathrm{d}x)$ be the $\beta^\star_m$-average of the path divergence Equation (110) on the block targets $\mathcal{B}^{\pi_e}_{\widehat{P},m} F_Q(\cdot \mid x)$ of F_Q itself, which are the targets of one further sweep. Holding these targets fixed, as a further sweep does, Lemma C.25 and Remark C.26 show that it equals the averaged variational denoising loss minus terms that depend on the targets but not on the denoiser. Then:*

$$\varepsilon_{\mathrm{gen}} \le \frac{1}{1-\gamma^m}\sqrt{\tfrac{1}{2}\,\mathcal{L}_{\mathrm{exc}}(F_Q)}. \tag{118}$$

Proof. Step 4 of the proof of Proposition C.24 with $F = F_Q$ gives $\int \mathrm{Res}_{F_Q}\ \mathrm{d}\beta^\star_m \le \sqrt{\frac{1}{2}\mathcal{L}_{\mathrm{exc}}(F_Q)}$; divide by $1-\gamma^m$ and use $\varepsilon_{\mathrm{gen}} = (1-\gamma^m)^{-1}\int \mathrm{Res}_{F_Q}\ \mathrm{d}\beta^\star_m$. □

Remark C.28 (The first reverse step). For continuous data the $k = 1$ term is a reconstruction likelihood rather than a KL from a degenerate posterior. Step 1 of Proposition C.24 is unaffected; Lemma C.25 is used only for $k \ge 2$.

Corollary C.29 (Continuous-time form). *Let $\mathrm{d}z_t = f_t(z_t)\,\mathrm{d}t + g_t\,\mathrm{d}W_t$ on $[0,T]$ with $g_t > 0$, let the exact and model reverse processes satisfy Girsanov's conditions, and let $s_t^{\mu_x}, s_\theta$ be the exact and model scores. Then:*

$$\mathrm{KL}\big(\mu_x \| F(\cdot \mid x)\big) \le \mathrm{KL}\big(\mu_{x,T} \| p_T(\cdot \mid x)\big) + \frac{1}{2}\int_0^T g_t^2\, \mathbb{E}_{z_t \sim \mu_{x,t}} \big\| s_\theta(z_t, t, x) - s_t^{\mu_x}(z_t, x)\big\|_2^2\ \mathrm{d}t. \tag{119}$$

Proof. The reverse SDEs share diffusion g_t and have drift difference $g_t^2(s_\theta - s_t^{\mu_x})$. Girsanov gives $\mathrm{KL}(\mathrm{path}) = \mathrm{KL}(\mu_{x,T}\|p_T) + \frac{1}{2}\int_0^T g_t^{-2}\,\mathbb{E}\|g_t^2(s_\theta - s_t^{\mu_x})\|^2\,\mathrm{d}t$, and $g_t^{-2} g_t^4 = g_t^2$. Data processing to z_0 gives the left side. □

The integral in Equation (119) is the excess of the weighted denoising score matching objective over its minimum: the objective differs from it by a θ-independent constant, which, as in Remark C.26, is not small.

C.10 The end-to-end OPE bound

Lemma C.30 (Clipping does not worsen the reward error). *If $\|r\|_\infty \le R_{\max}$ and $\|\hat{r} - r\|_\infty \le \varepsilon_r$, then $\bar{r} := \mathrm{clip}(\hat{r}, -R_{\max}, R_{\max})$ satisfies:*

$$\|\bar{r} - r\|_\infty \le \varepsilon_r, \qquad \|\bar{r}\|_\infty \le R_{\max}. \tag{120}$$

Proof. $\mathrm{clip} = \Pi := \Pi_{[-R_{\max}, R_{\max}]}$ is 1-Lipschitz and $\Pi r = r$, so:

$$|\bar{r} - r| = |\Pi\hat{r} - \Pi r| \le |\hat{r} - r| \le \varepsilon_r. \tag{121}$$

The range bound holds by construction. □

Proof of the third claim of Theorem 3.5. Define:

$$J_1 := \frac{1}{1-\gamma} \iint \bar{r}(e)\, F_Q(\mathrm{d}e \mid x)\, d_0^{\pi_e}(\mathrm{d}x), \qquad \text{(generator, learned reward)} \tag{122}$$

$$J_2 := \frac{1}{1-\gamma} \iint r(e)\, F_Q(\mathrm{d}e \mid x)\, d_0^{\pi_e}(\mathrm{d}x), \qquad \text{(generator, true reward)} \tag{123}$$

$$J_3 := \frac{1}{1-\gamma} \iint r(e)\, F_{\widehat{P}}^{\pi_e}(\mathrm{d}e \mid x)\, d_0^{\pi_e}(\mathrm{d}x) = J(\pi_e, \widehat{P}), \qquad \text{(learned-model occupancy)} \tag{124}$$

$$J_4 := J(\pi_e) = J(\pi_e, P^\star), \qquad \text{(true value)} \tag{125}$$

the identities for J_3, J_4 by Equation (5). Then:

$$|\widehat{J}(\pi_e) - J(\pi_e)| \le |\widehat{J}(\pi_e) - J_1| + |J_1 - J_2| + |J_2 - J_3| + |J_3 - J_4|. \tag{126}$$

Term 1 (Monte Carlo). The summands $\bar{r}(\tilde{s}_i, \tilde{a}_i, \tilde{s}'_i)$ in Equation (11) are i.i.d. with mean $(1-\gamma)J_1$ and, by Lemma C.30, range in an interval of width $2R_{\max}$. By Hoeffding, with probability $\ge 1 - \delta_{\mathrm{MC}}$:

$$\Big| \frac{1}{M} \sum_{i=1}^{M} \bar{r}(\tilde{s}_i, \tilde{a}_i, \tilde{s}'_i) - (1-\gamma)J_1 \Big| \le 2R_{\max} \sqrt{\frac{\log(2/\delta_{\mathrm{MC}})}{2M}}, \tag{127}$$

hence:

$$|\widehat{J}(\pi_e) - J_1| \le \frac{2R_{\max}}{1-\gamma} \sqrt{\frac{\log(2/\delta_{\mathrm{MC}})}{2M}}. \tag{128}$$

Term 2 (reward). Same measure, different integrand; by Lemma C.30:

$$|J_1 - J_2| \le \frac{1}{1-\gamma} \sup_e |\bar{r}(e) - r(e)| \le \frac{\varepsilon_r}{1-\gamma}. \tag{129}$$

Term 3 (generative). Same integrand $r \in [-R_{\max}, R_{\max}]$, different measures; by Lemma C.3 at each x, averaging over $d_0^{\pi_e}$, and Equation (16):

$$|J_2 - J_3| \le \frac{2R_{\max}}{1-\gamma} D_{d_0^{\pi_e}}(F_Q, F_{\widehat{P}}^{\pi_e}) \le \frac{2R_{\max}}{1-\gamma} \varepsilon_{\mathrm{gen}}. \tag{130}$$

Term 4 (dynamics). As in Term 3 with $(F_{\widehat{P}}^{\pi_e}, F_{P^\star}^{\pi_e})$, $D_{d_0^{\pi_e}} \le D_\infty$, and Proposition 3.4:

$$|J_3 - J_4| \le \frac{2R_{\max}}{1-\gamma} D_\infty(F_{\widehat{P}}^{\pi_e}, F_{P^\star}^{\pi_e}) \le \frac{2R_{\max}}{(1-\gamma)^2} \varepsilon_{\mathrm{dyn}}. \tag{131}$$

Inserting Equations (128) to (131) into Equation (126) gives Equation (17). □

Corollary C.31 (Loss-certified form)**.** *Under the assumptions of Proposition C.24, term (a) of Equation (17) may be replaced by:*

$$\frac{2R_{\max}}{(1-\gamma)(1-\gamma^m)} \sqrt{\tfrac{1}{2}\, \mathcal{L}_{\mathrm{exc}}(F_Q)}, \tag{132}$$

with $\mathcal{L}_{\mathrm{exc}}$ as in Corollary C.27.

Proof. Insert Equation (118) into Equation (130). □

C.11 Sequential-oracle lower bound

Opaque sequential model. The algorithm starts from a visible root y_0. A query is allowed only at a visible root and returns the transition taken there and the next root label. A label not yet returned cannot be named. This excludes analytic access to $(T_P^{\pi_e})^n$.

Theorem C.32 (Sequential-oracle lower bound)**.** *If an algorithm makes at most h queries and outputs $\widehat{F}$ with $\mathbb{E}\ \mathrm{TV}\big(\widehat{F}, F_P^{\pi_e}(\cdot \mid y_0)\big) \le \varepsilon \in (0, \frac{1}{2})$ on every instance, then:*

$$h \ge \frac{\log\big(1/(2\varepsilon)\big)}{-\log\gamma} = \Omega\Big(\frac{1}{1-\gamma} \log\frac{1}{\varepsilon}\Big). \tag{133}$$

Proof. Step 1 (common transcript). Let P_0, P_1 share a deterministic single-action chain $y_0 \to \cdots \to y_h$ emitting distinct $e_0, \ldots, e_{h-1}$; at y_h, P_i enters an absorbing region emitting only into A_i, with $A_0 \cap A_1 = \emptyset$ and $A_i \cap \{e_j\} = \emptyset$. With at most h queries the responses coincide, so $\mathrm{Law}(\widehat{F})$ is the same under P_0 and P_1.

Step 2 (separation). Both prefixes carry mass $(1-\gamma)\sum_{t<h}\gamma^t = 1-\gamma^h$; the tails carry γ^h on A_0 resp. A_1. Hence:

$$F^{\pi_e}_{P_0}(A_0 \mid y_0) = \gamma^h, \qquad F^{\pi_e}_{P_1}(A_0 \mid y_0) = 0, \tag{134}$$

and, the tails being mutually singular, $\mathrm{TV}\big(F^{\pi_e}_{P_0}(\cdot \mid y_0), F^{\pi_e}_{P_1}(\cdot \mid y_0)\big) = \gamma^h$.

Step 3 (conclusion). For every output, $\gamma^h \le \mathrm{TV}(F^{\pi_e}_{P_0}, \widehat{F}) + \mathrm{TV}(\widehat{F}, F^{\pi_e}_{P_1})$. Taking expectations under the common law of Step 1:

$$\max_{i\in\{0,1\}} \mathbb{E}\ \mathrm{TV}\big(\widehat{F}, F^{\pi_e}_{P_i}\big) \ge \frac{\gamma^h}{2}. \tag{135}$$

Thus $\gamma^h \le 2\varepsilon$, i.e. $h \ge \log(1/2\varepsilon)/(-\log\gamma)$. For $\gamma \ge \frac{1}{2}$, $-\log\gamma \le (1-\gamma)/\gamma \le 2(1-\gamma)$. □

Remark C.33 (Scope). Theorem C.32 concerns one query answered from scratch. It does not imply that BRD avoids horizon-dependent cost: sampling $\beta^\star_m$ via Lemma C.21 pays a comparable cost offline. The relevant trade-off is one-time offline cost against repeated per-query cost (Appendix D.12).

C.12 A Wasserstein variant

Total variation is pessimistic for near-deterministic continuous systems, where laws may be close in transport distance yet mutually singular. Let $\mathcal{X}, \mathcal{E}$ be Polish with metrics $d_{\mathcal{X}}, d_{\mathcal{E}}$, all laws with finite first moments, and assume for constants $L_T, L_G \ge 0$ and all $x, y \in \mathcal{X}$:

$$W_1\big(T^{\pi_e}_P(\cdot \mid x), T^{\pi_e}_P(\cdot \mid y)\big) \le L_T d_{\mathcal{X}}(x,y), \qquad W_1\big(G^{\pi_e}_{P,0}(\cdot \mid x), G^{\pi_e}_{P,0}(\cdot \mid y)\big) \le L_G d_{\mathcal{X}}(x,y),$$
$$\gamma L_T < 1. \tag{136}$$

Proposition C.34 (The resolvent is Lipschitz)**.** *Under Equation (136), for all $x, y \in \mathcal{X}$:*

$$W_1\big(F^{\pi_e}_P(\cdot \mid x), F^{\pi_e}_P(\cdot \mid y)\big) \le L_F d_{\mathcal{X}}(x,y), \qquad L_F \le \frac{(1-\gamma)L_G}{1-\gamma L_T}. \tag{137}$$

Proof. Let $F^{(0)} := G^{\pi_e}_{P,0}$, $F^{(\ell+1)} := \mathcal{B}^{\pi_e}_{P,1} F^{(\ell)}$, with Lipschitz constants L_ℓ. Fix x, y and couple $F^{(\ell+1)}(\cdot \mid x)$ with $F^{(\ell+1)}(\cdot \mid y)$ as follows: a common coin stops with probability $1-\gamma$; on stopping, couple $G^{\pi_e}_{P,0}(\cdot \mid x), G^{\pi_e}_{P,0}(\cdot \mid y)$ optimally (cost $\le L_G d_{\mathcal{X}}(x,y)$); otherwise take a coupling λ of $T^{\pi_e}_P(\cdot \mid x), T^{\pi_e}_P(\cdot \mid y)$ with $\int d_{\mathcal{X}}\ \mathrm{d}\lambda \le L_T d_{\mathcal{X}}(x,y) + \epsilon$ and couple $F^{(\ell)}(\cdot \mid x'), F^{(\ell)}(\cdot \mid y')$ optimally (cost $\le L_\ell d_{\mathcal{X}}(x',y')$). Hence:

$$W_1\big(F^{(\ell+1)}(\cdot \mid x), F^{(\ell+1)}(\cdot \mid y)\big) \le \big((1-\gamma)L_G + \gamma L_T L_\ell\big) d_{\mathcal{X}}(x,y) + \gamma L_\ell \epsilon. \tag{138}$$

With $\epsilon \downarrow 0$, $L_{\ell+1} \le (1-\gamma)L_G + \gamma L_T L_\ell$. As $\gamma L_T < 1$, $\limsup_\ell L_\ell \le L_F$ with $L_F = (1-\gamma)L_G + \gamma L_T L_F$, which is Equation (137); and $F^{(\ell)} \to F^{\pi_e}_P$ by Proposition 3.1. □

Proposition C.35 (Wasserstein dynamics bias)**.** *Assume* $\sup_x W_1\big(G^{\pi_e}_{\widehat{P},0}(\cdot \mid x), G^{\pi_e}_{P,0}(\cdot \mid x)\big) \le \varepsilon_G$, $\sup_x W_1\big(T^{\pi_e}_{\widehat{P}}(\cdot \mid x), T^{\pi_e}_P(\cdot \mid x)\big) \le \varepsilon_T$*, and $F^{\pi_e}_P$ is L_F-Lipschitz. Then:*

$$\sup_x W_1\big(F^{\pi_e}_{\widehat{P}}(\cdot \mid x), F^{\pi_e}_P(\cdot \mid x)\big) \le \varepsilon_G + \frac{\gamma L_F}{1-\gamma}\varepsilon_T, \tag{139}$$

and if r is L_r-Lipschitz on $\mathcal{E}$, then $|J(\pi_e, \widehat{P}) - J(\pi_e, P)| \le L_r(1-\gamma)^{-1}\big(\varepsilon_G + \gamma L_F \varepsilon_T/(1-\gamma)\big)$.

Proof. Let D denote the left side of Equation (139). Couple the one-step fixed-point equations with a common stopping coin as in Proposition C.34. The stopping branch costs $\le (1-\gamma)\varepsilon_G$. On continuation, comparing $F^{\pi_e}_{\widehat{P}}, F^{\pi_e}_P$ at a common next root costs $\le \gamma D$, and moving the next root from $T^{\pi_e}_{\widehat{P}}$ to $T^{\pi_e}_P$ under $F^{\pi_e}_P$ costs $\le \gamma L_F \varepsilon_T$. Hence:

$$D \le (1-\gamma)\varepsilon_G + \gamma D + \gamma L_F \varepsilon_T, \tag{140}$$

which rearranges to Equation (139). The value bound follows from Kantorovich–Rubinstein duality, $|\int r\ \mathrm{d}\mu - \int r\ \mathrm{d}\nu| \le L_r W_1(\mu,\nu)$, and the factor $(1-\gamma)^{-1}$ in Equation (5). □

Remark C.36 (Scope of the Lipschitz assumption). The condition $\gamma L_T < 1$ in Equation (136) requires the root kernel to expand distances by less than a factor $1/\gamma$ per step. It holds for contracting or mildly expanding dynamics but can fail for contact-rich locomotion, where nearby states separate quickly; there the Wasserstein bounds do not apply as stated.

D Empirical Details

This appendix documents every number in Section 4: the environments and artifacts (Appendix D.1), the estimator configuration (Appendix D.2), the reward readout (Appendix D.3), the reference returns and their uncertainty (Appendix D.4), the metrics (Appendix D.5), comparator provenance (Appendices D.6 and D.7), the statistical protocol (Appendix D.8), the full OPE results (Appendix D.9), the block-depth sweep with longer one-step training (Appendix D.10), the training-root ablation (Appendix D.11), and inference cost (Appendix D.12). We write $\widehat{J}$ for the value estimate of a trained generator and $J_{\widehat{P}} := J(\pi_e, \widehat{P})$ for the value of the frozen model, obtained by rolling it out directly with the same frozen reward readout; in the depth and root ablations it is the model-side fidelity target.

Three practices shape what is reported. Every baseline number comes from our own execution of the authors' released code, never from a published constant. The set of environments was fixed before any outcome was seen, and the environments on which BRD loses are reported like the others. The tables are generated from the run artifacts rather than transcribed.

D.1 Environments and Artifacts

Table 4 lists, for each environment, the logged dataset, the target-policy source, the dynamics kernel the generators were trained against, that kernel's digest, and the reward readout. The dynamics network is fitted once per environment from the logged data and frozen; it is not refitted per target policy, training seed, or block depth. Two settings of its wrapper, the box slack and the scale ζ of Table 6, belong to an arm and are held fixed within every comparison that varies the generator, so each such comparison holds the dynamics model fixed.

The three D4RL environments use the `medium-v2` builds with the ten DOPE target policies shipped with the benchmark, evaluated under the `deep_ope_tanh_gaussian` protocol. The two classic-control environments use the target policies of the STITCH-OPE release under its own stochastic TD3 protocol ($\sigma = 1$): six distinct policies on Acrobot, and on Pendulum five configured targets of which only four are distinct, because two of the shipped policy files are byte-identical. That duplication is visible in Table 8 as two equal reference returns. It caps the rank correlation of any estimator that separates the two copies at 0.975, so Pendulum's ρ is not counted as a win in any claim.

In total 283 generators were trained and audited for the main table, one per (environment, target policy, training seed) triple.

D.2 Estimator Configuration

Table 5 gives the configuration shared by all five environments and Table 6 the entries that differ between them. Both are read from the recorded command lines and audit files of the released runs, and every value in Table 5 was checked against a second, independent record of the runs.

Two conventions in Table 6 are easy to misread and are therefore stated explicitly. First, ζ multiplies the learned kernel's standard deviation when targets are drawn; $\zeta = 1$ means no rescaling, and the audit files record the field as absent in that case rather than as zero. Second, the box slack is the tolerance by which a generated state may leave the support box of the logged data before it is clamped; it is a property of the kernel wrapper, not of the diffusion model.

Generator training. Each generator is first fitted to one-step transitions of $\widehat{P}$ for $2{,}000$ steps, which gives F_0, and then trained for 8 sweeps of $2{,}000$ steps, with block depth $m = 64$, root weight $\lambda = 1/2$ (half of the training roots drawn from $d_0^{\pi_e}$ and half from $\widehat{\beta}_m^{\star}$), batch size $2{,}048$, and learning rate 2×10^{-4}, on a residual network of width 512 and depth 6, with 64 diffusion steps and 32 sampling steps, at $\gamma = 0.99$. At every sweep a pool of $65{,}536$ block targets is drawn from the frozen model

Table 4: **Artifacts per environment.** One dynamics kernel is fitted per environment and frozen before any generator is trained; no kernel is refitted per policy or per seed. D4RL rows use the `medium-v2` datasets; classic-control rows use the logged data of the STITCH-OPE release. Kernels are identified by the first 12 hex characters of their SHA-256. "Generators" counts the audited checkpoints behind that row of Table 2 (policies $\times$ training seeds), 283 in total. λ_k is the hyper-parameter of the kernel loss. The reward readout is the frozen rule-selected one: quadratic in the transition features where that form is exact, and a fitted MLP on the two classic-control environments where it is not (Appendix D.3).

Env.	Data	Kernel digest	Kernel loss	Readout	Generators
Hopper	D4RL	`f6036b9ccd6b`	energy, λ_k=1	quad.	150
Walker2d	D4RL	`20149600306f`	MSE	quad.	50
HalfCheetah	D4RL	`5c50f0814bdd`	MSE, λ_k=1	quad.	50
Acrobot	STITCH-OPE	`c9877509fdba`	MSE	MLP	18
Pendulum	STITCH-OPE	`1435054acf23`	MSE	MLP	15
Total					**283**

Table 5: **Generator configuration shared by all five environments**, read from the recorded command lines of the released runs. Per-environment entries are in Table 6. Total optimization is $2000 + 8 \times 2000 = 18{,}000$ steps per generator.

Group	Setting	Value
Bellman target	block depth m	64
	discount γ	0.99
	root weight λ	$1/2$
	target pool size	65536
Diffusion	forward steps	64
	sampling steps	32
Network	residual width	512
	residual depth	6
Optimization	F_0 fitting steps	2000
	sweeps	8
	steps per sweep	2000
	batch size	2048
	learning rate	2×10^{-4}
Audit	samples M	65536
	residual samples	256
	sampling steps	32
	audit seed	123

and the previous generator, and minibatches are drawn from this pool. One generator is trained per (environment, policy, training seed) triple on a single GPU.

Auditing. Each trained generator is audited at $M = 65{,}536$ samples with 256 residual samples and 32 sampling steps, under audit seed 123 and a random stream that is independent of the model. The audit reports $\widehat{J}$ and, on the same stream, $J_{\widehat{P}}$. The latter is the fidelity target throughout the ablations: it does not depend on the generator, so an intervention that changes it has changed the dynamics reference rather than the generator.

D.3 Reward Readout

The generator produces transition features, and a reward readout maps those features to a scalar reward. Which readout is used is decided by a rule fixed before these numbers were produced: fit

Table 6: **Per-environment configuration**, read from the audit files of the released runs. "Seeds" counts independent generator-training replications. ζ multiplies the learned kernel's standard deviation, and $\zeta = 1$ denotes no rescaling. The two classic-control environments use a different model class and a clamped kernel because they do not terminate; the three D4RL environments model an absorbing terminal state. Every cell is audited at the same budget under audit seed 123.

Environment	Seeds	Policies	Protocol	Model class	Terminal	Slack	ζ
Hopper	15	10	DOPE	Bernoulli cemetery	absorbing	0.0	0.001
Walker2d	5	10	DOPE	Bernoulli cemetery	absorbing	0.5	1
HalfCheetah	5	10	DOPE	Bernoulli cemetery	absorbing	0.0	0.001
Acrobot	3	6	TD3, σ=1	cond. trans. score	clamped	0.0	1
Pendulum	3	5	TD3, σ=1	cond. trans. score	clamped	0.0	1

Table 7: **Reference-return protocol.** Read from each environment's frozen reference file. References are Monte Carlo estimates from the simulator, never published constants. "MC s.e." is the mean standard error of the reference across that environment's target policies: on Walker2d and HalfCheetah it is about two units of return, which is the scale some gaps in Table 2 sit at, and the classic-control references are two orders of magnitude sharper because they pool 10^5 episodes per policy. Pendulum is scored at a truncation that does not match its configured horizon, which is why its log-RMSE is descriptive throughout.

Environment	Data / policy build	Rollout H	Scored at	γ	Episodes	MC s.e.
Hopper	`hopper-medium-v2`	768	768	0.99	100	0.540
Walker2d	`walker2d-medium-v2`	768	768	0.99	100	2.230
HalfCheetah	`halfcheetah-medium-v2`	768	768	0.99	100	2.099
Acrobot	`td3_stitch_stochastic`	768	256	0.99	100000	0.057
Pendulum	`td3_stitch_stochastic`	768	256	0.99	100000	0.330

both a quadratic form in the transition features and an MLP on held-out logged transitions, and keep whichever attains the higher coefficient of determination.

The rule selects the quadratic form on all three D4RL environments, where it is essentially exact (R^2 between 0.997 and 0.99999), and the MLP on both classic-control environments, where the quadratic form is not: Acrobot improves from 0.343 to 0.980 and Pendulum from 0.939 to 0.99997. The rows of Table 2 therefore use different readouts, as the rule dictates. The choice matters: scoring the two classic-control cells with the quadratic readout gives a materially different Acrobot number, and the rule, fixed in advance, decides which is reported.

The readout is fitted once per environment on logged data and frozen, like the kernel. It never sees a reference return.

D.4 Reference Returns

Protocol. Reference returns are Monte Carlo estimates from the simulator, not published constants. Table 7 gives the protocol per environment together with the mean Monte Carlo standard error of the reference itself. That column is reported because several gaps in Table 2 are of comparable magnitude: on Walker2d and HalfCheetah the reference is resolved to about 2 units of return, while the classic-control references, pooled over 10^5 episodes per policy, are two orders of magnitude sharper. Pendulum is additionally evaluated at a truncation that does not match its configured horizon, which is why its log-RMSE is reported descriptively throughout.

The reference has its own error bar. A reference return is a Monte Carlo estimate, not a ground truth, and Tables 7 and 8 report its standard error alongside it. On Walker2d and HalfCheetah that error is about two units of return, which is the same scale as some gaps in Table 2; on the two classic-control environments it is two orders of magnitude smaller, because those references pool 10^5 episodes per policy. A reader comparing two methods in a single cell should carry this column: it

Table 8: **Reference return of every target policy**, with its Monte Carlo standard error, in the units each environment is scored in. These are the values both metrics are computed against. Two entries are worth noting: Pendulum's first two targets are identical because the release ships two byte-identical policy files, and Acrobot's first target has zero spread because every episode reaches the same truncated return.

Environment	π_1	π_2	π_3	π_4	π_5	π_6	π_7	π_8	π_9	π_{10}
Hopper	130.2	230.5	236.6	224.6	237.7	245.0	258.7	254.3	256.8	260.7
	±1.18	±0.78	±0.31	±0.71	±0.72	±0.34	±0.32	±0.32	±0.54	±0.18
Walker2d	16.7	73.4	167.7	195.5	125.5	218.9	242.6	247.6	246.7	224.1
	±1.95	±3.48	±2.43	±1.66	±7.47	±1.50	±0.73	±0.84	±0.70	±1.53
HalfCheetah	−29.4	453.3	520.6	565.2	615.1	639.9	656.1	678.8	699.7	728.5
	±1.77	±1.53	±1.67	±1.62	±1.45	±1.56	±2.11	±2.17	±3.07	±4.05
Acrobot	−92.4	−61.3	−58.9	−49.2	−48.1	−46.7				
	±0.00	±0.06	±0.06	±0.06	±0.06	±0.07				
Pendulum	−650.1	−650.1	−286.4	−145.5	−206.9					
	±0.35	±0.35	±0.45	±0.26	±0.23					

bounds how finely any method can be resolved there, independently of how many seeds either side has.

Horizon conventions. D4RL references are rolled to $H = 768$ at $\gamma = 0.99$ and scored at the same horizon. The classic-control references are rolled to 768 and scored at a truncation of 256. Acrobot's configured evaluation horizon is 256 and therefore matches. Pendulum's is 200, which matches neither, so its log-RMSE is reported descriptively in Tables 2 and 14 and no winner is marked in that cell.

Pendulum at its configured horizon. We also scored Pendulum at its configured horizon. A reference at truncation 200 was generated before use, and the released reference file carries truncations 200, 256, and 768. Because each BRD estimate is a single infinite-horizon scalar per policy, changing the truncation changes only the reference vector; nothing is retrained. Re-scoring the same BRD runs and the same seven STITCH-OPE runs against each column gives:

Truncation	BRD	STITCH-OPE	
256 (reported above)	−2.281	−2.087	BRD ahead by 0.194
200 (their estimand)	−1.732	−2.588	**BRD behind by** 0.855

The 256 row is scored against the main-table reference, which pools 10^5 episodes per policy; the 200 row is scored against a separate reference with 10^4 episodes per policy, the only one that carries a 200-step truncation. At truncation 256, where both references exist, they give log-RMSE values that differ by at most 0.004. The sign reverses: at the matched horizon, BRD trails STITCH-OPE by about ten times the latter's run-to-run standard deviation at that horizon (0.081). Spearman's ρ is $+1.000$ at both truncations, so the ranking does not depend on this choice (it is not counted as a win for the separate reason given in Appendix D.1); the level does, and at the horizon STITCH-OPE targets, BRD loses this cell.

We report the 256 column in the tables because both methods are scored at it consistently throughout this paper, and the 200 column here because it matches the comparator's estimand. Neither is free of a residual mismatch: our estimate is infinite-horizon, and at $\gamma = 0.99$ a 200-step truncation discards $0.99^{200} = 13.4\%$ of the discounted mass. Acrobot carries the same residual at $0.99^{256} = 7.6\%$ and is still reported as a comparison, because there the reference matches the comparators' estimand. We treat Pendulum's level as a loss under the matched estimand and count it as neither a win nor a tie in any claim; the tables still bold the best mean at truncation 256. Table 9 records the simulator and software versions behind every reference return.

Table 9: **Simulator and software provenance**, recorded inside the reference file itself. The four digests are the first 12 hex characters of SHA-256 over the simulator manifest, the Gym source, the model XML and the physics engine. They are recorded because a reference return is only reproducible against the simulator that produced it, and because the STITCH-OPE release pins neither its D4RL revision nor its oracle RNG (Appendix D.6). These versions are those of the reference-return computation, which is a CPU simulator rollout; generator training and auditing ran on a single GPU per job under the same Python and PyTorch releases.

Component	Version or digest
Python	`3.10.21`
PyTorch	`2.5.1+cpu`
NumPy	`1.23.5`
Gym	`0.23.1`
mujoco-py	`2.1.2.14`
Simulator manifest	`5c2c3eb2f6cf`
Gym source digest	`3cdf42416ab5`
Model XML digest	`a181c527c392`
Physics engine digest	`9e7ffba152bf`

D.5 Metric Definitions

log-RMSE is computed on the min–max normalization of the *scored set*, matching the convention of the benchmark we compare against. For n evaluated policies with reference returns J_i and estimates $\widehat{J}_i$, write:

$$\tilde{J}_i = \frac{J_i - \min_k J_k}{\max_k J_k - \min_k J_k}, \qquad \tilde{\widehat{J}}_i = \frac{\widehat{J}_i - \min_k J_k}{\max_k J_k - \min_k J_k}, \tag{141}$$

so both are divided by the *reference* range, and an estimate may fall outside $[0, 1]$. Then:

$$\text{log-RMSE} = \log \sqrt{\tfrac{1}{n} \textstyle\sum_i \big(\tilde{\widehat{J}}_i - \tilde{J}_i\big)^2}, \tag{142}$$

$$\rho = \text{Spearman}\big(\widehat{J}_{1:n}, J_{1:n}\big). \tag{143}$$

Two consequences follow and are used when reading the tables. log-RMSE is a logarithm, so a difference of 0.4 is a factor of about 1.5 in RMSE, not a 40% change. Spearman's ρ is computed on the raw estimates, so it is invariant to the normalization and to any monotone recalibration of a method, which is why a change in value calibration can move one metric without the other.

D.6 Comparators and Reproducibility

MBR, STITCH-OPE, fitted Q evaluation, stepwise importance sampling, and weighted importance sampling are external comparators, and every value reported for them comes from an executable implementation rather than a transcribed published constant. MBR is the standard model-based baseline: it fits a one-step dynamics and reward model to the logged data and evaluates the target policy by direct rollout, with no diffusion generator. For STITCH-OPE, fitted Q evaluation, and importance sampling, we run the released implementation with its shipped configuration and compare the resulting executable output with the published value below. Two further families in the STITCH-OPE paper cannot be run from the release as shipped: the `dre` configuration names a class that the release's own baseline module does not contain, and PGD is not in the release. Their doubly robust estimator is implemented but configured nowhere; we run it with the hyper-parameters their paper reports (Table 11). Run counts per family and environment are listed beneath Table 14 and are uneven, since several traditional estimators are deterministic given their configuration and were run once; a missing standard deviation means a single run, not zero variance.

Published against reproduced. Table 10 audits the external comparators we can run against the values the STITCH-OPE paper prints for them. For each run it also gives the score that *the release's own code* computes against its own reference returns, beside our score of the same run. Off Pendulum the two agree to within 0.1 in every cell; on Pendulum the release scores at the configured

Table 10: **Published against reproduced external comparators.** Pub.: the value printed in the STITCH-OPE paper. Own: the value *the release's* code computes for our run of it, against *its* own reference returns. Ours: the same run scored against our reference returns, as in Table 2. Own and Ours agree almost everywhere, so wherever the paper and our table differ, the released code does not reproduce the published value; the gap is not introduced by our scoring. *Released code differs from the published value by more than 0.3. The published IS column is matched against both our stepwise and weighted variants because it agrees with the weighted one on some environments and the stepwise one on others.

		log-RMSE			Spearman ρ		
Environment	Estimator	Pub.	Own	Ours	Pub.	Own	Ours
Hopper	FQE	-0.42^{*}	-1.019	-1.018	$+0.17$	$+0.302$	$+0.321$
	IS, stepwise	-0.48^{*}	$+0.595$	$+0.594$	-0.06^{*}	$+0.273$	$+0.321$
	IS, weighted	-0.48^{*}	-1.182	-1.191	-0.06^{*}	-0.485	-0.497
	STITCH-OPE	-2.33	-2.256	-2.278	$+0.76$	$+0.775$	$+0.812$
Walker2d	FQE	-0.48^{*}	-0.107	-0.091	$+0.41$	$+0.297$	$+0.273$
	IS, stepwise	-0.71^{*}	-0.213	-0.187	$+0.51^{*}$	-0.721	-0.733
	IS, weighted	-0.71^{*}	-1.029	-1.033	$+0.51$	$+0.406$	$+0.455$
	STITCH-OPE	-1.33	-1.255	-1.255	$+0.63$	$+0.499$	$+0.504$
HalfCheetah	FQE	-0.05	-0.246	-0.240	-0.03^{*}	-0.491	-0.491
	IS, stepwise	-0.84^{*}	-0.252	-0.248	$+0.57^{*}$	-0.042	-0.042
	IS, weighted	-0.84	-0.747	-0.744	$+0.57^{*}$	$+0.879$	$+0.879$
	STITCH-OPE	-0.85^{*}	-0.446	-0.441	$+0.87^{*}$	$+0.376$	$+0.376$
Acrobot	FQE	-0.14	$+0.097$	$+0.093$	$+0.75$	$+0.943$	$+0.943$
	IS, stepwise	-1.00^{*}	-0.229	-0.168	$+0.52^{*}$	-0.623	-0.543
	IS, weighted	-1.00	-1.049	-0.976	$+0.52$	$+0.509$	$+0.543$
	STITCH-OPE	-2.02^{*}	-1.430	-1.509	$+0.82$	$+0.801$	$+0.751$
Pendulum	FQE	-0.58^{*}	-2.300	-2.045	$+0.89$	$+0.900$	$+0.975$
	IS, stepwise	-0.15	-0.137	-0.180	-0.60	-0.640	-0.667
	IS, weighted	-0.15	-0.193	-0.232	-0.60^{*}	-0.280	-0.154
	STITCH-OPE	-2.34	-2.504	-2.087	$+0.96$	$+0.954$	$+0.975$

Table 11: **Doubly robust, the release's implementation** with the hyper-parameters of the STITCH-OPE paper. "Published" is the value that paper prints. †Not horizon-matched (Appendix D.4).

	log-RMSE			Spearman ρ		
Environment	Published	DR	BRD	Published	DR	BRD
Hopper	-0.57	–	-2.139	0.69	–	0.751
Walker2d	-1.25	-0.656	$\mathbf{-1.710}$	0.50	0.600	**0.697**
HalfCheetah	0.01	-0.119	$\mathbf{-0.854}$	-0.48	-0.648	**0.886**
Acrobot	-0.49	$+0.191$	$\mathbf{-2.085}$	0.63	-0.657	**0.962**
Pendulum†	-1.02	-1.370	$\mathbf{-2.281}$	0.72	0.872	**1.000**

horizon of 200 and we at 256 (Appendix D.4). The differences between the paper and Table 2 are therefore differences between the paper and its release, not artifacts of our scoring. They run in both directions: released STITCH-OPE on HalfCheetah and Acrobot is weaker than published, while fitted Q evaluation on Hopper and Pendulum is stronger. The published IS column is matched by our weighted variant on Acrobot and by the stepwise one on Pendulum; on the other three environments neither variant matches it on both metrics. We report the reproduced values throughout, including where the published ones would favor us.

Doubly robust. We run the release's `DoublyRobustEstimator` unchanged, registering it in the release's own entry point and using the hyper-parameters of the STITCH-OPE paper's appendix, on the same data and targets as the IS column (Table 11). BRD is more accurate than DR on both metrics on Walker2d, HalfCheetah, and Acrobot. Hopper is missing: at the paper's batch size of 32 the release's implementation needs about 20 CPU-hours per target policy, and a faster re-implementation did not pass the equivalence test we fixed in advance, so we report none.

Table 12: **Same model, different Bellman targets, with standard deviations.** Table 1 with the sample standard deviation over training seeds. Within each environment the $m = 1$ and $m = 16$ columns share one frozen dynamics kernel and reward readout, so their difference is caused by how the generator is trained. $m = 1$ is the one-step TD objective of γ-models and TD-flows run on that kernel; $m = 16$ is BRD. TD^2-flow is the released implementation of the same one-step objective, which bootstraps on the logged transitions and uses no dynamics model (same data and targets, training seed 0). †Comparable within the table (Table 1).

	Environment	m=1	TD^2-flow	m=16
Log-RMSE↓	Hopper	-0.06 ±0.07	-0.22	$\mathbf{-1.30}$ ±0.12
	Walker2d	-0.20 ±0.03	-0.59	$\mathbf{-1.86}$ ±0.06
	HalfCheetah	-0.37 ±0.01	-0.57	$\mathbf{-0.85}$ ±0.01
	Acrobot	-0.71 ±0.13	$+1.61$	$\mathbf{-0.93}$ ±0.11
	Pendulum†	-0.38 ±0.03	-0.21	$\mathbf{-1.57}$ ±0.08
Spearmanρ↑	Hopper	$+0.03$ ±0.15	$+0.32$	$\mathbf{+0.53}$ ±0.08
	Walker2d	-0.14 ±0.19	$+0.04$	$\mathbf{+0.72}$ ±0.09
	HalfCheetah	$+0.58$ ±0.18	$+0.27$	$\mathbf{+0.84}$ ±0.08
	Acrobot	-0.05 ±0.09	$\mathbf{+0.71}$	$+0.69$ ±0.08
	Pendulum†	$+0.60$ ±0.05	$+0.21$	$\mathbf{+1.00}$ ±0.00

D.7 The Released One-Step Occupancy Baseline

TD-flows (Farebrother et al., 2025) introduced the one-step flow-matching objective for occupancy models, but its authors released no code. InFOM (Zheng et al., 2026) ships the objective as `flow_occupancy_loss`, in its model-free form that bootstraps on logged transitions, and we run that implementation (commit `2eb4625`, MIT license) with its default hyper-parameters and its 10^6-step budget, and refer to it as TD^2-flow.

Three changes are forced by off-policy evaluation, and there are no others. First, the bootstrap action is drawn from the target policy, $a' \sim \pi(s')$, rather than taken from the log, since the logged next action would make the model learn the behavior policy's occupancy. Second, the reward head is trained alongside the flow, because the release trains it only during online fine-tuning. Third, true terminations are routed to an absorbing state (Kostrikov et al., 2019). The release has no terminal handling, and an arm without this change is worse in level on both environments that have terminations.

The value readout is the release's own critic target:

$$\widehat{V} = \frac{\mathbb{E}[\hat{r}(x)]}{1-\gamma},$$

where the expectation is estimated from 64 flow samples for each of 1024 roots, that is, 65,536 samples, as in our audits. We evaluate the same 41 targets against the same reference returns as Table 2, at training seed 0. All three pre-registered predictions held, with BRD taken at $m = 16$, the depth run in all five environments: BRD has a lower log-RMSE in $5/5$ environments and a higher rank correlation in $4/5$, and TD^2-flow's error is closer to our one-step implementation than to BRD in $5/5$. The exception to the ranking is Acrobot, where TD^2-flow reaches $\rho = 0.71$ against 0.66 for BRD at $m = 16$ and seed 0; its log-RMSE there, $+1.61$, is the worst in Table 1. The same counts hold when BRD is averaged over training seeds, as in Table 1.

D.8 Statistical Protocol

Bold in Tables 2 and 14 marks the best unrounded mean and is not a significance claim. Where we do test, two axes are used, and they answer different questions.

Axis A: variation across runs. An exact one-sided permutation test on the pooled runs of BRD and STITCH-OPE, the only comparator with repeated runs in every environment, enumerating all $\binom{n_1+n_2}{n_1}$ arrangements rather than sampling them. The statistic is the difference of means; the reported

Table 13: **Axis A: exact one-sided permutation over runs against STITCH-OPE**, scored once at the final run counts. "floor" is the smallest attainable p, $1/\binom{n_1+n_2}{n_1}$; a p at the floor means complete separation with no resolution left, which is why it is quoted. Pendulum is omitted: its level is not horizon-matched (Appendix D.4) and its ranking saturates on four distinct policies.

Environment	Metric	BRD	STITCH-OPE	gap	p	
Hopper	log-RMSE	-2.1392	-2.2784	-0.1392	0.9692	fail
	ρ	+0.7511	+0.8121	-0.0610	0.9788	fail
Walker2d	log-RMSE	-1.7099	-1.2552	+0.4547	0.0003	**PASS** (at floor)
	ρ	+0.6970	+0.5042	+0.1927	0.0033	**PASS**
HalfCheetah	log-RMSE	-0.8542	-0.4414	+0.4128	0.0008	**PASS** (at floor)
	ρ	+0.8861	+0.3758	+0.5103	0.0008	**PASS** (at floor)
Acrobot	log-RMSE	-2.0849	-1.5092	+0.5756	0.0485	**PASS**

p is the fraction of arrangements at least as extreme as the observed one, and the attainable floor $1/\binom{n_1+n_2}{n_1}$ is quoted with it, because a p at the floor means complete separation with no resolution left, not strong evidence. This axis asks whether the gap survives retraining and re-running.

Axis B: variation across target policies. A paired bootstrap that resamples the *target policies* rather than the runs, recomputing both methods' metrics on each resample and reporting the lower confidence bound of the paired difference. This axis asks whether the gap is a property of the environment or of a few policies within it; a cell can pass axis A and fail axis B when most of its gain is concentrated in two or three targets.

Axis B is the stricter test. We report the outcomes of both axes, in the main text as well as here.

Outcomes. Axis A was scored once at the final run counts (Table 13). Walker2d and HalfCheetah separate completely on both metrics: their p sits at the attainable floor, and Walker2d's rank correlation is additionally *resolved* rather than at the floor. Acrobot clears 0.05 on log-RMSE at $p = 0.0485$, a margin of 0.0015 that should be read as such. Hopper fails both metrics in the direction we would need. On HalfCheetah, WIS comes within 0.007 of BRD in ρ; it is deterministic, was run once, and admits no run-level test.

We state the axis-B outcomes in words, at the final run counts. Acrobot fails axis B with a lower bound of -0.113 and is more accurate on three of its six targets. HalfCheetah passes axis B, while Walker2d and Hopper fail it, Walker2d because most of its gain sits in three of ten policies. **HalfCheetah is therefore the only cell that passes both axes against STITCH-OPE**, and no claim in this paper rests on axis B for any other cell.

Seeds. Four seed roles are recorded separately in every artifact: the training seed, which indexes independent generator-training replications on the same frozen kernel, and the audit, bootstrap, and permutation seeds. Table 6 gives the number of training seeds per environment. Hopper carries fifteen. The mean over its first six is -2.271 and over all fifteen -2.139; the subset would read as a tie with STITCH-OPE and the full set reads as a loss. We report the full set.

D.9 Complete OPE Results

Table 14 is Table 2 with sample standard deviations and run counts. Bold marks the best unrounded mean in a row and is not a significance claim. Standard deviations are across the runs counted beneath the table; a missing one means a single run.

D.10 Block Depth: Complete Sweep

Figure 2 plots these numbers normalized by each curve's own mean at $m = 1$, which is what makes five environments of different natural scale comparable on one axis. Table 15 gives them unnormalized, with the seed standard deviation and the pairwise resolution behind every ordering claim in Section 4.

Table 14: **Main OPE results on five control environments.** MBR is direct rollout of a learned one-step model. Entries are means with sample standard deviations across available runs; no SD is shown for single-run entries. Bold denotes the best unrounded mean within each row, not statistical significance. †Pendulum is not horizon-matched, so its log-RMSE is descriptive (Appendix D.4).

Environment	IS	WIS	FQE	MBR	STITCH-OPE	BRD
Log-RMSE ↓						
Hopper	$0.594_{\pm 0.000}$	$-1.191_{\pm 0.011}$	-1.018	0.099	$\mathbf{-2.278}_{\pm 0.022}$	$-2.139_{\pm 0.228}$
Walker2d	-0.187	-1.033	$-0.091_{\pm 0.000}$	-0.829	$-1.255_{\pm 0.029}$	$\mathbf{-1.710}_{\pm 0.051}$
HalfCheetah	-0.248	-0.744	-0.240	-0.369	$-0.441_{\pm 0.013}$	$\mathbf{-0.854}_{\pm 0.012}$
Acrobot	-0.168	-0.976	0.093	-1.416	$-1.509_{\pm 0.561}$	$\mathbf{-2.085}_{\pm 0.090}$
Pendulum†	-0.180	-0.232	-2.045	-0.482	$-2.087_{\pm 0.043}$	$\mathbf{-2.281}_{\pm 0.088}$
Spearman ρ ↑						
Hopper	$0.321_{\pm 0.000}$	$-0.497_{\pm 0.026}$	0.321	-0.176	$\mathbf{0.812}_{\pm 0.006}$	$0.751_{\pm 0.111}$
Walker2d	-0.733	0.455	$0.273_{\pm 0.000}$	0.503	$0.504_{\pm 0.109}$	$\mathbf{0.697}_{\pm 0.101}$
HalfCheetah	-0.042	0.879	-0.491	0.527	$0.376_{\pm 0.060}$	$\mathbf{0.886}_{\pm 0.043}$
Acrobot	-0.543	0.543	0.943	0.943	$0.751_{\pm 0.048}$	$\mathbf{0.962}_{\pm 0.033}$
Pendulum†	-0.667	-0.154	0.975	0.872	$0.975_{\pm 0.000}$	$\mathbf{1.000}_{\pm 0.000}$

Run counts (IS/WIS/FQE/STITCH-OPE/BRD): Hopper 2/2/1/10/15; Walker2d 1/1/2/10/5; HalfCheetah 1/1/1/8/5; Acrobot 1/1/1/14/3; Pendulum 1/1/1/7/3. Pendulum has five configured targets but only four distinct policy files.

A gap between two depths is called *resolved* when it exceeds twice the standard error of the paired difference across the five training seeds. Of the 90 ordered pairs (5 environments $\times$ 3 metrics $\times$ 6 depth pairs), 88 are resolved. The two that are not are Pendulum's value error between $m = 64$ and $m = 16$, and Acrobot's between $m = 1$ and $m = 4$; the latter is the inversion that makes Acrobot the one environment whose ordering is not monotone in every metric.

Depth and sweep budget. All arms use the same optimization budget: 8 sweeps of 2,000 steps. Deeper blocks therefore also resolve a longer horizon mQ before any fitting error enters, and the a priori bound equation 15 predicts exactly this joint effect: the initial error decays as γ^{mQ}, which is 0.92 at $m = 1$ and 0.006 at $m = 64$, and fitting error is amplified by at most $1/(1-\gamma^m)$, which is 100 at $m = 1$ and 2.1 at $m = 64$. The sweep compares depths at matched optimization cost, not at matched frozen-model calls, which grow with m as Theorem 3.2 requires. The $m = 1$ arm is under-iterated at this budget; it is run longer below.

Depth and OPE accuracy. Table 15 measures fidelity to the learned kernel, whereas Table 16 scores the same generators against reference returns on the family where block depth is the only knob. The two objectives need not be monotone together. Fidelity improves through $m = 64$, while on Walker2d OPE accuracy peaks at $m = 16$, which beats $m = 64$ by 0.148 in log-RMSE at a seed standard deviation of about 0.05. At this budget $m = 1$ is worse than the finite-depth blocks on every reported environment. This distinction is expected: depth controls how accurately the generator represents the frozen model's discounted future, but it does not remove error in the learned dynamics model itself.

One-step bootstrapping run longer. On Walker2d we re-ran the $m = 1$ arm with $Q \in \{32, 128, 400\}$ sweeps, keeping the target pool and the steps per sweep unchanged (training seed 0; Table 17). At $Q = 400$ the a priori bound leaves $\gamma^{mQ} = 0.018$ of the initial error, so the arm has essentially converged, with fifty times the sweeps of BRD. Its OPE level improves monotonically, from -0.18 at $Q = 8$ to -1.42, -1.50, and -1.68, and at $Q = 400$ its rank correlation reaches 0.59. Its fidelity to the kernel nevertheless does not converge. From $Q = 32$ on, its sliced Wasserstein distance stays between 0.61 and 0.69 and its value error between 0.10 and 0.21 of the reference span, against 0.083 and 0.018 for $m = 64$ at eight sweeps. The pre-registered test of the $1/(1-\gamma^m)$ amplification, which asks whether converged $m = 1$ remains less faithful than $m = 64$ on both metrics, is therefore supported. One-step bootstrapping thus approaches the OPE accuracy of the recursion only at fifty times its training budget, and never its fidelity to the frozen model. The released

Table 15: **Block-depth sweep, complete.** Mean ± sample standard deviation over five training seeds, against the same frozen kernel; lower is better. "Res." counts how many of the six ordered depth pairs are resolved, a gap counting as resolved when it exceeds twice the standard error of the paired difference across seeds (88 of 90 overall). These are distances to a *learned* kernel, not to the environment, so a decrease here is not by itself a bound on OPE error. Acrobot's value error is the one non-monotone cell.

Environment	Metric	$m = 1$	$m = 4$	$m = 16$	$m = 64$	Res.
Hopper	Sliced Wasserstein	$0.678_{\pm 0.0040}$	$0.426_{\pm 0.0073}$	$0.169_{\pm 0.0071}$	$0.0546_{\pm 0.0014}$	6/6
	Value error	$77.1_{\pm 9.0}$	$33.3_{\pm 4.1}$	$25.7_{\pm 3.8}$	$6.33_{\pm 1.6}$	6/6
	Feature RMSE	$0.744_{\pm 0.0031}$	$0.386_{\pm 0.0073}$	$0.137_{\pm 0.011}$	$0.0475_{\pm 0.0028}$	6/6
Walker2d	Sliced Wasserstein	$0.968_{\pm 0.022}$	$0.467_{\pm 0.011}$	$0.144_{\pm 0.0059}$	$0.0781_{\pm 0.0069}$	6/6
	Value error	$153_{\pm 4.3}$	$28.4_{\pm 8.0}$	$18.5_{\pm 5.9}$	$6.71_{\pm 2.2}$	6/6
	Feature RMSE	$1.13_{\pm 0.034}$	$0.507_{\pm 0.018}$	$0.121_{\pm 0.0083}$	$0.0713_{\pm 0.012}$	6/6
HalfCheetah	Sliced Wasserstein	$0.686_{\pm 0.031}$	$0.301_{\pm 0.022}$	$0.156_{\pm 0.020}$	$0.0855_{\pm 0.0061}$	6/6
	Value error	$195_{\pm 5.1}$	$52.8_{\pm 6.3}$	$8.00_{\pm 1.2}$	$5.87_{\pm 1.9}$	6/6
	Feature RMSE	$0.669_{\pm 0.029}$	$0.234_{\pm 0.027}$	$0.131_{\pm 0.029}$	$0.0721_{\pm 0.0075}$	6/6
Acrobot	Sliced Wasserstein	$0.510_{\pm 0.015}$	$0.369_{\pm 0.0078}$	$0.155_{\pm 0.019}$	$0.0557_{\pm 0.0064}$	6/6
	Value error[‡]	$25.5_{\pm 2.9}$	$25.8_{\pm 2.4}$	$15.5_{\pm 4.1}$	$5.03_{\pm 2.4}$	5/6
	Feature RMSE	$0.387_{\pm 0.0036}$	$0.305_{\pm 0.0090}$	$0.104_{\pm 0.021}$	$0.0449_{\pm 0.0077}$	6/6
Pendulum	Sliced Wasserstein	$0.507_{\pm 0.0048}$	$0.163_{\pm 0.0093}$	$0.0807_{\pm 0.010}$	$0.0385_{\pm 0.0054}$	6/6
	Value error	$306_{\pm 5.5}$	$74.5_{\pm 6.9}$	$13.3_{\pm 3.9}$	$10.9_{\pm 3.7}$	5/6
	Feature RMSE	$0.477_{\pm 0.0060}$	$0.153_{\pm 0.017}$	$0.0688_{\pm 0.017}$	$0.0313_{\pm 0.0072}$	6/6

[‡]The mean is not monotone in m: Acrobot's value error $|\widehat{J} - J_{\widehat{P}}|$ rises from 25.46 at $m = 1$ to 25.83 at $m = 4$ before falling to 5.03 at $m = 64$. That m=1 versus m=4 gap is also one of the two unresolved pairs; the other is Pendulum's m=64 versus m=16.

Table 16: **OPE accuracy against block depth**, five training seeds, scored against reference returns. Walker2d varies only the block depth of the Table 2 arm, whose m=64 row is that arm. Hopper and HalfCheetah report the m=1 and m=16 arms of their depth sweeps, which use box slack 0.5 and therefore differ from the Table 2 arms.

Environment	m	log-RMSE	ρ
Walker2d	1	$-0.204_{\pm 0.028}$	$-0.144_{\pm 0.194}$
	4	$-1.208_{\pm 0.121}$	$+0.362_{\pm 0.096}$
	16	$\mathbf{-1.858}_{\pm 0.055}$	$+0.719_{\pm 0.088}$
	64	$-1.710_{\pm 0.051}$	$+0.697_{\pm 0.101}$
Hopper	1	$-0.063_{\pm 0.070}$	$+0.028_{\pm 0.152}$
	16	$-1.303_{\pm 0.122}$	$+0.530_{\pm 0.077}$
HalfCheetah	1	$-0.368_{\pm 0.010}$	$+0.578_{\pm 0.181}$
	16	$-0.855_{\pm 0.013}$	$+0.842_{\pm 0.079}$

model-free implementation, trained for its default 10^6 steps, also remains closer in OPE error to our one-step arm than to BRD (Appendix D.7).

D.11 Training-Root Ablation

Setup. We examine whether training only on initial state-action pairs is sufficient. The implementation draws half its training roots from the initial law $d_0^{\pi_e}$ and half from $\widehat{\beta}_m^{\star}$ ($\lambda = 1/2$). We compare this mixture with initial roots alone ($\lambda = 1$), holding the dynamics model, block depth ($m = 64$), and number of sweeps (8) fixed. This mechanism experiment uses the Walker2d arm of Table 2 at training seed 0, audit seed 123, and all ten policies. The comparison matches the training schedule, not total computation: initial-only roots avoid the rollout cost of constructing continuation roots.

Table 17: **One-step bootstrapping run longer**, Walker2d, training seed 0, ten policies, one frozen kernel for every row. SW is the sliced Wasserstein distance to the frozen model's occupancy and $|\widehat{J} - J_{\widehat{P}}|$ the value error normalized by the reference span. BRD $m = 64$ is training seed 0 of the main-table Walker2d arm (Tables 2 and 16; five-seed mean -1.710), trained on the same frozen kernel as every $m = 1$ row.

Arm	Sweeps Q	SW	$\lvert\widehat{J} - J_{\widehat{P}}\rvert$	log-RMSE	ρ
$m = 1$	8	0.972	0.679	-0.179	-0.236
$m = 1$	32	0.609	0.100	-1.415	$+0.285$
$m = 1$	128	0.688	0.209	-1.501	$+0.297$
$m = 1$	400	0.677	0.164	-1.679	$+0.588$
BRD $m = 64$	8	**0.083**	**0.018**	-1.741	$+\mathbf{0.685}$
BRD $m = 16$	8	0.144	0.096	$\mathbf{-1.943}$	$+0.636$

Table 18: **Training-root ablation**, Walker2d, training seed 0, ten policies, same frozen kernel in both arms. SW is the mean sliced Wasserstein distance to the frozen model's occupancy.

Root distribution	SW ↓	Log-RMSE ↓	ρ ↑
50/50 mixture	0.0828	-1.741	0.685
Initial only	0.2010	-1.125	0.758

Results and interpretation. Removing continuation roots increases the sliced Wasserstein distance to the frozen model on all ten policies, by 142.8% on average (Table 18). The frozen dynamics kernel is identical between the two arms, so the change is caused by the generator-side intervention. This is the failure Proposition C.22 predicts: with $\lambda = 1$ the bootstrap queries at m-step successors fall outside the training distribution. The ablation does not establish that $\lambda = 1/2$ is optimal, and it does not test $\lambda = 0$; the audit distribution $\beta_m^\star$ is optimal for certification, which is a different question from which training distribution is best at a finite budget. The OPE consequences are mixed: initial-only roots worsen log-RMSE while improving ranking (Table 18), so fidelity is not used as a proxy for OPE.

D.12 Inference Cost

What is measured. BRD draws each evaluation sample with a fixed number of reverse-diffusion steps, 32 here, independent of the horizon. A rollout that samples one discounted transition by geometric stopping needs $1/(1-\gamma) = 100$ dynamics calls in expectation at $\gamma = 0.99$. The profiled denoiser has 2.606 times the forward-pass FLOPs of the dynamics model, so a BRD sample costs about 83.4 dynamics-call equivalents; measured forward-pass time ratios are 2.043 at batch size 2048 and 1.705 at batch size 1.

What these numbers do not show. The measurements are per sample and per component. They exclude the cost of training the generator, which comprises the rollouts that build block targets and root samplers and $18{,}000$ optimization steps, and which exceeds the cost of a single rollout-based evaluation. They also compare against geometric stopping: a rollout that follows each trajectory over the full horizon and sums discounted rewards reads a complete return from every chain and typically needs fewer dynamics calls for the same Monte Carlo accuracy. We therefore make no claim of end-to-end speedup. The generator provides a fixed sequential depth per sample for an occupancy that must be queried often enough to amortize its training.