CS-DICE: Offline Reinforcement Learning with Coherent Occupancy Regularization

Jeremías Figueiredo Paschmann
Escuela de Ingeniería
Universidad de San Andrés

Abstract

Offline RL methods often regularize toward the empirical occupancy of the dataset, but this reference can mismatch the deployed policy. In partially observed or decentralized settings, data may be pooled from hidden modes, histories, or conventions that are unavailable at execution time. The pooled occupancy can then require incompatible action laws for the same deployed input, or joint correlations that a decentralized policy class cannot represent, so regularization toward it can favor a low-return projection. We formalize this failure through *reference coherence*: a reference is coherent when it does not rely on information hidden from the deployed policy, and when its action law is representable by that policy. We propose *Closest Slice DICE (CS-DICE)*, a DICE variant that regularizes toward a candidate slice of the training data rather than the pooled occupancy. A hard or soft selector chooses the regularization reference, while the deployed observations, actor class, and factorization constraints remain unchanged. Most of the results focus on the soft reverse-KL case, where the selector only changes the reference used by an otherwise standard DICE update. Controlled examples show that incoherent pooled references collapse to suboptimal behavior, while CS-DICE recovers high-return policies when the selected or inferred slice is coherent with the deployed interface. We also identify a boundary case where pooling is benign because the relevant signal is revealed before the conflict. Finally, a CS-CoMA-DICE study on MaMuJoCo shows that selected references can be integrated into a neural cooperative MARL pipeline while preserving decentralized execution, with competitive observed peaks but mixed final performance.

1 Introduction

Offline reinforcement learning (RL) learns policies from fixed datasets, avoiding the cost and risk of online interaction [Fu et al., 2020, Kumar et al., 2020]. Since the learned policy cannot collect corrective data, many offline methods bias learning toward the logged behavior in order to avoid out-of-distribution or poorly supported actions. DualDICE estimates occupancy ratios for off-policy evaluation [Nachum et al., 2019a]. DICE policy-optimization methods such as AlgaeDICE and OptiDICE use the dataset occupancy as a regularization reference [Nachum et al., 2019b, Lee et al., 2021]. This choice is natural when the pooled dataset describes behavior that the deployed policy can actually represent. However, we show it can be misleading when the behavior in the dataset was generated using distinctions that the deployed policy will not observe.

Consider a two-step partially observable Markov decision process with action set $A = \{0, 1\}$. From a fixed initial state, the first action $i \in A$ deterministically selects state x_i. At the second decision, both x_0 and x_1 produce observation $\bar{o}$, and the terminal reward is $r(x_i, a) = \mathbf{1}\{a = i\}$. Thus $0{\to}0$ and $1{\to}1$ are both successful demonstrations with return 1. The deployed policy is reactive, $\pi(a \mid o)$, so it must use the same action distribution at both hidden states. If the dataset mixes the

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

two demonstrations equally, collapsing the states to their shared observation gives a pooled action marginal that assigns probability $1/2$ to each action. A reactive policy can match this marginal, but doing so loses the dependence between the branch and the second action, yielding expected return $1/2$. With full-support smoothing, strong regularization toward the pooled reference can therefore favor this compromise over commitment to either successful trajectory.

In this work, we study how to choose the regularization reference when training data can be partitioned into slices, but deployment must still use a fixed reactive or decentralized policy interface. Closest Slice DICE (CS-DICE) changes the training reference rather than the deployed policy. Instead of regularizing toward the pooled reference, it uses slices available during training to choose a reference better aligned with the behavior the deployed policy can express. The learned policy uses the original deployment interface. This separates two questions that are usually entangled: whether the data contain good behavior, and whether the pooled occupancy is a coherent reference for deployment. Applying the same selected reference idea to CoMA-DICE yields a cooperative multi-agent reinforcement learning variant with decentralized actors conditioned on local observations. We call this instantiation **CS-CoMA-DICE**.

Our contributions are:

- **Reference coherence.** We formulate reference coherence as compatibility between a regularization reference, the information available at deployment, and the deployed policy class. This identifies information and factorization constraints that pooled action laws can violate.
- **Reference selection for DICE.** We introduce CS-DICE, which selects a training reference from dataset slices while retaining the deployed actor. Hard selection uses one slice; the soft reverse-KL instantiation uses a logarithmic pool of slice occupancies in a standard DICE update.
- **Finite counterexamples and recovery conditions.** We give a two-step partially observed construction in which pooled reverse-KL regularization selects a unique policy with return $1/2$, while coherent endpoints attain 1. We characterize exact-support recovery, derive a finite-smoothing separation condition for the cooperative witness, and give a reveal-before-action control in which pooling succeeds.
- **Controlled tests and neural implementation.** Single-agent and cooperative experiments test the predicted failure, recovery, and reveal boundary, and show how slice representations affect return. Across twelve cells in a divergence-matched MaMuJoCo study, CS-CoMA-DICE attains mean selected/pooled ratios of 0.98 at observed best evaluations and 0.89 at fixed final checkpoints.

2 Related Work

Offline reinforcement learning. Offline RL learns policies from datasets collected beforehand without further environment interaction [Fu et al., 2020, Kumar et al., 2020]. A central challenge is distribution shift: learned policies may visit actions or states that are poorly supported by the dataset. Conservative value learning, actor rectification, and policy improvement restricted to the data address this issue through value penalties, action constraints, or updates restricted to well supported behavior [Kumar et al., 2020, Pan et al., 2022, Liu et al., 2024]. DualDICE estimates discounted occupancy ratios for off-policy evaluation [Nachum et al., 2019a]. AlgaeDICE and OptiDICE use occupancy regularization relative to the dataset distribution for policy optimization [Nachum et al., 2019b, Lee et al., 2021].

Offline cooperative MARL. Offline MARL combines offline distribution shift with the decentralized execution constraints and large joint action spaces of multi-agent systems, as formalized by Dec-POMDPs [Oliehoek and Amato, 2016, Chapter 2]. Cooperative methods commonly use centralized training with decentralized execution, so training can use joint or state information while deployed actors condition only on their local observations. CoMA-DICE and AlberDICE carry occupancy regularization into this cooperative setting [Bui et al., 2025, Matsunaga et al., 2023], while other offline MARL methods adapt conservative value learning, actor regularization, or policy improvement restricted to the data to decentralized actors [Pan et al., 2022, Liu et al., 2024]. Other-play and label-free coordination study compatibility between independently trained agents [Hu et al., 2020,

Treutlein et al., 2021]. Off-Team Learning addresses belief and policy mismatch in cross-play [Cui et al., 2022], while CoMeDi learns diverse conventions for collaboration with new partners [Sarkar et al., 2023].

Mixtures and mode structure. Heterogeneous offline data may contain multiple partners, skills, latent modes, or conventions. OG-MARL and subsequent dataset analysis document how dataset composition affects offline MARL performance [Formanek et al., 2023, 2024]. Tilbury et al. propose joint-action-similarity sampling to mitigate coordination failure [Tilbury et al., 2024]. Other methods model subgroup skills [Chen et al., 2024], use latent or mixture policies [Chen et al., 2022, Osa et al., 2023], cluster trajectories by their generating policies [Hu et al., 2025], or select a promising action mode [Wang et al., 2025]. Scaled offline Q-learning studies representation learning from diverse multi-task data [Kumar et al., 2023], and MACKRL uses common knowledge to coordinate decentralized policies [Schroeder de Witt et al., 2018]. CS-DICE instead holds the deployed information and actor class fixed and changes the reference inside a DICE objective. Clustering or learned representations can supply its training slices, while richer actors address deployments that must express several conventions. We study reference choice within DICE.

3 Setup and Reference Coherence

The behavior that generated an offline dataset may have used more information than the actor that will be deployed. That information may be a state, history, centralized joint record, convention, partner identity, inferred mode, or latent branch, and it need not be recorded in the dataset. We represent such distinctions used during generation by a refinement $x \in X$ and a collapse map $c : X \to O$, where $o = c(x)$ is the input available to the deployed actor. The deployed actor chooses actions in A and must act through the deployed policy class Π_{dep}. A reference is coherent with respect to that refinement when its prescribed action law can be written using only $c(x)$ and is representable by the deployed policy class; equivalently, it cannot ask the actor to condition on distinctions erased by c.

When a reference is specified on a refinement, let d^{ref} be a normalized reference occupancy on $X \times A$, and let $p^{\mathrm{ref}}(\cdot \mid x)$ be the action distribution it prescribes at refined record x. We call a reference occupancy d^{ref} coherent with respect to (c, Π_{dep}) when its action law is expressible by a deployed policy:

$$p^{\mathrm{ref}}(\cdot \mid x) = \pi(\cdot \mid c(x)) \quad \text{for reference-a.e. } x, \qquad \pi \in \Pi_{\mathrm{dep}}.$$

This applies action-law compatibility under partial information [Warrington et al., 2021, Definition 2] to a fixed regularization reference and the information and factorization constraints of the deployed actor.

In words, refined records that collapse to the same deployed input must prescribe behavior the deployed actor can express from that input. In the finite tabular examples, this is just row normalization. On refined records with positive row mass,

$$p^{\mathrm{ref}}(a \mid x) = \frac{d^{\mathrm{ref}}(x, a)}{\sum_{a'} d^{\mathrm{ref}}(x, a')}.$$

For unrestricted reactive policies, coherence with respect to a refinement means that all refined records with the same deployed input $c(x)$ have identical normalized rows. For decentralized product policies, equality of normalized rows is not sufficient: the common row must also factorize in a way the actors can implement. If $c(x) = (o_1, \ldots, o_n)$ and $a = (a_1, \ldots, a_n)$, the deployed policy must have the form

$$\pi(a \mid c(x)) = \prod_i \pi_i(a_i \mid o_i),$$

with no convention signal shared across agents. Coherence therefore also requires representability inside this product class, denoted $\Pi^{\mathrm{cb}}_{\mathrm{prod}}$, where cb means convention blind. We write $\Pi_{\mathrm{react}} = \{\pi(a \mid o)\}$ for the reactive single-agent class. The same local kernels $\{\pi_i\}$ must represent all deployed input cells simultaneously; it is not enough for each joint row to factorize in isolation if this would assign different local marginals to the same local observation in different cells.

When a log records only the deployed input $c(x)$, it can show the average behavior at each deployed input, but not the refinement that produced it. That average may be representable even when the uncollapsed reference is not coherent with respect to the relevant refinement. Thus collapsed logs

alone do not diagnose whether pooling is coherent, nor which coherent slice reference should be selected; that requires extra context. Appendix A.1 gives the formal projection and measurability details.

Slices. A slice is a grouping of offline data used during training to form a candidate reference. Slices may come from known behavior modes, hidden branches, trajectory clusters, or conventions in a multi-agent dataset. They are used to replace a pooled reference that is incoherent with respect to those distinctions by candidate references that may be coherent with the deployed interface. Let $z \in \mathcal{Z}$ index slices, let $d^z_{\mathcal{D}}$ be the normalized slice occupancy, and let

$$d_{\mathcal{D}} = \sum_{z \in \mathcal{Z}} \rho(z) d^z_{\mathcal{D}}$$

be the pooled occupancy with empirical slice weights ρ. Slice labels are training annotations used to build references. They are not part of the deployed interface.

We use *reference occupancy* for any occupancy used inside the regularizer, *pooled reference* for $d_{\mathcal{D}}$, *slice reference* for $d^z_{\mathcal{D}}$, and *selected reference* for the slice or geometric reference chosen by CS-DICE.

Reference coherence concerns action conditionals; it does not require the full reference occupancy to be realizable under the environment dynamics. In particular, a deployed policy can satisfy these action-law constraints and still have low return.

4 Closest Slice DICE

CS-DICE keeps the deployed policy class fixed while changing the regularization reference. It treats slices as candidate references, selects the one closest to the current occupancy estimate, and regularizes against that selected reference. Both CS-DICE and the pooled baseline deploy policies from the same class using the same observations. We first describe the hard selector, then its soft relaxation, and finally the reverse-KL instantiation used in the experiments.

4.1 Closest slice reference selection

Let $\mathcal{D}_{\mathrm{dep}}$ be the set of normalized occupancies induced by the deployed policy class, and let $D_f(d \,\|\, d^{\mathrm{ref}})$ be the occupancy f-divergence from candidate occupancy to reference occupancy. A slice is "closest" when it minimizes this divergence to the current learner occupancy; slice costs are recomputed relative to the learner's current occupancy. A pooled baseline with a single reference uses $d_{\mathcal{D}}$ and solves

$$\max_{d \in \mathcal{D}_{\mathrm{dep}}} V(d) - \alpha D_f(d \,\|\, d_{\mathcal{D}}).$$

Hard CS-DICE replaces $d_{\mathcal{D}}$ by the closest slice reference under the same divergence:

$$z^\star(d) \in \arg\min_{z \in \mathcal{Z}} D_f(d \,\|\, d^z_{\mathcal{D}}).$$

The hard objective is

$$\max_{d \in \mathcal{D}_{\mathrm{dep}}} V(d) - \alpha \min_{z \in \mathcal{Z}} D_f(d \,\|\, d^z_{\mathcal{D}}).$$

It is nonsmooth because the selected slice changes with d. We therefore view hard CS-DICE as a block procedure. At the current occupancy estimate, it chooses the closest slice and then performs a standard DICE update against that slice using the same D_f.

Algorithm 1 Hard f-Divergence Closest Slice Reference Selection

Require: slice occupancies $\{d^z_{\mathcal{D}}\}_{z \in \mathcal{Z}}$, divergence D_f, weight α

1: initialize a deployed policy occupancy d
2: **repeat**
3: compute slice costs $D_f(d \,\|\, d^z_{\mathcal{D}})$ for $z \in \mathcal{Z}$
4: select $z^\star \in \arg\min_z D_f(d \,\|\, d^z_{\mathcal{D}})$
5: set the training reference to $d^{z^\star}_{\mathcal{D}}$
6: perform a DICE update over $d \in \mathcal{D}_{\mathrm{dep}}$ against $d^{z^\star}_{\mathcal{D}}$ using D_f
7: **until** training stops

Hard selection chooses one reference for a single learner. The separate-slice comparator instead trains and evaluates an independent policy on each partition. Oracle versus inferred slices describes how the partition is obtained, independently of whether reference selection is hard or soft.

Practical cost estimation. The finite experiments compute slice costs from smoothed empirical occupancies. The neural implementation uses one CoMA-DICE critic/mixer to obtain normalized minibatch occupancy-ratio proxy weights, with the critic/mixer shared across slices. Scoring $K = |\mathcal{Z}|$ references costs $O(|B|K)$ on a minibatch B; storing transition-slice probabilities takes $O(NK)$ space for N replay transitions. Refreshing the normalized geometric reference over replay support also takes $O(NK)$. Actor, critic, and mixer updates use the single selected reference.

4.2 Soft selector

Hard selection can be smoothed by putting a distribution over slices instead of choosing a single minimizer. Given a slice prior u, temperature τ, and an entropy or proximal regularizer $\Omega(q, u)$, the soft CS-DICE objective is

$$\max_{d \in \mathcal{D}_{\mathrm{dep}},\, q \in \Delta(\mathcal{Z})} \left[V(d) - \alpha \sum_z q(z) D_f(d \,\|\, d^z_{\mathcal{D}}) - \alpha\tau\, \Omega(q, u) \right].$$

For a fixed candidate occupancy d, the selector is the optimizer over q:

$$q^\star_d \in \arg\min_{q \in \Delta(\mathcal{Z})} \left[\sum_z q(z) D_f(d \,\|\, d^z_{\mathcal{D}}) + \tau\, \Omega(q, u) \right].$$

With $\Omega(q, u) = \mathrm{KL}(q \,\|\, u)$, provided the denominator is positive, the selector has the Gibbs form

$$q^\star_d(z) = \frac{u(z) e^{-D_f(d \,\|\, d^z_{\mathcal{D}})/\tau}}{\sum_{z'} u(z') e^{-D_f(d \,\|\, d^{z'}_{\mathcal{D}})/\tau}}.$$

Small τ approximates hard nearest slice selection among positive-prior slices, while finite τ smooths the choice. A soft selector can combine incompatible slices, so its reference may remain incoherent.

4.3 Reverse-KL CS-DICE

Our main instantiation sets D_f to reverse KL, meaning the KL from the candidate occupancy to the reference. Specializing the soft objective to $\Omega(q, u) = \mathrm{KL}(q \,\|\, u)$ gives

$$\max_{d \in \mathcal{D}_{\mathrm{dep}},\, q \in \Delta(\mathcal{Z})} \left[V(d) - \alpha \sum_z q(z) \mathrm{KL}(d \,\|\, d^z_{\mathcal{D}}) - \alpha\tau\, \mathrm{KL}(q \,\|\, u) \right].$$

For a fixed selector q, the classical logarithmic opinion pool [Heskes, 1997, Section 2, Eqs. (1)–(3)] gives the reference

$$\tilde{d}_q(y) = \prod_{z \in \mathrm{supp}(q)} d^z_{\mathcal{D}}(y)^{q(z)}, \qquad Z_q = \sum_y \tilde{d}_q(y), \qquad \bar{d}_q = \tilde{d}_q / Z_q.$$

When $Z_q > 0$,

$$\sum_z q(z) \mathrm{KL}(d \,\|\, d^z_{\mathcal{D}}) = \mathrm{KL}(d \,\|\, \bar{d}_q) - \log Z_q.$$

Thus the fixed-selector occupancy update uses a single reverse-KL DICE reference. The normalization term is constant with respect to d and remains part of the selector objective.

Thus, after the selector is fixed, the policy update is an ordinary reverse-KL DICE update against the geometric mean of the active slice occupancies. If active slices have disjoint support, this geometric reference has zero mass and the identity must be interpreted in the extended value sense. We smooth occupancies over the finite pooled tuple or replay support. Hard selection needs only the chosen slice support. For continuous occupancy densities, the displayed sums have integral analogues. Appendix A.4–A.5 state the support conditions and the extended value version.

4.4 Cooperative instantiation: CS-CoMA-DICE

CoMA-DICE applies DICE regularization in cooperative CTDE. Its decomposed value objective estimates nonnegative occupancy ratio weights on offline joint transition tuples

$$\xi = (o, a, r, o', \text{done}), \qquad o = (o_1, \ldots, o_n), \quad a = (a_1, \ldots, a_n).$$

For a reference occupancy d^{ref}, write these weights as $w_{d^{\text{ref}}}(\xi)$. They define the occupancy proxy

$$\hat{d}_{d^{\text{ref}}}(\xi) \propto w_{d^{\text{ref}}}(\xi) d_{\mathcal{D}}(\xi).$$

The local actor update is weighted BC:

$$\max_{\pi=\prod_i \pi_i} \sum_{i=1}^{n} \mathbb{E}_{\xi\sim d_{\mathcal{D}}} \left[w_{d^{\text{ref}}}(\xi) \log \pi_i(a_i \mid o_i) \right].$$

CS-CoMA-DICE changes the reference in this weighting step:

$$d^{\text{ref}} = d_{\mathcal{D}} \quad \text{for pooled CoMA-DICE,} \qquad d^{\text{ref}} = d_{\mathcal{D}}^{z^\star} \quad \text{or} \quad \bar{d}_q \quad \text{for CS-CoMA-DICE.}$$

Slice labels shape the training weights; deployed agents use local observations. The controlled learner enumerates tuple weights, whereas MaMuJoCo retains the CoMA-DICE per-agent networks and state-conditioned mixing network. The controlled and neural learners apply reference selection through their respective optimization procedures. Appendix A.14 records smoothing, selector refresh, and symmetry-breaking initialization; global convergence of the alternating updates remains open.

The weighted local-policy extraction identity explains why the reference swap can survive decentralized extraction. Weighted BC projects the weighted joint tuple distribution onto local action marginals. A selected coherent slice can preserve one convention through those marginals. A pooled reference over incompatible conventions can instead produce mixed local marginals, and the final product actor then averages the conventions.

5 Theory

The results below show what reference incoherence does to the regularized objective. In the witnesses before the reveal, an incoherent pooled reference makes reverse-KL favor the mixture projection, whose return can be lower than either successful mode. Exact slice references recover high return solutions by supplying a coherent action law to the regularizer. In the control after the reveal, the pooled reference is already coherent, and pooled reverse-KL succeeds as smoothing vanishes. The examples use finite horizon normalized occupancies.

5.1 Aliasing before the reveal

Consider a single agent example with two decisions and latent states s_0, s_L, s_R. At the first decision, the agent observes o_0, with $o_0 \neq \bar{o}$. Action 0 moves to s_L, and action 1 moves to s_R. At the second decision, both s_L and s_R emit the same observation $\bar{o}$, so a reactive deployed policy must use one action conditional for both branches. The terminal reward is

$$r(s_L, 0) = r(s_R, 1) = 1, \qquad r(s_L, 1) = r(s_R, 0) = 0.$$

The successful trajectories of length two are $0{\to}0$ and $1{\to}1$. The pooled reference asks for different second step actions at s_L and s_R, although $c(s_L) = c(s_R) = \bar{o}$. Thus the pooled reference is not coherent with respect to a refinement that distinguishes s_L from s_R. The two deterministic successful policies, $0{\to}0$ and $1{\to}1$, are the successful endpoints. After full support smoothing, the pooled occupancy with respect to the branch refinement on coordinates $((s_0, 0), (s_0, 1), (s_L, 0), (s_L, 1), (s_R, 0), (s_R, 1))$ is

$$d_{\mathcal{D}} = \left(\tfrac{1}{4}, \tfrac{1}{4}, \tfrac{a}{2}, \tfrac{b}{2}, \tfrac{b}{2}, \tfrac{a}{2}\right), \qquad a > b > 0, \quad a + b = \tfrac{1}{2}.$$

Each endpoint has return 1. Write $p = \pi(1 \mid o_0)$ and $r = \pi(1 \mid \bar{o})$ for the reactive policy parameters.

Theorem 1 (Aliased DICE counterexample)**.** *For the two-step construction above, set* $B_\alpha = 1 + \frac{\alpha}{2}\log(a/b)$*. If* $\alpha > 0$ *and* $|B_\alpha| < \sqrt{2}\alpha$*, the unique maximizer of*

$$V(d_\pi) - \alpha \mathrm{KL}(d_\pi \,\|\, d_{\mathcal{D}})$$

over the reactive class $\pi_{p,r}$ *is* $p = r = 1/2$*, with return* $1/2$*. Each coherent endpoint has return* 1*.*

This construction makes the information mismatch studied in asymmetric imitation learning [Warrington et al., 2021] explicit in an occupancy objective with a finite regularization weight. Appendix A.6 proves the optimizer claim by strict concavity.

Under this condition, the reverse-KL penalty for leaving the pooled reference exceeds the return gain from committing to an endpoint. The closed form is specific to reverse-KL, but Appendix A.3 states the general margin condition under which other divergences also exclude the endpoints.

Corollary 1 (Exact-support recovery)**.** *For the exact slices* $d^0_{\mathcal{D}} = (\frac{1}{2}, 0, \frac{1}{2}, 0, 0, 0)$ *and* $d^1_{\mathcal{D}} = (0, \frac{1}{2}, 0, 0, 0, \frac{1}{2})$*, a uniform slice prior, and an admissible* f *satisfying Assumption 1 with* $f_\infty = \infty$*, the soft closest-slice objective has exactly two maximizers over* Π_{react}*:* $\pi_{0,0}$ *and* $\pi_{1,1}$*. For every* $\alpha, \tau > 0$*, both have return* 1 *and objective* $1 - \alpha\tau\log 2$*.*

Proof. Finite divergence to either exact slice forces the corresponding endpoint policy. Every other reactive policy incurs infinite divergence to both slices. At either endpoint the soft-min penalty is $\tau \log 2$. □

The corollary relies on disjoint slice supports. Smoothing replaces the infinite support penalty by a finite one. Appendix A.13 gives the reverse-KL separation calculation for the four-coordinate cooperative witness; the smoothed bound applies to the cooperative witness analyzed there.

By contrast, pooling is harmless when the branch is revealed before the final action. The second observation identifies the branch, so the pooled reference is coherent. In this version of the fork, pooled reverse-KL approaches return 1 as smoothing vanishes and attains return 1 with exact support. Appendix A.8 gives the formal statement and proof.

5.2 Cooperative product policies

A cooperative version of the same failure appears when two agents must coordinate without observing a convention signal. Each agent chooses an action in $\{0, 1\}$, deployment uses a product policy, and the shared reward is

$$r(a_1, a_2) = \mathbf{1}\{a_1 = a_2\}.$$

The coherent conventions are 00 and 11. A balanced pooled reference over these conventions has local $50/50$ marginals, so its product projection mixes them even though either endpoint succeeds. The pooled joint law can encode correlation, while the deployed product actor can only combine local action conditionals. Smoothing the pooled occupancy to

$$d_{\mathcal{D}} = (a, b, b, a), \qquad a > b > 0, \quad a + b = \tfrac{1}{2},$$

keeps every action pair in support but does not remove the projection effect. Theorem 2 shows that the direct reverse-KL product policy objective can still select the midpoint. The controlled cooperative pipeline in Section 6 adds tuple weighting and weighted BC actor extraction, and the weighted local-policy extraction identity connects that pipeline to the same local marginal mixing failure.

Theorem 2 (Cooperative DICE counterexample)**.** *On the cooperative witness above, write* $p_i = \pi_i(1)$ *and let* $B_\alpha = 1 + \alpha\log(a/b)$*. If* $\alpha > 0$ *and* $|B_\alpha| < 2\alpha$*, the pooled reverse-*KL *objective over product policies has the unique maximizer* $p_1 = p_2 = 1/2$*. Its return is* $1/2$*, while either coherent endpoint attains* 1*.*

Writing $m_i = 2p_i - 1$ and $H(m) = h((1 + m)/2)$ gives

$$J_\alpha/\alpha = H(m_1) + H(m_2) + \frac{B_\alpha}{2\alpha} m_1 m_2 + C.$$

This is the two-spin mean-field Ising variational objective [Wainwright and Jordan, 2008, Section 5.4]. Under the displayed condition its Hessian is negative definite and symmetry places its maximizer at $m_1 = m_2 = 0$.

Appendix A.12 gives the proof.

Weighted local-policy extraction. Weighted behavior cloning over product policies separates into local maximum-likelihood problems [Bui et al., 2025, Eqs. (8)–(9)]. For finite local action sets and nonnegative tuple weights, let $C_i^w(o_i, a_i)$ be the weighted empirical count and $C_i^w(o_i) = \sum_{a_i} C_i^w(o_i, a_i)$. At every observation with positive count,

$$\pi_i^\star(a_i \mid o_i) = \frac{C_i^w(o_i, a_i)}{C_i^w(o_i)}.$$

The objective is indifferent at zero-count observations. The formula follows by expanding $\log \prod_i \pi_i = \sum_i \log \pi_i$ and normalizing the counts. A multimodal weighted joint law can yield mixed local marginals and poorly coordinated product actions [Matsunaga et al., 2023, Appendix C]. In CS-CoMA-DICE, reference selection changes the tuple weights entering this extraction step.

Thus pooled reverse-KL selects a product projection with mixed local marginals, and weighted behavior cloning preserves that mixing rather than the correlated joint convention. Appendix A.10 extends the same projection failure to the balanced n-agent case, where the midpoint return is 2^{1-n}.

6 Experiments

The experiments are organized around three questions. The first asks whether coherence predicts when the pooled reference helps or hurts. The second asks whether representations built without rewards can recover the structure needed to choose a coherent reference. The third asks whether the same selected reference update can be inserted into a neural CoMA-DICE pipeline using the original decentralized actors. The controlled tasks answer the first two questions. MaMuJoCo tests the third.

Protocol. CS-DICE with oracle slices uses the true training partition and isolates the reference selection mechanism. Oracle slices need not be coherent: the noisy one-step slices do not factorize. CS-DICE with inferred slices builds the partition from representations without rewards. Separate slice training fits one policy per slice and averages their independently evaluated returns, providing a diagnostic average over separately deployed policies. Controlled main tables report mean $\pm$ sample standard deviation over paired seeds $\{0, 1, 2, 3, 4\}$, using fixed final checkpoints. Separate slice rows first average slice policy returns within each seed. Cooperative rows in these main tables use tuple weighting followed by decentralized weighted BC extraction, with the theorem's direct product-policy objective evaluated separately in Appendix A.16. Implementation details are in Appendix A.14.

We use the following controlled benchmarks.

- **Aliased single agent.** This is an observation-level analogue of the branch example with two decisions from the introduction. The branch matters for success, but both branches share the same deployed observation at the final action. This implementation uses one observation at both decisions and four observation-action coordinates, rather than the theorem's six refined-state/action coordinates.
- **Single agent after the reveal.** This is the same fork after the relevant branch information is revealed before the conflicting action. It tests the boundary case where pooling should be benign.
- **Cooperative one step.** Two decentralized agents must choose a shared convention without observing a convention signal. It tests whether pooled tuple weights induce mixed local marginals.
- **Horizon-2.** Successful trajectories have the same summary statistics but different temporal order. It tests whether slice inference preserves the temporal structure needed for a coherent reference.
- **ConventionRepeat-HH.** The cooperative convention is repeated across overlapping histories. It tests the same reference choice mechanism in a longer temporal setting.
- **Cooperative exact after the reveal.** The relevant cooperative information is available before the conflicting action. It is the cooperative boundary control.

Table 1 shows the predicted split. Before the reveal, pooled regularization converges to low return behavior, while CS-DICE with oracle slices stays near the successful modes. Horizon-2 improves

Table 1: Controlled mechanism and boundary results. Entries are mean ± standard deviation over five seeds.

Benchmark	Role	Pooled reference	Pooled reference update	CS-DICE, oracle slices	Separate slice comparison
Aliased single-agent	aliasing before the reveal	incoherent	0.5002 ± 0.0002	0.9434 ± 0.0073	0.9436 ± 0.0042
Single-agent post-reveal	after the reveal control	coherent	0.9924 ± 0.0134	0.9999 ± 0.0001	0.9999 ± 0.0001
Cooperative one-step	product mixing	incoherent	0.500 ± 0.000	0.988 ± 0.000	0.985 ± 0.000
Horizon-2	temporal alias	incoherent	0.125 ± 0.000	0.988 ± 0.000	0.976 ± 0.000
ConventionRepeat-HH	repeat overlap	incoherent	0.0005 ± 0.0000	0.9220 ± 0.0000	0.8513 ± 0.0000
Cooperative post-reveal exact	after the reveal control	coherent	0.9843 ± 0.0000	0.9837 ± 0.0000	–

Table 2: Supplied and inferred slices using tuple weighting and weighted BC. Entries are mean ± standard deviation over paired seeds.

Benchmark	Slices	Acc.	CS-DICE	Separate slice
Cooperative one-step	Supplied	–	0.988 ± 0.000	0.985 ± 0.000
Cooperative one-step	Summary	0.52 ± 0.03	0.502 ± 0.003	0.334 ± 0.001
Horizon-2	Supplied	–	0.988 ± 0.000	0.976 ± 0.000
Horizon-2	Summary	0.50 ± 0.00	0.125 ± 0.000	0.125 ± 0.000
Horizon-2	Temporal	1.00 ± 0.00	0.988 ± 0.000	0.976 ± 0.000
ConventionRepeat-HH	Supplied	–	0.9220 ± 0.0000	0.8513 ± 0.0000
ConventionRepeat-HH	Summary	0.90 ± 0.21	0.7467 ± 0.4171	0.6848 ± 0.3823

from 0.125 to 0.988, and ConventionRepeat-HH improves from 0.0005 to 0.9220. The boundary rows show the complementary case. When the relevant information is visible before the conflicting action, the pooled reference is coherent and pooled training is already near perfect. In the cooperative rows, the same pattern appears after tuple weighting and decentralized weighted BC extraction, as described by the weighted local-policy extraction identity.

We also ask whether the slices have to be supplied. The inferred rows use two representations built without rewards. Summary features compress each episode, while temporal features cluster GRU embeddings trained by reward-free next-step prediction. Accuracy is shown only for inferred slices.

Table 2 compares representations within the same tuple-weighting pipeline. Summary features fail on one-step and Horizon-2; temporal features recover Horizon-2. ConventionRepeat-HH temporal results use a separate direct-policy objective with $\alpha = 1$, rather than the main pipeline's $\alpha = 0.25$, and are reported with their supplied-slice controls in Appendix A.18. Later appendix sections report label-noise and selector-temperature checks.

MaMuJoCo continuous control. We compare pooled and selected references in the same neural CoMA-DICE stack using reverse KL for both methods and $K = 2$ reward-free summary clusters. The evaluation covers Hopper-v2, HalfCheetah-v2, and Ant-v2, each with expert, medium, medium-replay, and medium-expert data: 120 runs across twelve cells, with five seeds per method. Every run uses 25 epochs and 100 prespecified evaluations of 32 episodes. Fixed final checkpoints are the primary score; per-seed maxima over the same evaluation budget measure observed peak return. Appendix A.15 gives full returns, uncertainty, and implementation details.

Table 3: MaMuJoCo selected/pooled return ratios, equally averaged over robot/dataset cells with five seeds per method. Higher-mean counts use fixed-final scores and are descriptive. Best compares means of per-seed maxima over the same 100 evaluations; it is a hindsight diagnostic.

Robot	Cells	Higher final mean	Final ratio	Best ratio
Hopper-v2	4	2/4	0.70	0.92
HalfCheetah-v2	4	1/4	0.89	1.02
Ant-v2	4	3/4	1.08	1.01
All cells	12	6/12	0.89	0.98

CS-CoMA-DICE has a higher final mean in 6/12 cells, with a mean selected/pooled ratio of 0.89. Best-versus-best yields a mean ratio of 0.98, and selected best means reach at least 95% of pooled best means in 10/12 cells. At the final checkpoint, Ant expert and medium improve by 18.7% and 14.0%, whereas Hopper medium declines by 86.1%. Both methods show substantial best-to-final degradation. These descriptive comparisons establish neither a general advantage nor statistical significance. Best scores are hindsight diagnostics based on online evaluation, not an offline checkpoint selection rule. The clusters have no annotated semantic convention, so the controlled tasks provide the direct mechanism evidence.

7 Limitations

We study deployment through a single reactive or decentralized product policy, with slice information available only during training. The selector formulation accommodates other divergences; the closed-form midpoint analysis and neural implementation use reverse KL. The experiments cover controlled mechanisms and continuous control, with broader function-approximation tests and comparisons against multimodal policies left open.

Reliable slice construction remains essential: temporal representations can preserve relevant structure that summary features erase. Hard selection also need not yield a coherent slice reference. A soft selector can assign weight to incompatible slices and need not define a coherent reference. The divergence-matched MaMuJoCo results show competitive peaks and mixed final returns. Hindsight maxima do not provide an offline policy-selection metric or identify the cause of return degradation. The HalfCheetah medium-expert cell combines CUDA and ARM CPU execution, limiting hardware-controlled comparisons. Neither semantic convention recovery nor the prevalence of reference incoherence in standard offline datasets is established. The formal coherence condition is separate from assignment stability, nearest-slice divergence, and partition sensitivity.

The selector is global for an occupancy update and cannot choose different references for individual observation cells or trajectory phases. The choice of support smoothing, initialization, and slice quality remains a determinant of return.

8 Conclusion

Reference coherence makes the regularization target part of the deployment specification: its action conditionals must use the available information and fit the deployed policy class. Our theory and controlled experiments identify failures caused by hidden branches and joint conventions, recovery conditions with coherent slices, and a boundary where revealing the relevant information makes pooling benign. Inferred slices help when their representations retain that structure.

CS-DICE chooses the training reference while keeping the deployed actor unchanged. Its neural CoMA-DICE implementation reaches competitive observed peaks with mixed final returns. Together, these results distinguish useful behavior in a dataset from an appropriate occupancy reference for deployment, and make retention of peak performance a practical limitation to address.

Acknowledgments and Disclosure of Funding

The author funded this work entirely personally and received no third-party financial support. The author declares no competing financial interests.

References

J. Birrell, P. Dupuis, M. A. Katsoulakis, Y. Pantazis, and L. Rey-Bellet. (f,Γ)-divergences: Interpolating between f-divergences and integral probability metrics. *Journal of Machine Learning Research*, 23(39):1–70, 2022. URL http://jmlr.org/papers/v23/21-0100.html.

T. V. Bui, H. T. Nguyen, and T. Mai. ComaDICE: Offline cooperative multi-agent reinforcement learning with stationary distribution shift regularization. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/477d1531c260f9605bbf9d14d00952d4-Abstract-Conference.html.

J. Chen, B. Ganguly, T. Lan, and V. Aggarwal. Variational offline multi-agent skill discovery. *arXiv preprint arXiv:2405.16386*, 2024. URL https://arxiv.org/abs/2405.16386v1. Version 1.

X. Chen, A. Ghadirzadeh, T. Yu, J. Wang, Y. Gao, W. Li, B. Liang, C. Finn, and C. Zhang. LAPO: Latent-variable advantage-weighted policy optimization for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/efb2072a358cefb75886a315a6fcf880-Abstract-Conference.html.

B. Cui, H. Hu, A. Lupu, S. Sokota, and J. Foerster. Off-team learning. In *Advances in Neural Information Processing Systems*, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/631f99d8e860054410c239fc90d18270-Abstract-Conference.html.

C. Formanek, A. Jeewa, J. Shock, and A. Pretorius. Off-the-grid MARL: Datasets with baselines for offline multi-agent reinforcement learning. *arXiv preprint arXiv:2302.00521*, 2023. URL https://arxiv.org/abs/2302.00521.

C. Formanek, L. Beyers, C. R. Tilbury, J. P. Shock, and A. Pretorius. Putting data at the centre of offline multi-agent reinforcement learning. *arXiv preprint arXiv:2409.12001*, 2024. URL https://arxiv.org/abs/2409.12001.

J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine. D4RL: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020. URL https://arxiv.org/abs/2004.07219.

T. Heskes. Selecting weighting factors in logarithmic opinion pools. In M. Jordan, M. Kearns, and S. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. MIT Press, 1997. URL https://proceedings.neurips.cc/paper_files/paper/1997/file/59f51fd6937412b7e56ded1ea2470c25-Paper.pdf.

H. Hu, A. Lerer, A. Peysakhovich, and J. Foerster. Other-Play for zero-shot coordination. In *International Conference on Machine Learning*, 2020. URL https://proceedings.mlr.press/v119/hu20a.html.

H. Hu, X. Wang, and S. S. Du. Policy-based trajectory clustering in offline reinforcement learning. *arXiv preprint arXiv:2506.09202*, 2025. URL https://arxiv.org/abs/2506.09202.

A. Kumar, A. Zhou, G. Tucker, and S. Levine. Conservative Q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2020. URL https://arxiv.org/abs/2006.04779.

A. Kumar, R. Agarwal, X. Geng, G. Tucker, and S. Levine. Offline q-learning on diverse multi-task data both scales and generalizes. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=4-k7kUavAj.

J. Lee, W. Jeon, B.-J. Lee, J. Pineau, and K.-E. Kim. OptiDICE: Offline policy optimization via stationary distribution correction estimation. In *International Conference on Machine Learning*, 2021. URL https://proceedings.mlr.press/v139/lee21f.html.

Z. Liu, Q. Lin, C. Yu, X. Wu, Y. Liang, D. Li, and X. Ding. Offline multi-agent reinforcement learning via in-sample sequential policy optimization. *arXiv preprint arXiv:2412.07639*, 2024. URL https://arxiv.org/abs/2412.07639.

D. E. Matsunaga, J. Lee, J. Yoon, S. Leonardos, P. Abbeel, and K.-E. Kim. AlberDICE: Addressing out-of-distribution joint actions in offline multi-agent RL via alternating stationary distribution correction estimation. In *Advances in Neural Information Processing Systems*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/e5b6eb1dbabff82838d5e99f62de37c8-Abstract-Conference.html.

O. Nachum, Y. Chow, B. Dai, and L. Li. DualDICE: Behavior-agnostic estimation of discounted stationary distribution corrections. In *Advances in Neural Information Processing Systems*, 2019a. URL https://papers.nips.cc/paper_files/paper/2019/hash/cf9a242b70f45317ffd281241fa66502-Abstract.html.

O. Nachum, B. Dai, I. Kostrikov, Y. Chow, L. Li, and D. Schuurmans. AlgaeDICE: Policy gradient from arbitrary experience. *arXiv preprint arXiv:1912.02074*, 2019b. URL https://arxiv.org/abs/1912.02074.

F. A. Oliehoek and C. Amato. *A Concise Introduction to Decentralized POMDPs*. Springer, 2016. doi: 10.1007/978-3-319-28929-8. URL https://doi.org/10.1007/978-3-319-28929-8.

T. Osa, A. Hayashi, P. Deo, N. Morihira, and T. Yoshiike. Offline reinforcement learning with mixture of deterministic policies. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=zkRCp4RmAF.

L. Pan, L. Huang, T. Ma, and H. Xu. Plan better amid conservatism: Offline multi-agent reinforcement learning with actor rectification. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 17221–17237, 2022. URL https://proceedings.mlr.press/v162/pan22a.html.

B. Sarkar, A. Shih, and D. Sadigh. Diverse conventions for human-ai collaboration. In *Advances in Neural Information Processing Systems*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/4818263715b25dc137d393af8af6d2fc-Abstract-Conference.html.

C. A. Schroeder de Witt, J. N. Foerster, G. Farquhar, P. H. S. Torr, W. Boehmer, and S. Whiteson. Multi-agent common knowledge reinforcement learning. *arXiv preprint arXiv:1810.11702*, 2018. URL https://arxiv.org/abs/1810.11702.

G. Taraldsen. Optimal learning from the Doob-Dynkin lemma, 2018. URL https://arxiv.org/abs/1801.00974.

C. R. Tilbury, C. Formanek, L. Beyers, J. P. Shock, and A. Pretorius. Coordination failure in cooperative offline MARL. *arXiv preprint arXiv:2407.01343*, 2024. URL https://arxiv.org/abs/2407.01343.

J. Treutlein, M. Dennis, C. Oesterheld, and J. Foerster. A new formalism, method and open issues for zero-shot coordination. In *International Conference on Machine Learning*, 2021. URL https://proceedings.mlr.press/v139/treutlein21a.html.

M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2008. doi: 10.1561/2200000001. URL https://doi.org/10.1561/2200000001.

M. Wang, Y. Jin, and G. Montana. Learning on one mode: Addressing multi-modality in offline reinforcement learning. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/be62c4a943675195ff5a2a98d5b9724f-Abstract-Conference.html.

A. Warrington, J. W. Lavington, A. Scibior, M. Schmidt, and F. Wood. Robust asymmetric learning in POMDPs. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11013–11023. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/warrington21a.html.

A Additional Derivations and Proofs

This appendix contains the measure-theoretic form of reference coherence with respect to a refinement, selector identities, oracle gap and perturbation bounds, exact slice admissible f proofs, the reverse KL reduction, the actor extraction bridge, additional controlled experiments, MaMuJoCo integration tables, and reproducibility details. The main paper presents CS-DICE as a closest-slice principle for choosing the regularization reference: the positive selector results and the exact-slice aliased single agent recovery corollary are developed for admissible f divergences, while reverse KL is the primary worked optimization instantiation used in the controlled suite and the MaMuJoCo CS-CoMA-DICE integration study.

A.1 Reference coherence and deployed measurability

Let U, O, and A be standard Borel measurable spaces. Here U, u correspond to X, x in Section 3. The space U refines the deployed input O. It records distinctions that may have been used during data generation or analysis and are collapsed before deployment; it need not appear in the raw log. Let $c : U \to O$ be the deployment map and let Π_{dep} be the deployed policy class on O. For a normalized reference occupancy d^{ref} on $U \times A$, let m^{ref} be its U-marginal and let $p^{\mathrm{ref}}(\cdot \mid u)$ be a regular conditional action law.

We say d^{ref} is coherent with respect to the refinement and deployed class (c, Π_{dep}) if there exists a deployed policy $\pi \in \Pi_{\mathrm{dep}}$ such that

$$p^{\mathrm{ref}}(\cdot \mid u) = \pi(\cdot \mid c(u)) \quad m^{\mathrm{ref}}\text{-a.e. } u.$$

For unrestricted deployed kernels on O, this is equivalent to saying that the map $u \mapsto p^{\mathrm{ref}}(\cdot \mid u)$ is $\sigma(c)$-measurable m^{ref}-almost everywhere. Equivalently, for every measurable action event $B \subseteq A$, the scalar map $u \mapsto p^{\mathrm{ref}}(B \mid u)$ is $\sigma(c)$-measurable. By the Doob–Dynkin factorization lemma [Taraldsen, 2018, Lemma 5], $\sigma(c)$-measurability yields a deployed-input kernel κ satisfying

$$p^{\mathrm{ref}}(\cdot \mid u) = \kappa(\cdot \mid c(u)) \quad m^{\mathrm{ref}}\text{-a.e. } u.$$

With a restricted deployed class, this kernel must also lie in Π_{dep}. For decentralized execution, this includes product factorization and local consistency constraints. Equivalently, for unrestricted kernels, coherence says that under the reference law $A \perp U \mid c(U)$. In finite tabular examples, the condition reduces to equality of normalized rows across all refined records u, u' with $c(u) = c(u')$, plus any deployed-class constraints such as product factorization.

Assumption 1 (Admissible generators for positive closest-slice results)**.** *For the positive selector results below, let $f : [0, \infty) \to \mathbb{R} \cup \{+\infty\}$ be convex, lower semicontinuous, and normalized by $f(1) = 0$. Define*

$$f_\infty := \lim_{u \to \infty} \frac{f(u)}{u}.$$

The exact-support corollary assumes $f_\infty = \infty$. The finite ε smoothed support bound below is proved directly for reverse KL using the explicit support mismatch lower bound developed there. Finiteness on some neighborhood of 1 is a sufficient condition for right continuity at 1, while the reverse KL bound uses the stronger worked structure.

Theorems 1 and 2, together with Corollary 1, are stated in Section 5. This appendix gives their full proofs. It also records the fork boundary after the reveal, the weighted projection onto local marginals, a reverse KL finite ε selector bound, the experimental tables behind Tables 1 and 2, the MaMuJoCo CoMA-DICE and CS-CoMA-DICE tables, and the protocol details needed to reproduce the experiments.

A.2 Variational form of the closest-slice regularizer

Let $u \in \Delta(\mathcal{Z})$ be a prior over slices, let $\tau > 0$, and let

$$D_z := D_f(d \,\|\, d^z_{\mathcal{D}})$$

denote the disparity between a candidate occupancy d and slice z. Define

$$\Psi_{\tau,f}(d) = -\tau \log \left(\sum_{z \in \mathcal{Z}} u(z)\, e^{-D_z/\tau} \right).$$

If the displayed sum is zero, we set $\Psi_{\tau,f}(d) = +\infty$.

Gibbs variational identity for slice costs. The Gibbs variational principle [Birrell et al., 2022, Eqs. (12)–(13)] gives

$$\Psi_{\tau,f}(d) = \min_{q \in \Delta(\mathcal{Z})} \left\{ \sum_z q(z) D_z + \tau \mathrm{KL}(q \,\|\, u) \right\}.$$

For $Z = \sum_z u(z) e^{-D_z/\tau} > 0$, the unique minimizer is $q^\star(z) = u(z) e^{-D_z/\tau}/Z$. *Proof.* On $\mathcal{Z}_{\mathrm{fin}} = \{z : u(z) > 0,\ D_z < \infty\}$, substitution gives

$$\sum_z q(z) D_z + \tau \mathrm{KL}(q \,\|\, u) = -\tau \log Z + \tau \mathrm{KL}(q \,\|\, q^\star).$$

Selectors assigning positive mass outside $\mathcal{Z}_{\mathrm{fin}}$ have infinite objective. Nonnegativity of $\mathrm{KL}(q \,\|\, q^\star)$ yields the minimum. If $Z = 0$, every positive-prior slice has infinite cost and both sides equal $+\infty$.

A.3 Extensions beyond reverse KL

The Gibbs variational identity and soft-min bounds apply to generic slice costs D_f. Exact-support recovery additionally uses $f_\infty = \infty$. In particular, normalized Pearson χ^2 and standard Cressie–Read generators that satisfy Assumption 1 fall inside the same closest-slice template.

The negative pooled result has two levels. The general mechanism is pooled f projection failure against a designated comparator family: if the pooled f projection onto the deployed class is closer to the pooled reference than the relevant endpoint policies are, then sufficiently strong regularization excludes those endpoints. The exact midpoint location depends on the divergence. Reverse KL gives that solution on the canonical witnesses, where the entropy decomposition gives a closed form global optimization proof.

We develop the reverse KL case in detail because it supplies the synthetic reference reduction and the finite ε worked bound used by the main experiments. The closest-slice formulation is general; reverse KL provides the worked optimization case.

Penalty dominance. Let $R = \sup_{\Pi_{\mathrm{dep}}} V - \inf_{\Pi_{\mathrm{dep}}} V < \infty$. Suppose $\pi_f^\star$ has finite divergence and every $\pi \in \Pi_{\mathrm{cmp}}$ satisfies

$$D_f(d_\pi \,\|\, d_{\mathcal{D}}) \geq D_f(d_{\pi_f^\star} \,\|\, d_{\mathcal{D}}) + \Delta_f, \qquad \Delta_f > 0.$$

For finite-penalty comparisons,

$$J_{\alpha,f}(\pi_f^\star) - J_{\alpha,f}(\pi) \geq -R + \alpha \Delta_f > 0 \qquad \text{when } \alpha > R/\Delta_f.$$

An infinite penalty gives objective $-\infty$. Thus the assumed divergence gap excludes the comparator family once the regularization weight exceeds R/Δ_f.

Soft-min approximation bound. For a finite nonempty positive-prior set $\mathcal{Z}_u = \{z : u(z) > 0\}$, define $D_{\min} = \min_{z \in \mathcal{Z}_u} D_z$ and $u_{\min} = \min_{z \in \mathcal{Z}_u} u(z) > 0$. Then

$$D_{\min} \leq \Psi_{\tau,f}(d) \leq D_{\min} + \tau \log(1/u_{\min}).$$

For a uniform prior, the error is at most $\tau \log |\mathcal{Z}|$. *Proof.* The partition sum obeys

$$u_{\min} e^{-D_{\min}/\tau} \leq \sum_z u(z) e^{-D_z/\tau} \leq e^{-D_{\min}/\tau}.$$

Applying $-\tau \log$ gives the two inequalities, with the extended-value convention when all costs are infinite.

Sensitivity to bounded cost perturbations. Fix an occupancy d, a prior u, and $\tau > 0$. For finite costs D_z, $\widehat{D}_z$ on $\mathcal{Z}_u$ satisfying $\sup_{z \in \mathcal{Z}_u} |\widehat{D}_z - D_z| \leq \varepsilon$, the corresponding soft-min values and Gibbs selectors satisfy

$$|\widehat{\Psi}_{\tau,f}(d) - \Psi_{\tau,f}(d)| \leq \varepsilon,$$

$$\left|\log \frac{\widehat{q}^\star(z)/\widehat{q}^\star(z')}{q^\star(z)/q^\star(z')}\right| \le 2\varepsilon/\tau \qquad (u(z), u(z') > 0).$$

Proof. Writing $Z(D) = \sum_z u(z)e^{-D_z/\tau}$, the cost bound gives

$$e^{-\varepsilon/\tau} Z(D) \le Z(\widehat{D}) \le e^{\varepsilon/\tau} Z(D).$$

Taking logarithms yields the value bound. Subtracting the two Gibbs log-odds expressions gives $-[(\widehat{D}_z - D_z) - (\widehat{D}_{z'} - D_{z'})]/\tau$, whose absolute value is at most $2\varepsilon/\tau$.

A.4 Composite penalty and coordinate dependent generator

Fix a selector $q \in \Delta(\mathcal{Z})$ and write x for a generic coordinate. Abbreviate

$$d_z(x) := d^z_{\mathcal{D}}(x).$$

The next identities are used in the full-support regime and, more generally, on the common support domain

$$\mathcal{D}(q) := \{d \in \mathcal{D}_{\mathrm{dep}} : d(x) = 0 \text{ whenever } \exists z \in \mathcal{Z} \text{ with } q(z) > 0 \text{ and } d_z(x) = 0\}.$$

Thus every $d \in \mathcal{D}(q)$ is supported on the intersection of the supports of the active slices. Exact support mismatch arguments appear later under Assumption 1. This subsection works directly on the common support domain. Define

$$\mathcal{X}_q := \{x : \exists z \in \mathcal{Z} \text{ with } q(z)\, d_z(x) > 0\}, \qquad \mathcal{Z}_{q,x} := \{z \in \mathcal{Z} : q(z)\, d_z(x) > 0\} \quad (x \in \mathcal{X}_q),$$

and for $x \in \mathcal{X}_q$,

$$\phi_{q,x}(t) := \sum_{z \in \mathcal{Z}_{q,x}} q(z)\, d_z(x)\, f\left(\frac{t}{d_z(x)}\right).$$

On the common-support domain $\mathcal{D}(q)$, expanding each divergence and interchanging the finite sums gives the coordinatewise penalty:

$$\begin{aligned}\sum_z q(z) D_f(d \,\|\, d_z) &= \sum_{x \in \mathcal{X}_q} \sum_{z \in \mathcal{Z}_{q,x}} q(z) d_z(x) f\left(\frac{d(x)}{d_z(x)}\right) \\ &= \sum_{x \in \mathcal{X}_q} \phi_{q,x}(d(x)).\end{aligned}$$

Set $m_q(x) = \sum_z q(z) d_z(x)$ and, for $m_q(x) > 0$, $\lambda_z(x) = q(z)d_z(x)/m_q(x)$. Substituting these weights into the coordinatewise penalty yields

$$\sum_z q(z) D_f(d \,\|\, d_z) = \sum_{x \in \mathcal{X}_q} m_q(x) \bar{f}_{q,x}\left(\frac{d(x)}{m_q(x)}\right),$$

where

$$\bar{f}_{q,x}(r) = \sum_{z \in \mathcal{Z}_{q,x}} \lambda_z(x) f\left(r \frac{m_q(x)}{d_z(x)}\right).$$

The function $\bar{f}_{q,x}$ depends on the coordinate x, and the identity holds for $d \in \mathcal{D}(q)$.

A.5 Reverse KL synthetic reference

For reverse KL, let

$$D_z(d) := \mathrm{KL}(d \,\|\, d_z).$$

Given a selector $q \in \Delta(\mathcal{Z})$, define the geometric synthetic reference

$$\tilde{d}_q(x) := \prod_{z \in \mathrm{supp}(q)} d_z(x)^{q(z)}, \qquad Z_q := \sum_x \tilde{d}_q(x).$$

If $Z_q > 0$, define

$$\bar{d}_q := \frac{\tilde{d}_q}{Z_q}.$$

When $Z_q = 0$, the active slices have empty common support and the reverse KL multiple slice penalty is not representable as $\mathrm{KL}(d \,\|\, \bar{d}_q)$ against an ordinary reference occupancy. In fact, if d is a normalized occupancy and $Z_q = 0$, then $\sum_z q(z)\,\mathrm{KL}(d \,\|\, d_z) = +\infty$: any coordinate carrying positive mass under d is absent from at least one active slice. Equivalently, there is no finite reverse KL subproblem in that case.

Reverse KL fixed selector reduction. If $Z_q > 0$, then for every normalized feasible occupancy d,

$$\sum_{z \in \mathcal{Z}} q(z)\,\mathrm{KL}(d \,\|\, d_z) = \mathrm{KL}(d \,\|\, \bar{d}_q) - \log Z_q.$$

Proof. If there exists a coordinate x with $d(x) > 0$ and $\tilde{d}_q(x) = 0$, then some active slice $z \in \mathrm{supp}(q)$ has $d_z(x) = 0$. The corresponding coordinate contributes $+\infty$ to $\mathrm{KL}(d \,\|\, d_z)$, so the left side is $+\infty$. The same coordinate also has $\bar{d}_q(x) = 0$, so $\mathrm{KL}(d \,\|\, \bar{d}_q) = +\infty$ and the right side is $+\infty$ as well. Hence the identity holds in the extended value sense in every support mismatch case.

Assume from now on that $d(x) > 0$ implies $\tilde{d}_q(x) > 0$. Then every active slice has $d_z(x) > 0$ whenever $d(x) > 0$, so all logarithms below are well defined on the support of d.

Using $\sum_z q(z) = 1$,

$$\sum_z q(z)\,\mathrm{KL}(d \,\|\, d_z) = \sum_x d(x) \log d(x) - \sum_x d(x) \sum_z q(z) \log d_z(x).$$

The second term equals $\sum_x d(x) \log \tilde{d}_q(x)$, so

$$\sum_z q(z)\,\mathrm{KL}(d \,\|\, d_z) = \sum_x d(x) \log \frac{d(x)}{\tilde{d}_q(x)}.$$

Because $\tilde{d}_q = Z_q \bar{d}_q$,

$$\sum_x d(x) \log \frac{d(x)}{\tilde{d}_q(x)} = \sum_x d(x) \log \frac{d(x)}{\bar{d}_q(x)} - \log Z_q = \mathrm{KL}(d \,\|\, \bar{d}_q) - \log Z_q.$$

□

A.6 Full proof of Theorem 1

Theorem 1 (aliased single agent reverse KL midpoint optimizer). Consider the two-step single agent witness with latent states s_0, s_L, s_R, observations $o_0, \bar{o}$, deterministic transition $s_0 \xrightarrow{0} s_L$, $s_0 \xrightarrow{1} s_R$, and terminal reward $r(s_L, 0) = r(s_R, 1) = 1$, $r(s_L, 1) = r(s_R, 0) = 0$. Over the reactive class

$$\Pi_{\mathrm{react}} = \{\pi_{p,r} : p = \pi(1 \mid o_0),\ r = \pi(1 \mid \bar{o})\},$$

use equal step occupancies on coordinates

$$((s_0, 0), (s_0, 1), (s_L, 0), (s_L, 1), (s_R, 0), (s_R, 1)).$$

Let the pooled occupancy be

$$d_{\mathcal{D}} = \Big(\frac{1}{4}, \frac{1}{4}, \frac{a}{2}, \frac{b}{2}, \frac{b}{2}, \frac{a}{2}\Big), \qquad a > b > 0, \qquad a + b = \frac{1}{2}.$$

Define

$$B_\alpha := 1 + \frac{\alpha}{2} \log(a/b).$$

If

$$|B_\alpha| < \sqrt{2}\,\alpha,$$

then the unique global maximizer of

$$J^{\alpha}_{\mathrm{pool}}(\pi) = V(\pi) - \alpha\,\mathrm{KL}(d^\pi \,\|\, d_{\mathcal{D}})$$

over Π_{react} is $\pi_{1/2,1/2}$, which attains return $1/2$, while each coherent endpoint $\pi_{0,0}$ and $\pi_{1,1}$ attains return 1.

Proof. The occupancy of $\pi_{p,r}$ is

$$d^{\pi_{p,r}} = \Big(\frac{1-p}{2}, \frac{p}{2}, \frac{(1-p)(1-r)}{2}, \frac{(1-p)r}{2}, \frac{p(1-r)}{2}, \frac{pr}{2}\Big),$$

and the return is

$$V(\pi_{p,r}) = (1-p)(1-r) + pr.$$

A direct expansion of the reverse KL term against $d_{\mathcal{D}}$ yields

$$J^{\alpha}_{\mathrm{pool}}(p,r) = \alpha h(p) + \frac{\alpha}{2} h(r) + B_\alpha\big((1-p)(1-r) + pr\big) + C,$$

where $h(u) = -u\log u - (1-u)\log(1-u)$ and C is constant in (p,r). Therefore

$$\frac{\partial J^{\alpha}_{\mathrm{pool}}}{\partial p} = \alpha \log\frac{1-p}{p} + B_\alpha(2r-1), \qquad \frac{\partial J^{\alpha}_{\mathrm{pool}}}{\partial r} = \frac{\alpha}{2}\log\frac{1-r}{r} + B_\alpha(2p-1),$$

so

$$\nabla J^{\alpha}_{\mathrm{pool}}\left(\frac{1}{2}, \frac{1}{2}\right) = 0.$$

The Hessian is

$$\nabla^2 J^{\alpha}_{\mathrm{pool}}(p,r) = \begin{pmatrix} -\frac{\alpha}{p(1-p)} & 2B_\alpha \\ 2B_\alpha & -\frac{\alpha}{2r(1-r)} \end{pmatrix}.$$

The diagonal entries are always negative. Moreover,

$$\det \nabla^2 J^{\alpha}_{\mathrm{pool}}(p,r) = \frac{\alpha^2}{2p(1-p)r(1-r)} - 4B_\alpha^2.$$

Since $p(1-p) \le 1/4$ and $r(1-r) \le 1/4$,

$$\det \nabla^2 J^{\alpha}_{\mathrm{pool}}(p,r) \ge 8\alpha^2 - 4B_\alpha^2 > 0$$

under $|B_\alpha| < \sqrt{2}\alpha$. Hence the Hessian is negative definite throughout $(0,1)^2$, so $J^{\alpha}_{\mathrm{pool}}$ is strictly concave there and has at most one interior critical point.

No boundary point can be optimal. Indeed, the displayed gradient formulas imply

$$\lim_{p\downarrow 0} \frac{\partial J^{\alpha}_{\mathrm{pool}}}{\partial p} = +\infty, \qquad \lim_{p\uparrow 1} \frac{\partial J^{\alpha}_{\mathrm{pool}}}{\partial p} = -\infty,$$

uniformly over $r \in [0,1]$, because the logarithmic term diverges while $B_\alpha(2r-1)$ stays bounded. Similarly,

$$\lim_{r\downarrow 0} \frac{\partial J^{\alpha}_{\mathrm{pool}}}{\partial r} = +\infty, \qquad \lim_{r\uparrow 1} \frac{\partial J^{\alpha}_{\mathrm{pool}}}{\partial r} = -\infty,$$

uniformly over $p \in [0,1]$. Thus every boundary point can be improved by moving slightly into the interior. Since $(1/2, 1/2)$ is the only interior critical point, it is the unique global maximizer on $[0,1]^2$.

Finally,

$$V(\pi_{1/2,1/2}) = \frac{1}{2}, \qquad V(\pi_{0,0}) = V(\pi_{1,1}) = 1.$$

Thus the unique optimizer is the midpoint on the full-support reverse KL witness even though both coherent endpoints solve the task perfectly. □

A.7 Numerical analysis for the smoothed cooperative witness

For the direct-objective cooperative one-step witness used in Appendix A.16 and Figure 1, we use the parameters

$$\alpha = 1, \qquad \tau = 0.05, \qquad \varepsilon = 10^{-3}.$$

The main cooperative controlled pipeline uses the tuple weighting pipeline with regularization weight 0.25. The two smoothed slice occupancies are

$$d^{0,\varepsilon}_{\mathcal{D}} = (a_\varepsilon, b_\varepsilon, b_\varepsilon, c_\varepsilon), \qquad d^{1,\varepsilon}_{\mathcal{D}} = (c_\varepsilon, b_\varepsilon, b_\varepsilon, a_\varepsilon),$$

with

$$a_\varepsilon = \frac{2/3+\varepsilon}{1+4\varepsilon}, \qquad b_\varepsilon = \frac{1/6+\varepsilon}{1+4\varepsilon}, \qquad c_\varepsilon = \frac{\varepsilon}{1+4\varepsilon}.$$

For a decentralized product policy $\pi_{p,q}$, the occupancy and return are

$$d_{p,q} = \big((1-p)(1-q),\ (1-p)q,\ p(1-q),\ pq\big), \qquad V(p,q) = (1-p)(1-q) + pq.$$

Let $D_z(p,q) := D_{\mathrm{KL}}(d_{p,q} \,\|\, d_{\mathcal{D}}^{z,\varepsilon})$ for $z \in \{0,1\}$. Since $\alpha = 1$ in this exact setting, the exact closest-slice objective is

$$J_{\varepsilon,\tau}^{\mathrm{CS}}(p,q) = V(p,q) + \tau \log\left(\frac{\exp(-D_0(p,q)/\tau) + \exp(-D_1(p,q)/\tau)}{2}\right).$$

Let

$$\Lambda_\varepsilon := \log \frac{a_\varepsilon}{c_\varepsilon}.$$

Then

$$D_1(p,q) - D_0(p,q) = (1-p-q)\Lambda_\varepsilon.$$

So along any fixed sum line $p+q=s$, the slice disparity gap, and therefore the selector weights, depend only on s, while the soft min value can still vary through the common shift in D_0 and D_1.

Lemma (global maximizers are symmetric). Write

$$m := \frac{p+q}{2}, \qquad \delta := \frac{p-q}{2}, \qquad \kappa_\varepsilon := 1 + \log \frac{\sqrt{a_\varepsilon c_\varepsilon}}{b_\varepsilon},$$

and let $h(u) := -u\log u - (1-u)\log(1-u)$. Along any fixed sum line with $m \in [0,1]$ and $|\delta| \le \min\{m, 1-m\}$,

$$J_{\varepsilon,\tau}^{\mathrm{CS}}(m+\delta, m-\delta) - J_{\varepsilon,\tau}^{\mathrm{CS}}(m,m) = h(m+\delta) + h(m-\delta) - 2h(m) - 2\kappa_\varepsilon\delta^2.$$

For these parameters $\varepsilon = 10^{-3}$,

$$\kappa_\varepsilon = 1 + \log \frac{\sqrt{a_\varepsilon c_\varepsilon}}{b_\varepsilon} \approx -0.870 > -2.$$

Since

$$h''(u) = -\frac{1}{u} - \frac{1}{1-u} \le -4 \qquad \text{for } u \in (0,1),$$

Taylor's theorem gives, for interior points with $0 < m-\delta < m+\delta < 1$,

$$h(m+\delta) + h(m-\delta) - 2h(m) \le -4\delta^2.$$

Both sides are continuous on the closed feasible set

$$\{(m,\delta) : m \in [0,1],\ |\delta| \le \min\{m, 1-m\}\},$$

so the same inequality extends to boundary points by continuity. Hence

$$J_{\varepsilon,\tau}^{\mathrm{CS}}(m+\delta, m-\delta) - J_{\varepsilon,\tau}^{\mathrm{CS}}(m,m) \le -2(2+\kappa_\varepsilon)\delta^2 < 0 \qquad (\delta \neq 0).$$

Therefore every global maximizer of the exact closest-slice objective over decentralized product policies is symmetric: $p = q = t$.

Restrict to symmetric decentralized product policies $\pi_{t,t}$, whose occupancy is

$$d_t = \big((1-t)^2,\ t(1-t),\ t(1-t),\ t^2\big)$$

and whose return is

$$V(t) = 1 - 2t(1-t).$$

Because the witness and the two slices are symmetric under $t \mapsto 1-t$, it suffices to optimize on $t \in [0, \frac{1}{2}]$.

On $t \in [0, \frac{1}{2}]$, the slice disparity gap is

$$D_1(t) - D_0(t) = (1-2t)\Lambda_\varepsilon,$$

where

$$D_0(t) = (1-t)^2 \log \frac{(1-t)^2}{a_\varepsilon} + 2t(1-t) \log \frac{t(1-t)}{b_\varepsilon} + t^2 \log \frac{t^2}{c_\varepsilon}.$$

Therefore the exact closest-slice objective on this symmetric slice is the one dimensional function

$$J^{\mathrm{CS}}_{\varepsilon,\tau}(t) = V(t) - D_0(t) + \tau \log\left(\frac{1+\exp(-(1-2t)\Lambda_\varepsilon/\tau)}{2}\right).$$

Differentiating gives

$$\frac{1}{2}\frac{d}{dt}J^{\mathrm{CS}}_{\varepsilon,\tau}(t) = 2t - 1 + \log\frac{1-t}{t} + \log\frac{b_\varepsilon}{a_\varepsilon} + t\log\frac{a_\varepsilon c_\varepsilon}{b_\varepsilon^2} + \frac{\Lambda_\varepsilon}{1+\exp((1-2t)\Lambda_\varepsilon/\tau)}.$$

A high precision one dimensional scan of $\frac{d}{dt}J^{\mathrm{CS}}_{\varepsilon,\tau}(t)$ finds a unique sign change on $(0, \frac{1}{2})$ between $t = 0.0749$ and $t = 0.0750$. A subsequent root solve gives the corresponding interior critical point

$$t_\star \approx 0.074998883,$$

and symmetry gives the matching point $1 - t_\star \approx 0.925001117$. By symmetry, $J'^{\mathrm{CS}}_{\varepsilon,\tau}(\frac{1}{2}) = 0$ as well. The corresponding objective values are

$$J^{\mathrm{CS}}_{\varepsilon,\tau}(t_\star) \approx 0.723093, \qquad J^{\mathrm{CS}}_{\varepsilon,\tau}(0) \approx 0.557384, \qquad J^{\mathrm{CS}}_{\varepsilon,\tau}\big(\tfrac{1}{2}\big) \approx -0.838517.$$

Together with the symmetry lemma above, these one-dimensional calculations provide numerical evidence that the candidate maximizers over decentralized product policies are $t_\star$ and $1 - t_\star$, with $t = \frac{1}{2}$ and the boundary point $t = 0$ both suboptimal. The resulting return is

$$V(t_\star) = 1 - 2t_\star(1 - t_\star) \approx 0.861251899.$$

We use this numerical candidate as the sensitivity benchmark; global optimality remains unproved.

A.8 Proof of the boundary result post-reveal (single agent analogue)

Boundary post-reveal under pooled reverse KL. Assume $\alpha > 0$. Consider the two-step single agent witness with common first observation $o_\perp$, branch revealing second observations o_0, o_1, and terminal reward

$$r = \mathbf{1}\{a_2 = a_1\}.$$

Let slice 0 generate trajectory 00 with weight $1 - \rho$ and slice 1 generate trajectory 11 with weight ρ, where $\rho \in (0, 1)$. Smooth the branch conditional second step occupancies to full support with probabilities $(1 - \varepsilon, \varepsilon)$ for $\varepsilon \in (0, \frac{1}{2})$. For the policy class

$$\pi_{p,c_0,c_1} : \qquad p = \pi(1 \mid o_\perp), \qquad c_0 = \pi(0 \mid o_0), \qquad c_1 = \pi(1 \mid o_1),$$

the pooled reverse KL objective

$$J^{\mathrm{fork}}_{\alpha,\varepsilon}(p, c_0, c_1) = (1-p)c_0 + pc_1 - \alpha\,\mathrm{kl}(p \,\|\, \rho) - \frac{\alpha}{2}(1-p)\,\mathrm{kl}(c_0 \,\|\, 1-\varepsilon) - \frac{\alpha}{2}p\,\mathrm{kl}(c_1 \,\|\, 1-\varepsilon)$$

has unique maximizer

$$p^\star = \rho, \qquad c_0^\star = c_1^\star = c_\varepsilon^\star := \left(1 + e^{-2/\alpha}\frac{\varepsilon}{1-\varepsilon}\right)^{-1}.$$

The resulting return is $V^\star = c_\varepsilon^\star$, so $V^\star \to 1$ as $\varepsilon \downarrow 0$. In the exact support case, the pooled optimum attains return 1.

Proof. Order the normalized occupancy coordinates as

$$(o_\perp, 0),\ (o_\perp, 1),\ (o_0, \mathrm{corr}),\ (o_0, \mathrm{err}),\ (o_1, \mathrm{err}),\ (o_1, \mathrm{corr}).$$

Under π_{p,c_0,c_1},

$$d_\pi = \tfrac{1}{2}\big((1-p),\ p,\ (1-p)c_0,\ (1-p)(1-c_0),\ p(1-c_1),\ pc_1\big).$$

The pooled dataset occupancy is

$$d_{\mathcal{D}} = \tfrac{1}{2}\big((1-\rho),\ \rho,\ (1-\rho)(1-\varepsilon),\ (1-\rho)\varepsilon,\ \rho\varepsilon,\ \rho(1-\varepsilon)\big).$$

Expand the reverse KL term coordinate wise:

$$\begin{aligned}\mathrm{KL}(d_\pi \,\|\, d_{\mathcal{D}}) = &\frac{1-p}{2}\log\frac{1-p}{1-\rho} + \frac{p}{2}\log\frac{p}{\rho} \\ &+ \frac{(1-p)c_0}{2}\log\frac{(1-p)c_0}{(1-\rho)(1-\varepsilon)} + \frac{(1-p)(1-c_0)}{2}\log\frac{(1-p)(1-c_0)}{(1-\rho)\varepsilon} \\ &+ \frac{p(1-c_1)}{2}\log\frac{p(1-c_1)}{\rho\varepsilon} + \frac{pc_1}{2}\log\frac{pc_1}{\rho(1-\varepsilon)}.\end{aligned}$$

Grouping the branch 0 and branch 1 terms gives

$$\mathrm{KL}(d_\pi \,\|\, d_{\mathcal{D}}) = \mathrm{kl}(p \,\|\, \rho) + \frac{1-p}{2}\,\mathrm{kl}(c_0 \,\|\, 1-\varepsilon) + \frac{p}{2}\,\mathrm{kl}(c_1 \,\|\, 1-\varepsilon).$$

Because only correct second step completion is rewarded, the terminal reward is 1, and the equal step return convention multiplies occupancy rewards by the horizon $H = 2$,

$$V(d_\pi) = (1-p)c_0 + pc_1.$$

This yields the displayed objective.

For fixed p, the objective separates as

$$J^{\mathrm{fork}}_{\alpha,\varepsilon}(p, c_0, c_1) = -\alpha\,\mathrm{kl}(p \,\|\, \rho) + (1-p)\phi(c_0) + p\,\phi(c_1),$$

where

$$\phi(c) := c - \frac{\alpha}{2}\,\mathrm{kl}(c \,\|\, 1-\varepsilon).$$

The inner objective ϕ has second derivative

$$\phi''(c) = -\frac{\alpha}{2c(1-c)} < 0,$$

so ϕ is strictly concave on $(0, 1)$. The first order condition

$$\phi'(c) = 1 - \frac{\alpha}{2}\log\frac{c\varepsilon}{(1-c)(1-\varepsilon)} = 0$$

implies

$$\frac{c}{1-c} = e^{2/\alpha}\frac{1-\varepsilon}{\varepsilon},$$

hence

$$c = c^\star_\varepsilon = \left(1 + e^{-2/\alpha}\frac{\varepsilon}{1-\varepsilon}\right)^{-1}.$$

If $p \in (0, 1)$, both coefficients $1-p$ and p are positive, so the two active blocks are optimized independently by the same unique maximizer $c^\star_\varepsilon$. If $p = 0$, only the c_0 block is active and it is uniquely maximized at $c^\star_0 = c^\star_\varepsilon$, while c_1 is irrelevant. The case $p = 1$ is symmetric. In particular, for every fixed $p \in [0, 1]$, there exists an optimal pair with $c_0 = c_1 = c^\star_\varepsilon$.

Substituting back gives

$$J^{\mathrm{fork}}_{\alpha,\varepsilon}(p, c^\star_\varepsilon, c^\star_\varepsilon) = (1-p)c^\star_\varepsilon + pc^\star_\varepsilon - \alpha\,\mathrm{kl}(p \,\|\, \rho) - \frac{\alpha}{2}\,\mathrm{kl}(c^\star_\varepsilon \,\|\, 1-\varepsilon) = \phi(c^\star_\varepsilon) - \alpha\,\mathrm{kl}(p \,\|\, \rho).$$

Since $\mathrm{kl}(p \,\|\, \rho) \geq 0$ with equality only at $p = \rho$, the unique maximizer in p is $p^\star = \rho$. Because $\rho \in (0, 1)$, both c-blocks are active at that maximizer, and the strict concavity of ϕ gives the unique choices $c^\star_0 = c^\star_1 = c^\star_\varepsilon$. The resulting return is

$$V^\star = (1-p^\star)c^\star_\varepsilon + p^\star c^\star_\varepsilon = c^\star_\varepsilon.$$

Finally,

$$\lim_{\varepsilon \downarrow 0} c^\star_\varepsilon = 1.$$

In the exact support case $\varepsilon = 0$, the extended value reverse KL term is finite only if no probability mass is placed on an error coordinate. Thus, for $p \in (0, 1)$, finiteness forces $c_0 = c_1 = 1$, while at the boundary $p = 0$ only $c_0 = 1$ is required and c_1 is irrelevant, and symmetrically at $p = 1$. For interior p, the objective reduces to

$$J^{\mathrm{fork}}_{\alpha,0}(p, 1, 1) = 1 - \alpha\,\mathrm{kl}(p \,\|\, \rho),$$

which is uniquely maximized at $p = \rho$. The boundary finite points have objective values $1 - \alpha\,\mathrm{kl}(0 \,\|\, \rho)$ and $1 - \alpha\,\mathrm{kl}(1 \,\|\, \rho)$, both strictly below 1 because $\rho \in (0, 1)$. Hence the exact support pooled optimum attains return 1 exactly, at the interior policy $p = \rho,\ c_0 = c_1 = 1$. □

A.9 Weighted local marginal projection

Weighted BC is a projection onto local marginals. Assume each local action set A_i is finite. Let $w(o, a, r, o', \texttt{done}) \geq 0$ be arbitrary tuple weights on the pooled dataset, and consider

$$\max_{\pi \in \Pi^{\text{cb}}_{\text{prod}}} \mathbb{E}_{(o,a,r,o',\texttt{done})\sim\mathcal{D}} \left[w(o, a, r, o', \texttt{done}) \, \log \pi(a \mid o) \right].$$

For each agent i, define the weighted local counts

$$C_i^w(\tilde{o}_i, \tilde{a}_i) := \mathbb{E}_{(o,a,r,o',\texttt{done})\sim\mathcal{D}} \Big[w(o, a, r, o', \texttt{done}) \, \mathbf{1}\{o_i = \tilde{o}_i, \; a_i = \tilde{a}_i\} \Big].$$

Then for every local observation $\tilde{o}_i$ with

$$C_i^w(\tilde{o}_i) := \sum_{\tilde{a}_i} C_i^w(\tilde{o}_i, \tilde{a}_i) > 0,$$

every maximizer satisfies

$$\pi_i^\star(\tilde{a}_i \mid \tilde{o}_i) = \frac{C_i^w(\tilde{o}_i, \tilde{a}_i)}{C_i^w(\tilde{o}_i)}.$$

If $C_i^w(\tilde{o}_i) = 0$, the objective is indifferent over $\pi_i(\cdot \mid \tilde{o}_i)$.

Proof. Because $\pi(a \mid o) = \prod_i \pi_i(a_i \mid o_i)$, we have

$$\log \pi(a \mid o) = \sum_i \log \pi_i(a_i \mid o_i).$$

Substituting into the weighted BC objective gives

$$\sum_i \mathbb{E}_{(o,a,r,o',\texttt{done})\sim\mathcal{D}} \left[w(o, a, r, o', \texttt{done}) \, \log \pi_i(a_i \mid o_i) \right].$$

Reindexing by local observations and actions yields

$$\sum_i \sum_{\tilde{o}_i, \tilde{a}_i} C_i^w(\tilde{o}_i, \tilde{a}_i) \, \log \pi_i(\tilde{a}_i \mid \tilde{o}_i).$$

This separates across agent observation blocks. For a fixed pair $(i, \tilde{o}_i)$, write

$$c_{\tilde{a}_i} := C_i^w(\tilde{o}_i, \tilde{a}_i), \qquad c := \sum_{\tilde{a}_i} c_{\tilde{a}_i}.$$

The corresponding optimization problem is

$$\max_{p \in \Delta(A_i)} \sum_{\tilde{a}_i} c_{\tilde{a}_i} \, \log p(\tilde{a}_i).$$

If $c = 0$, the block contributes 0 for every p, so the objective is indifferent. Suppose now that $c > 0$. This objective is strictly concave on the simplex restricted to the support of $c_{\tilde{a}_i}$. Introducing a Lagrange multiplier λ for the normalization constraint $\sum_{\tilde{a}_i} p(\tilde{a}_i) = 1$, the first order conditions are

$$\frac{c_{\tilde{a}_i}}{p(\tilde{a}_i)} = \lambda \qquad \text{for every } \tilde{a}_i \text{ with } c_{\tilde{a}_i} > 0.$$

Any action with $c_{\tilde{a}_i} = 0$ must be assigned probability 0 at an optimum with $c > 0$, because moving mass from a positive count action to a zero count action strictly decreases the objective. Hence $p(\tilde{a}_i) = c_{\tilde{a}_i}/\lambda$, and summing over $\tilde{a}_i$ gives $\lambda = c$. Therefore the unique maximizer on this block is

$$\pi_i^\star(\tilde{a}_i \mid \tilde{o}_i) = \frac{c_{\tilde{a}_i}}{c} = \frac{C_i^w(\tilde{o}_i, \tilde{a}_i)}{C_i^w(\tilde{o}_i)}.$$

Combining the independently optimized blocks proves the proposition. □

A.10 n agent extension

Scaling with the number of agents. For demonstrations supported on 0^n and 1^n with weights $C_0, C_1 > 0$, weighted local extraction gives each agent probability $\xi = C_1/(C_0 + C_1)$ of action 1. The product policy therefore has return

$$V(\pi) = (1-\xi)^n + \xi^n.$$

Balanced demonstrations give $\xi = 1/2$ and return 2^{1-n}, the combined probability of the two successful joint actions.

A.11 Proof of Corollary 1

Corollary 1 (exact slice CS-DICE recovery on the aliased single agent witness). Consider the exact slice occupancies

$$d^0_{\mathcal{D}} = \Big(\frac{1}{2}, 0, \frac{1}{2}, 0, 0, 0\Big), \qquad d^1_{\mathcal{D}} = \Big(0, \frac{1}{2}, 0, 0, 0, \frac{1}{2}\Big),$$

with uniform slice prior, on coordinates

$$((s_0,0),(s_0,1),(s_L,0),(s_L,1),(s_R,0),(s_R,1)).$$

Under Assumption 1 with $f_\infty = \infty$, for any $\alpha > 0$ and $\tau > 0$, the only maximizers of

$$J^{\mathrm{CS}}_{\alpha,f}(\pi) = V(\pi) - \alpha\, \Psi_{\tau,f}\big(d^\pi; \{d^0_{\mathcal{D}}, d^1_{\mathcal{D}}\}\big)$$

over Π_{react} are the two coherent endpoints $\pi_{0,0}$ and $\pi_{1,1}$. Each attains objective value $1 - \alpha\tau \log 2$.

Proof. Write the reactive policy as $\pi = \pi_{p,r}$, where $p = \pi(1 \mid o_0)$ and $r = \pi(1 \mid \bar{o})$. Its occupancy is

$$d^\pi = \Big(\frac{1-p}{2}, \frac{p}{2}, \frac{(1-p)(1-r)}{2}, \frac{(1-p)r}{2}, \frac{p(1-r)}{2}, \frac{pr}{2}\Big).$$

Let $d = d^\pi$. Since the slice prior is uniform,

$$\Psi_{\tau,f}(d) = -\tau \log\left(\frac{1}{2} e^{-D_f(d \,\|\, d^0_{\mathcal{D}})/\tau} + \frac{1}{2} e^{-D_f(d \,\|\, d^1_{\mathcal{D}})/\tau}\right).$$

We first record the only support mismatch implication needed below. For an f divergence written in density ratio form,

$$D_f(p \,\|\, q) = \sum_x q(x)\, f\left(\frac{p(x)}{q(x)}\right),$$

the standard support mismatch extension on a coordinate with $q(x) = 0$ and $p(x) > 0$ is

$$\lim_{r \to \infty} \frac{p(x)}{r}\, f(r) = p(x)\, f_\infty.$$

Under $f_\infty = \infty$, any positive mass outside the support of q therefore contributes $+\infty$.

Applying this to the exact slice $d^0_{\mathcal{D}}$, finiteness of

$$D_f(d \,\|\, d^0_{\mathcal{D}})$$

implies

$$\mathrm{supp}(d) \subseteq \{(s_0,0),(s_L,0)\}.$$

From the displayed occupancy formula, that support condition forces $p = 0$ and then $r = 0$, so any policy other than $\pi_{0,0}$ has

$$D_f(d \,\|\, d^0_{\mathcal{D}}) = +\infty.$$

Similarly, finiteness of

$$D_f(d \,\|\, d^1_{\mathcal{D}})$$

implies

$$\mathrm{supp}(d) \subseteq \{(s_0,1),(s_R,1)\},$$

which forces $p = 1$ and then $r = 1$. Therefore any policy other than $\pi_{1,1}$ has

$$D_f(d \,\|\, d^1_{\mathcal{D}}) = +\infty.$$

Every reactive policy other than $\pi_{0,0}$ or $\pi_{1,1}$ is therefore assigned infinite divergence by both slices. Hence any policy other than $\pi_{0,0}$ or $\pi_{1,1}$ has

$$\Psi_{\tau,f}(d) = \infty,$$

so its objective is $-\infty$ and it cannot be optimal.

For $d = d^0_{\mathcal{D}}$, Assumption 1 gives $f(1) = 0$, so

$$\Psi_{\tau,f}(d^0_{\mathcal{D}}) = -\tau \log\left(\frac{1}{2}e^0 + \frac{1}{2}e^{-\infty}\right) = \tau \log 2.$$

The return is $V(d^0_{\mathcal{D}}) = 1$, hence

$$J^{\mathrm{CS}}_{\alpha,f}(d^0_{\mathcal{D}}) = 1 - \alpha\tau \log 2.$$

The same calculation holds for $d = d^1_{\mathcal{D}}$. Thus the only maximizers are the two coherent endpoints, and they tie. □

A.12 Full proof of Theorem 2

Theorem 2 (two agent coordination witness: reverse KL midpoint optimizer). Consider the one-step two agent coordination task with actions in $\{0, 1\}$ and fixed shared reward

$$r(a_1, a_2) = \mathbf{1}\{a_1 = a_2\}.$$

Let the pooled dataset occupancy on action order $(00, 01, 10, 11)$ be

$$d_{\mathcal{D}} = (a, b, b, a), \qquad a > b > 0, \qquad a + b = \tfrac{1}{2}.$$

Over decentralized product policies $\pi_{p,q}$, optimize

$$J^{\mathrm{pool}}_{\alpha}(\pi) = V(\pi) - \alpha\,\mathrm{KL}(d_\pi \,\|\, d_{\mathcal{D}}).$$

Define

$$B_\alpha := 1 + \alpha \log(a/b).$$

If

$$-2\alpha < B_\alpha < 2\alpha,$$

then the unique global maximizer is the midpoint policy

$$p = q = \tfrac{1}{2}.$$

Its return is $\frac{1}{2}$, while each coherent endpoint $(0, 0)$ and $(1, 1)$ attains return 1.

Proof. Write the decentralized product policy as $\pi_{p,q}$, where agent 1 chooses action 1 with probability p and agent 2 with probability q. Its joint occupancy is

$$d_{\pi_{p,q}} = \big((1-p)(1-q),\ (1-p)q,\ p(1-q),\ pq\big).$$

Define

$$E(p, q) := (1-p)(1-q) + pq.$$

Because the reward is 1 on coordinated actions and 0 otherwise,

$$V(\pi_{p,q}) = E(p, q).$$

Since $d_{\pi_{p,q}}$ factorizes,

$$H(d_{\pi_{p,q}}) = h(p) + h(q), \qquad h(u) := -u \log u - (1-u)\log(1-u).$$

The pooled dataset takes value a on the diagonal and b on the off diagonals, so

$$\mathbb{E}_{d_{\pi_{p,q}}}[\log d_{\mathcal{D}}] = E(p, q) \log a + (1 - E(p, q)) \log b.$$

Therefore

$$\mathrm{KL}(d_{\pi_{p,q}} \,\|\, d_{\mathcal{D}}) = -H(d_{\pi_{p,q}}) - \mathbb{E}_{d_{\pi_{p,q}}}[\log d_{\mathcal{D}}],$$

and the objective becomes

$$J_\alpha^{\mathrm{pool}}(p,q) = \alpha(h(p) + h(q)) + B_\alpha E(p,q) + \alpha \log b,$$

where

$$B_\alpha = 1 + \alpha \log(a/b).$$

Now introduce

$$m := \frac{p+q}{2}, \qquad \delta := \frac{p-q}{2}.$$

Then

$$E(m+\delta, m-\delta) = (1-m)^2 + m^2 - 2\delta^2.$$

Hence

$$J_\alpha^{\mathrm{pool}}(m+\delta, m-\delta) - J_\alpha^{\mathrm{pool}}(m,m) = \alpha\big(h(m+\delta) + h(m-\delta) - 2h(m)\big) - 2B_\alpha\delta^2.$$

Since

$$h''(u) = -\frac{1}{u(1-u)} \le -4 \qquad \text{for } u \in (0,1),$$

strong concavity gives, for interior points with $0 < m - \delta < m + \delta < 1$,

$$h(m+\delta) + h(m-\delta) - 2h(m) \le -4\delta^2.$$

Both sides are continuous on the closed feasible set

$$\{(m,\delta) : m \in [0,1],\ |\delta| \le \min\{m, 1-m\}\},$$

so the same inequality extends to boundary points by continuity. Therefore

$$J_\alpha^{\mathrm{pool}}(m+\delta, m-\delta) - J_\alpha^{\mathrm{pool}}(m,m) \le -2(2\alpha + B_\alpha)\delta^2.$$

If $B_\alpha > -2\alpha$, the right side is strictly negative whenever $\delta \neq 0$. Thus every maximizer is symmetric, so $p = q$.

Restrict now to the symmetric slice and define

$$g(p) := J_\alpha^{\mathrm{pool}}(p,p) = 2\alpha h(p) + B_\alpha\big((1-p)^2 + p^2\big) + \alpha \log b.$$

Differentiate:

$$g'(p) = 2\alpha \log \frac{1-p}{p} + B_\alpha(4p-2),$$

$$g''(p) = -\frac{2\alpha}{p(1-p)} + 4B_\alpha.$$

Because

$$-\frac{2\alpha}{p(1-p)} \le -8\alpha,$$

we obtain

$$g''(p) \le -8\alpha + 4B_\alpha < 0$$

whenever $B_\alpha < 2\alpha$. So g is strictly concave on $(0,1)$. By symmetry,

$$g'\big(\tfrac{1}{2}\big) = 0,$$

hence $p = \frac{1}{2}$ is the unique interior critical point on the symmetric slice. Moreover,

$$\lim_{p \downarrow 0} g'(p) = +\infty, \qquad \lim_{p \uparrow 1} g'(p) = -\infty,$$

so the symmetric maximizer is interior and therefore the unique global maximizer on $[0,1]$. Since every maximizer is symmetric, this is also the unique global maximizer over $[0,1]^2$.

Finally, the midpoint policy satisfies

$$V(\pi_{1/2,1/2}) = \frac{1}{2},$$

while the coherent endpoints $\pi_{0,0}$ and $\pi_{1,1}$ each satisfy

$$V(\pi_{0,0}) = V(\pi_{1,1}) = 1.$$

Moreover, the midpoint places positive mass on both off diagonal actions, so it mixes the two conventions instead of committing to either endpoint. This proves the theorem. □

A.13 Smoothed support selector bound

Finite-smoothing separation. Consider the cooperative witness with slices $d_0^\varepsilon = (1-3\varepsilon, \varepsilon, \varepsilon, \varepsilon)$ and $d_1^\varepsilon = (\varepsilon, \varepsilon, \varepsilon, 1-3\varepsilon)$, where $0 < \varepsilon < 1/4$. Let $\Psi_{\tau,\mathrm{rKL}}$ use a uniform slice prior and define

$$\eta(d) = d(01) + d(10), \qquad B_\varepsilon(t) = t \log \frac{t}{2\varepsilon} - t + 2\varepsilon,$$

$$L_{\varepsilon,\tau} = \tau \log 2 + \log \frac{1}{1-3\varepsilon}.$$

The soft closest-slice penalty obeys

$$\Psi_{\tau,\mathrm{rKL}}(d) \geq B_\varepsilon(\eta(d)),$$
$$\max_{a \in \{00,11\}} \Psi_{\tau,\mathrm{rKL}}(\delta_a) \leq L_{\varepsilon,\tau}.$$

For $\alpha, \tau > 0$ and $\bar{\eta} \geq 2\varepsilon$, the condition

$$\bar{\eta} + \alpha B_\varepsilon(\bar{\eta}) > \alpha L_{\varepsilon,\tau}$$

places every product policy with $\eta(d_\pi) \geq \bar{\eta}$ strictly below both coherent endpoints in CS objective. For fixed $\bar{\eta} > 0$, $\alpha > 0$, and $\tau > 0$, it holds for all sufficiently small $\varepsilon > 0$. *Proof.* Jensen's inequality on the two off-diagonal coordinates bounds each slice divergence below by $B_\varepsilon(\eta(d))$. The soft-min bound transfers this to Ψ, and retaining the matching-slice term bounds each endpoint penalty by $L_{\varepsilon,\tau}$. Since B_ε is increasing on $[2\varepsilon, 1]$ and $V = 1 - \eta$, comparison with an endpoint gives the displayed condition. As $\varepsilon \downarrow 0$, $B_\varepsilon(\bar{\eta}) \to \infty$ and $L_{\varepsilon,\tau} \to \tau \log 2$.

A.14 Implementation details

The controlled synthetic mechanism experiments use seeds $\{0, 1, 2, 3, 4\}$ and the configuration files `experiments/configs/sc_comadice_final.yaml` and `experiments/configs/sc_comadice_final_temporal.yaml`. All numbers in the main controlled tables are aggregated across the full seed list. Separate-slice rows use oracle or inferred partitions, as indicated. Within each seed we train one actor for each slice partition, evaluate those slice policies separately, average their returns within the seed, and then take the across seed mean and standard deviation. The main definitions are stated for occupancies induced by the deployed policy class. The controlled implementations below optimize finite enumerated or tuple weighting relaxations of those subproblems where noted.

For the cooperative pooled / CS / separate-slice rows in the main controlled tables, the reported values come from an enumerated reverse KL tuple weighting pipeline on pooled tuple support followed by decentralized product weighted BC extraction. The controlled implementation enumerates tuple weights and extracts product actors. The MaMuJoCo integration study below is the mixer implementation. Weighted BC controls remain separate behavior cloning baselines. Appendix A.16 evaluates the cooperative oracle rows with a finite horizon reverse KL direct policy objective comparison that aligns the regularization weight with the main controlled suite.

Occupancies are normalized to mass one. For an equal-step horizon-H witness, $V(d) := H \sum_\xi d(\xi)\, \bar{r}(\xi)$, so a terminal success reward of one yields return one without rescaling the environment reward.

Reward-free representations. The discrete summary concatenates mean and initial observations, a normalized joint-action histogram, and episode length. MaMuJoCo uses the analogous observation summaries, action means, standard deviations, root-mean-square magnitudes, and length. Rewards are excluded from clustering. The temporal encoder receives observation/action/next-observation/termination sequences, and minimizes masked next-observation squared error plus termination binary cross-entropy. We cluster final GRU hidden states with k-means for the main temporal rows; spectral clustering is confined to the separate K-slice diagnostic. The learned representations define candidate partitions whose semantic interpretation requires separate validation.

Table 4: Controlled suite and appendix protocol settings used throughout the empirical section.

Setting	Value
Dataset generation	Exact synthetic benchmarks instantiated from code, with no external simulator or pretrained model
Canonical cooperative seeds	$\{0, 1, 2, 3, 4\}$, paired across methods
Main controlled table mapping	Oracle rows use oracle training slices. Inferred one-step and ConventionRepeat-HH summary rows use summary k-means without reward labels. The Horizon-2 temporal row uses temporal GRU embeddings without reward labels before k-means. The cooperative row post-reveal is a boundary control with no inferred slice entry.
Cooperative main controlled execution path	Enumerated reverse KL tuple weighting learner on pooled tuple support followed by decentralized product weighted BC actor extraction for the cooperative pooled / CS / separate-slice rows. BC controls remain separate weighted BC baselines
Single agent controls	Produced by the standalone exact benchmark script with the same seed list and reported in the main controlled table as seed means and standard deviations
Policy hidden dimension	8
Optimizer	Adam, learning rate 0.05 (0.01 for the cooperative post-reveal control)
Selector prior	empirical slice frequency prior $u = \rho$ in the main controlled suite
Selector temperature / refresh	$\tau = 0.05$, $K_{\text{refresh}} = 1$; cooperative post-reveal: $\tau = 0.2$, $K_{\text{refresh}} = 25$
CS initialization warm start	Before the first selector refresh only, multiply the initial occupancy proxy by $\exp(10^{-3}\xi)$ with $\xi \sim \mathcal{N}(0, 1)$. If the refreshed selector still has $\max_z q(z) < 0.55$, sample one slice from the prior for that first reference and then continue with the ordinary refresh rule
Tuple weighting DICE regularization / discount	$\alpha = 0.25$, $\gamma = 0.0$ for the cooperative main controlled tuple weighting runs
Matched-α direct objective weight	Same α as the cooperative tuple weighting suite (canonical default 0.25)
Gradient steps	80 (one-step, Horizon-2), 200 (ConventionRepeat-HH), 1000 (cooperative boundary control post-reveal), with a 2000-step budget for the matched α cooperative comparison post-reveal
Smoothing ε	10^{-3} (one-step, Horizon-2, cooperative boundary control post-reveal), 10^{-4} (ConventionRepeat-HH)
Default inferred slices	summary k-means on episode summaries without reward labels
Temporal main paper supplement	GRU embeddings without reward labels before k-means for the Horizon-2 inferred row; direct-policy ConventionRepeat-HH settings are specified separately below
Temporal label noise pipeline	GRU + spectral clustering without reward labels for the K-slice ConventionRepeat-HH label noise study
Temporal slice encoder	hidden dim 64, 1 layer, 50 epochs, batch size 32, Adam at 10^{-3}
Inferred episodes per slice	32 in the main suite and 64 in the label noise and K slice appendix studies
Checkpoint protocol	fixed final checkpoints selected before evaluation
Aggregation rule	cooperative rows report mean $\pm$ std across paired seeds, while separate-slice first averages slice policy returns within seed
Compute	measured on Linux 6.8 / Python 3.13.12 / Intel i5-13450HX CPU workstation, with the full CPU suite running in under 1 hour and temporal inference adding ≈ 0.15s per Horizon-2 cell

The reproduction notes in `docs/reproduce/REPRODUCE.md` record the command checklist, environment setup, and expected artifacts. The main inferred Horizon-2 row comes from the temporal supplement.

The controlled suite parameters were fixed before the final table runs. Under the additive smoothing rule used in the code, any $\varepsilon > 0$ gives full support to every slice occupancy, so the listed values are stability choices for each benchmark. The main controlled suite uses 10^{-3} on the one-step, Horizon-2, and cooperative boundary controls post-reveal and 10^{-4} on ConventionRepeat-HH. For the selector,

$\tau = 0.05$ stays on the same broad performance plateau as $[0.005, 0.5]$ in the one-step sweep below. The cooperative boundary control post-reveal uses a slightly softer $\tau = 0.2$ and a lower learning rate because that post-reveal control is most stable when the pooled reference is already nearly optimal and the selector is not forced into premature specialization. Large temperatures move the selector toward its prior, giving the prior-weighted geometric reference. On the exactly balanced witnesses, the tiny warm start jitter and first step prior draw above break initialization symmetry. After that first refresh the implementation reverts to the ordinary selector rule.

A.15 MaMuJoCo continuous-control protocol and results

Tasks and protocol. The study uses Hopper-v2 3×1, Ant-v2 2×4, and HalfCheetah-v2 6×1, each with expert, medium, medium-replay, and medium-expert datasets. Both methods use the same CoMA-DICE network stack and runner with reverse KL; the comparison changes pooled versus selected reference shaping. Each method uses seeds $0, 1, 2, 3, 4$ in all twelve cells, for 120 runs. Training uses float64 arithmetic, Adam with learning rate 10^{-4}, $\alpha = 10$, batch size 128, and 25 complete epochs. CS uses $K = 2$ reward-free summary clusters, selector temperature 0.1, refresh interval 100 updates, selector batch size 4096, and assignment temperature 1. The hyperparameters are held fixed across the reported cells.

Evaluation and aggregation. Each run has 100 prespecified evaluation checkpoints, with 32 episodes per checkpoint. The primary score is the final checkpoint return. The best score is the maximum over those same 100 evaluations within each seed; we then compute its across-seed mean. Thus best-versus-best compares equal evaluation budgets, not a maximum of a mean learning curve. Tables 5 and 6 report mean $\pm$ sample standard deviation across five seeds, using the $n-1$ denominator. Standard deviations describe seed variation, not confidence intervals or standard errors. All runs are included regardless of return or recorded numerical recovery. Full training, evaluation schedules, finite scores, data identity, and checkpoint integrity are checked for every reported run.

For each cell, the relative score is the selected-reference mean divided by the pooled-reference mean, using the corresponding final or best score in both arms. Robot summaries average the four cell ratios equally; the overall summary averages all twelve. We do not average raw returns across robots. Higher-mean counts are descriptive, and no statistical significance is claimed.

Table 5: MaMuJoCo final returns, mean $\pm$ sample standard deviation across five seeds per method. Both methods use reverse KL. Each seed contributes its fixed-final score, averaged over 32 evaluation episodes.

Robot	Dataset	Pooled final	CS final
Hopper-v2	expert	1692.5 ± 1734.9	1719.1 ± 1684.2
Hopper-v2	medium	574.4 ± 321.7	79.7 ± 13.4
Hopper-v2	medium-replay	200.1 ± 107.4	124.9 ± 81.3
Hopper-v2	medium-expert	136.4 ± 87.7	140.6 ± 61.6
HalfCheetah-v2	expert	3050.9 ± 716.1	3194.2 ± 1057.3
HalfCheetah-v2	medium	2150.3 ± 402.8	1744.0 ± 140.6
HalfCheetah-v2	medium-replay	1738.8 ± 194.7	1522.6 ± 271.2
HalfCheetah-v2	medium-expert	2283.9 ± 863.2	1873.9 ± 753.7
Ant-v2	expert	1596.8 ± 422.0	1894.8 ± 236.8
Ant-v2	medium	973.4 ± 130.4	1109.7 ± 84.6
Ant-v2	medium-replay	947.8 ± 20.9	984.3 ± 151.2
Ant-v2	medium-expert	1957.8 ± 104.7	1850.9 ± 381.5

Final returns. The mean selected/pooled ratio is 0.889, with robot-level means 0.702 for Hopper, 0.889 for HalfCheetah, and 1.078 for Ant. Selected means exceed pooled means in six of twelve cells. Ant expert and medium have the largest relative gains, while Hopper medium has the largest relative decline. Small positive differences on Hopper expert and medium-expert coexist with substantial seed variation. These results do not establish a uniform benchmark improvement. Without annotated conventions or a coherence certificate for the inferred clusters, they also do not identify reference incoherence as the cause of a neural result.

Table 6: MaMuJoCo best returns, mean $\pm$ sample standard deviation across five seeds per method. Both methods use reverse KL. Each seed contributes its maximum among 100 evaluations; both methods have the same search budget. These hindsight maxima do not define offline checkpoint selection.

Robot	Dataset	Pooled best	CS best
Hopper-v2	expert	3565.6 ± 1.9	3565.5 ± 3.0
Hopper-v2	medium	2364.1 ± 56.7	1910.7 ± 405.1
Hopper-v2	medium-replay	1143.8 ± 538.5	989.1 ± 534.1
Hopper-v2	medium-expert	3560.1 ± 7.7	3544.7 ± 45.7
HalfCheetah-v2	expert	4660.1 ± 47.6	4631.1 ± 34.6
HalfCheetah-v2	medium	2895.4 ± 131.2	3041.8 ± 220.5
HalfCheetah-v2	medium-replay	3001.1 ± 40.5	3071.2 ± 181.9
HalfCheetah-v2	medium-expert	4554.4 ± 46.9	4625.9 ± 251.7
Ant-v2	expert	2069.9 ± 19.2	2080.8 ± 36.6
Ant-v2	medium	1425.7 ± 43.0	1437.3 ± 39.0
Ant-v2	medium-replay	1124.0 ± 151.1	1162.4 ± 145.8
Ant-v2	medium-expert	2096.1 ± 22.5	2100.3 ± 41.6

Peak returns and retention. The mean best-versus-best ratio is 0.983: 0.917 for Hopper, 1.021 for HalfCheetah, and 1.012 for Ant. Selected best means reach at least 95% of pooled best means in ten of twelve cells. The exceptions are Hopper medium and medium-replay, with ratios 0.808 and 0.865. Competitive observed peaks therefore coexist with mixed final performance. Both methods can degrade sharply: on Hopper medium-expert, pooled mean best/final returns are $3560.1/136.4$, and selected means are $3544.7/140.6$. In both arms, zero of five seeds finish within 90% of their own best return. These observations do not isolate a cause for the losses. These hindsight maxima selected with online evaluation can be optimistic and do not provide an offline checkpoint-selection rule; fixed-final scores remain primary.

Software, hardware, and reproducibility. The implementation uses Python 3.13.12, PyTorch 2.12.1, NumPy 2.5.4, Gymnasium 1.4.0, and MuJoCo 3.16.0. Of the 120 runs, 113 use Linux CUDA on A100 GPUs with up to 80 GiB memory, three use an RTX 4090 with 24 GiB, and four use native macOS ARM64 CPU execution on an M1 Pro with 16 GiB unified memory. The ARM runs are HalfCheetah medium-expert seeds 2 and 3 for both methods; the RTX 4090 runs are its selected-reference seed 1 and both seed-4 runs. The remaining runs use A100 GPUs. Platform and runtime provenance are retained per seed, and each run retains its computational backend. Evaluation uses one or two workers with exact return and training-state parity. These heterogeneous resources limit hardware-controlled timing comparisons.

The accompanying `paper/data/mamujoco/` records per-seed scores, configuration and data identities, and artifact checksums. `scripts/build_matched_mamujoco_tables.py` validates these inputs and regenerates the tables. Total project compute, including development runs, is not fully accounted for.

A.16 Tuple weighting and direct objective comparison

The main controlled table uses the tuple weighting pipeline described in Section 4. Table 7 compares that tuple weighting pipeline with direct product policy optimization at the same regularization weight. Each direct-objective row uses the same α as the tuple weighting suite, aligning the regularization strength across layers. The comparison therefore reflects both reference choice and the difference between direct product policy optimization and tuple weighting followed by weighted-BC extraction.

Table 7: Matched α tuple weighting versus direct objective comparison on the cooperative oracle witnesses before the reveal. Tuple weighting rows use the decentralized product weighted BC extraction pipeline from Table 1. Direct rows use the same oracle witnesses with the finite horizon product policy objective at the same α as the tuple weighting cooperative suite (canonical default 0.25). Entries are mean return over 5 paired seeds.

Benchmark	Pooled tuple weighting	CS tuple weighting	Separate-slice tuple weighting	Pooled direct (matched α)	CS direct (matched α)	Separate-slice direct (matched α)
Cooperative one-step	0.5000	0.9880	0.9851	0.9931	0.9995	0.9992
Horizon-2	0.1250	0.9881	0.9764	0.0000	0.9998	0.9998
ConventionRepeat-HH	0.0005	0.9220	0.8513	0.0005	1.0000	1.0000

The matched-α comparison sharpens the mechanism claim. Horizon-2 and ConventionRepeat-HH show the same broad reference choice pattern after aligning α: pooled direct optimization reaches low returns, while direct CS and direct separate-slice are essentially perfect. In Horizon-2, a successful endpoint has $J_{\mathrm{pool}} \approx 0.8257 > 0$, so the reported zero-return policy is not a global optimizer. Cooperative one-step highlights the role of the tuple weighting learner and weighted-BC extraction step. At the main cooperative weight $\alpha = 0.25$, the direct product policy pooled comparison already reaches 0.9931, while the enumerated one-step row remains at 0.500. The reproducibility archive also includes a matched-α direct rerun of the cooperative witness post-reveal, used as an optimization diagnostic alongside the main post-reveal control.

The tuple weighting main controlled table is the end-to-end algorithm result. Table 7 aligns α and shows where the direct product policy optimization matches the tuple weighting reference choice pattern.

A.17 Compact non-KL witness diagnostic

The exact midpoint depends on the f divergence. What persists is the pooled projection failure: the pooled f projection onto the deployed decentralized product class can still mix conventions.

We repeat the canonical full-support cooperative one-step witness with reverse KL, normalized Pearson χ^2 with generator $f(u) = \frac{1}{2}(u-1)^2$, and soft χ^2 with $f(t) = t\log t - t + 1$ for $0 \le t < 1$ and $f(t) = \frac{1}{2}(t-1)^2$ for $t \ge 1$ ($0 \log 0 = 0$), using the same pooled occupancy

$$d_{\mathcal{D}} = (1/3, 1/6, 1/6, 1/3)$$

and regularization weight $\alpha = 1$ as in the canonical witness computations. On the corresponding exact support witness, the pooled policies that mix conventions below would be excluded by infinite divergence. On this full-support witness, the pooled optimum keeps mixing conventions in all three cases. Reverse KL and soft χ^2 select the midpoint and obtain return 0.500. For this same witness and $\alpha = 1$, this normalized Pearson χ^2 selects a different symmetric product policy that mixes conventions at $p = q = 1/3$ or $p = q = 2/3$, and obtains return $5/9$. As a positive control, the exact-slice (zero-temperature closest-slice) proxy selects a coherent endpoint and obtains return 1.0 for all three penalties.

Thus the pooled failure persists beyond reverse KL, while the exact midpoint location is the reverse KL specialization. The relevant object for a general f divergence is the pooled f projection onto the deployed policy class, not the midpoint itself.

Table 8: Matched reference-only one-step controls reported in the study: the deployed class, data, divergence, and optimization budget are fixed within each row. Entries report numerical return estimates from finite optimization on these witnesses.

Divergence	Pooled reference	Coherent-slice reference
Reverse KL	0.5000	1.0000
Pearson χ^2	0.5561	1.0000
Soft χ^2	0.5000	1.0000

The Pearson numerical estimate is close to the analytic value $5/9$ above. These controls test reference choice in finite witnesses; Appendix A.15 reports the separate neural comparison.

A.18 Extended controlled benchmark results

Horizon-2. Horizon-2 is the appendix benchmark that most directly explains the slice inference paragraph in the main text: the two successful sequences share the same bag of features statistics, so summary clustering fails while temporal inference succeeds for the evaluated methods.

Table 9: Horizon-2 shared reward benchmark. The inferred rows report summary and temporal pipelines separately. Entries are mean ± standard deviation over 5 paired seeds.

Method	Return	Slice accuracy
Pooled BC	0.125 ± 0.000	—
Pooled reverse KL CoMA-DICE	0.125 ± 0.000	—
Closest-slice BC (oracle)	0.985 ± 0.002	—
Closest-slice BC (inferred, summary)	0.125 ± 0.000	0.500 ± 0.000
CS-CoMA-DICE (oracle)	$\mathbf{0.988 \pm 0.000}$	—
CS-CoMA-DICE (inferred, summary)	0.125 ± 0.000	0.500 ± 0.000
CS-CoMA-DICE (inferred, temporal)	$\mathbf{0.988 \pm 0.000}$	1.000 ± 0.000
Separate-slice CoMA-DICE (oracle)	0.976 ± 0.000	—
Separate-slice CoMA-DICE (inferred, summary)	0.125 ± 0.000	0.500 ± 0.000
Separate-slice CoMA-DICE (inferred, temporal)	0.976 ± 0.000	1.000 ± 0.000

CS-CoMA-DICE with oracle training slices reaches 0.988 ± 0.000 while pooled reverse KL CoMA-DICE stays at 0.125 ± 0.000. The selector-only BC control already improves over pooled BC, but summary k-means leaves inferred CS and separate-slice training at 0.125 ± 0.000. The main inferred Horizon-2 row uses the temporal run without rewards, where temporal embeddings recover perfect slice accuracy and the oracle level return.

ConventionRepeat-HH and the cooperative boundary control post-reveal. The main ConventionRepeat-HH rows use tuple weighting with $\alpha = 0.25$ followed by weighted BC. Pooled training returns 0.0005 ± 0.0000, supplied-slice CS returns 0.9220 ± 0.0000, and supplied separate-slice training returns 0.8513 ± 0.0000. Summary inference gives 0.7467 ± 0.4171 for CS and 0.6848 ± 0.3823 for separate-slice training. In the cooperative post-reveal control, pooled and selected references return 0.9843 ± 0.0000 and 0.9837 ± 0.0000, respectively; pooled BC returns 0.9765 ± 0.0000. These rows support the before-reveal versus post-reveal distinction within the tuple-weighting pipeline.

Direct-policy temporal companion. Table 10 reports a separate ConventionRepeat-HH study that directly optimizes the product-policy reverse-KL objective with $\alpha = 1$. It uses 200 Adam steps at learning rate 0.05, smoothing 10^{-4}, selector temperature 0.05, refresh interval one, and the same five seeds. Temporal GRU embeddings are fitted to 32 episodes per slice without reward features before k-means. Its supplied-slice controls share this direct objective. This study establishes temporal-inference feasibility for that objective; comparison against the main summary rows also changes the learner and regularization weight.

Table 10: ConventionRepeat-HH direct-policy companion at $\alpha = 1$. Entries are mean ± sample standard deviation over five paired seeds. Separate-slice returns are averaged within each seed before aggregation.

Slices	Accuracy	CS direct	Separate-slice direct
Supplied	–	0.9982 ± 0.0010	0.9972 ± 0.0007
Temporal	0.8656 ± 0.0738	0.9993 ± 0.0008	0.9990 ± 0.0004

A.19 Label noise robustness

Table 11 corrupts the training partition on ConventionRepeat-HH. This separate diagnostic directly optimizes the product-policy objective with $\alpha = 1$, keeping the benchmark, selector, and direct-policy learner fixed across noise levels. The corruption protocol samples 64 oracle episodes per slice, independently flips each sampled episode's training label to the opposite slice with probability

$\eta \in \{0.0, 0.1, 0.2, 0.3\}$, rebuilds the slice occupancies from those corrupted labels, and then runs CS-CoMA-DICE on the unchanged benchmark. For this binary success benchmark, sequence consistency is the same success probability metric as return. Performance is effectively unchanged through 20% corruption. Variance widens at 30%, where slice accuracy falls to 0.722.

Table 11: ConventionRepeat-HH direct-policy label corruption study, $\alpha = 1$. Entries are mean $\pm$ population standard deviation (denominator n) over five paired seeds. Moderate corruption is largely absorbed; heavy corruption increases return variance.

Noise rate	Return	Sequence consistency	Slice accuracy
0.0	0.998 ± 0.001	0.998 ± 0.001	1.000 ± 0.000
0.1	0.999 ± 0.001	0.999 ± 0.001	0.909 ± 0.022
0.2	0.999 ± 0.001	0.999 ± 0.001	0.822 ± 0.026
0.3	0.604 ± 0.485	0.604 ± 0.485	0.722 ± 0.029

A separate uniform-prior K-slice check fits reward-free GRU embeddings and spectral clustering after corrupting the labels used to score alignment. Because these labels are not inputs to the unsupervised encoder or clustering, this intervention changes the evaluation of assignment accuracy rather than the learned partition. Table 12 isolates the effect of corrupting evaluation labels; robustness to corrupted training inputs requires a separate intervention. The preceding experiment instead directly corrupts the training partition and rebuilds its reference occupancies.

Table 12: Uniform-prior K-slice alignment-label check. Entries are mean $\pm$ population standard deviation (denominator n) over five seeds. Here only the evaluation labels are perturbed; reward-free inference and returns are unchanged.

K	Uncorrupted return	30% noise return	Uncorrupted accuracy	30% noise accuracy
2	0.971 ± 0.007	0.971 ± 0.007	0.850 ± 0.065	0.650 ± 0.032
3	0.976 ± 0.012	0.976 ± 0.012	0.940 ± 0.039	0.677 ± 0.028
5	0.990 ± 0.007	0.990 ± 0.007	0.909 ± 0.007	0.641 ± 0.019

A.20 Selector hyperparameter sensitivity

Figure 1 gives a compact selector hyperparameter sensitivity analysis on the exact cooperative one-step setting. Nearest-slice consistency is $\max\{P_\pi(00), P_\pi(11)\}$; bands show population standard deviations (denominator n). The parameters are $\alpha = 1$, $\tau = 0.05$, and $\varepsilon = 10^{-3}$ as recorded in Appendix A.7. On that exact objective, the default $\tau = 0.05$ stays on the same broad plateau as $\tau \in [0.005, 0.5]$, close to the one dimensional numerical benchmark 0.861, and noticeable degradation appears only once the selector is made much softer ($\tau \geq 1$). The selector refresh interval is even less sensitive on this benchmark: performance varies little over $K_{\text{refresh}} \in \{1, 2, 5, 10, 20, 40, 80\}$. The remaining gap to one is consistent with the exact product policy objective under those plotted settings.

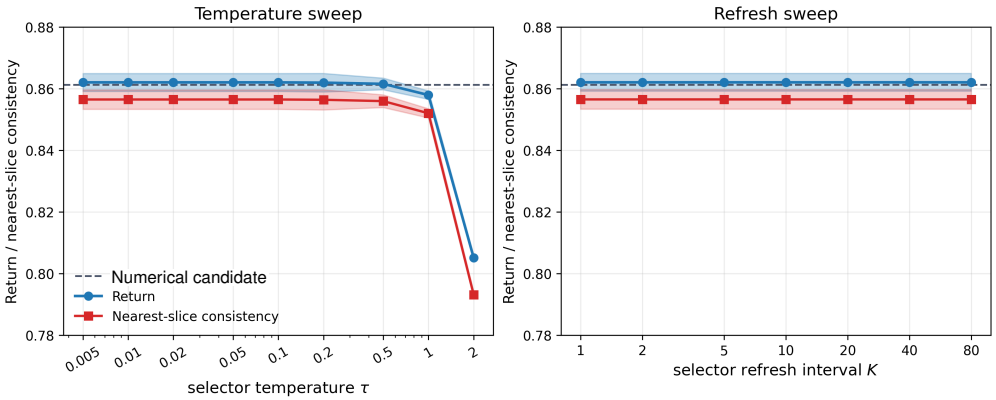


Figure 1: Selector hyperparameter sensitivity on the exact one-step setting (5 seeds, oracle training slices). Left: temperature sweep with $K_{\text{refresh}} = 1$. Right: refresh interval sweep with $\tau = 0.05$. Dashed: one dimensional numerical benchmark value 0.861.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Sections 1–7 distinguish controlled witness results, oracle and inferred slices, and descriptive continuous-control evidence.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 7 discusses deployment restrictions, support and smoothing, slice quality, initialization, and neural performance.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The main statements give their policy classes and support assumptions; the appendix supplies proofs and separates numerical companion evidence from proved results.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The accompanying code specifies controlled-task generators, learners, and reproduction commands. Appendix A.14 and Appendix A.15 specify the protocols; per-seed MaMuJoCo scores, data identities, and full-checkpoint checksums support verification.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Code and reproduction specifications accompany the study, but public access to every required asset is not established.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 6 and Appendix A.14 report seeds, checkpoint rules, defaults, smoothing, and initialization exceptions.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The controlled main tables, direct temporal companion, and MaMuJoCo tables report sample standard deviations across five seeds (denominator n-1). The appendix label-noise and alignment-label tables and selector-plot bands explicitly report population standard deviations (denominator n). Ratio summaries and higher-mean counts are descriptive; no statistical significance is claimed.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: The appendix specifies hardware types, memory, software versions, and training/evaluation budgets. Total project compute, including development runs, is not fully accounted for.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses synthetic tasks and existing offline simulation benchmarks without human subjects or sensitive personal data.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: The work studies offline regularization in simulation; deployment-level societal benefits and harms are not assessed.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The work does not release pretrained models or newly scraped datasets with identified high-risk use cases.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: Methods and datasets are cited; a complete catalog of asset licenses and terms of use is not provided.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: Synthetic task definitions and research code are described, but a complete asset license and release specification is not provided.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: No crowdsourcing or research with human subjects was conducted.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: No human-subject research requiring IRB review was conducted.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLMs are not components of the learning algorithms, datasets, or evaluation protocol.