MEMORY-EFFICIENT REPLAY-FREE POLICY EVALUATION FOR MEAN-FIELD GAMES

Anonymous authors
Paper under double-blind review

ABSTRACT

When the population size N is large, a finite N-agent game can be approximated by its mean-field limit, where the interaction among agents is represented through the population distribution. In finite-horizon evolutive mean-field games (MFGs), a common approach to computing a Nash equilibrium is to first generate the entire sequence of population distributions under a fixed policy and then fix this sequence as the external environment for policy evaluation. However, in many real-world systems, a simulator that can freely reproduce a prescribed distribution flow for subsequent policy evaluation is unavailable, which motivates a replay-free policy evaluation method. A straightforward alternative is to store the entire population history while a large finite population, approximated by the MFG, evolves forward to generate the distribution flow, and then perform policy evaluation after the forward evolution is completed. But this requires $\mathcal{O}(NT)$ memory, which can be substantial when both the population size N and the horizon T are large. In this work, we first propose a Private Population Dutch Trace Algorithm, which recursively compresses the relevant history of each agent into a trace, eliminating the memory's dependence on T. We further compress the collection of agent-wise traces and develop DoubleSWEET, which maintains state-wise traces instead, making the storage independent of N and T. Finally, we theoretically prove that the compressed state-wise traces in DoubleSWEET lead to similar policy evaluation as the private traces, for any fixed horizon, with an error upper bound of $\mathcal{O}(N^{-1/2})$; at the deterministic mean-field level, we further show that the same exact-expectation Dutch evaluator produces the same critic trajectory and learned value-function readout whether it is run live during population-flow generation or after that same flow has been frozen. In our experimental settings, DoubleSWEET achieves nearly the same policy-evaluation accuracy as the private-trace method while substantially reducing memory consumption; its memory usage does not grow with the population size N or the episode horizon T. At $N = 100{,}000$, the private-trace method requires $344.9\times$ as much memory as DoubleSWEET and the model-based projected evaluator requires approximately $40.6\times$ as much learner memory as DoubleSWEET-MC-LS while achieving the same 0.472% relative RMSE.

1 INTRODUCTION

In finite N-agent games, as the number of agents increases, computing the Nash equilibrium becomes increasingly complex due to the increased joint action space across agents. To determine the Nash equilibrium under large population, mean field games (MFGs) Laurière et al. (2024; 2022) provide a scalable formulation. In MFGs, each agent interacts with the other agents through the aggregate distribution of the other agents' states and actions. In the case with homogeneous agents, the N-agent game can be reduced to a game where a representative agent (which can be any arbitrary agent) plays against a continuum of agent population. The resulting equilibrium policies can approximate the Nash equilibria of the original finite N-agent game, with the approximation becoming more accurate as the number of agents grows Laurière et al. (2024). In this work, we focus on finite-horizon evolutive MFGs, where the population distribution evolves over time from a given initial distribution under a time-dependent policy.

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

To determine the Nash equilibrium of MFGs, a family of existing methods Laurière et al. (2024; 2022) adopt the following process based on policy evaluation. The process converging to Nash equilibrium takes multiple episodes. Within each episode, given a fixed policy, a two-step procedure is adopted: (S1) Population Flow Generation, the population is propagated forward under the shared and fixed policy to obtain a sequence of their state distributions over the entire episode; (S2) Policy Evaluation, the representative agent regards the generated sequence of state distributions as its external environment, and evaluates the given fixed policy by estimating the value (i.e., expected return) for each time-state pair. After this two-step procedure, the resulting value estimation is used to improve the policy by assigning higher probabilities to actions with higher estimated values. This process is repeated for multiple episodes until the policy and its induced sequence of distributions reach a fixed point, i.e., a mean-field Nash equilibrium. Note that the two-step separation is well-motivated, because once the entire sequence of state distribution is fixed, the representative agent's problem reduces to an ordinary time-inhomogeneous finite-horizon Markov decision process, and standard policy-evaluation tools can be applied.

Although this two-step separation makes policy evaluation tractable, it may be infeasible in real-world systems because the population distribution flow generated in (S1) cannot always be reproduced as a prescribed external environment for policy evaluation in (S2). For example, traffic routing is a representative MFG application Tanaka et al. (2021); in real systems, however, forcing a large population of drivers into a prescribed configuration solely for re-evaluation can be impractical, as highlighted in oracle-free MFG work Yardım et al. (2023); Benjamin & Abate (2023). Other examples include electric-vehicle charging Zhu et al. (2016), pedestrian crowds Lachapelle & Wolfram (2011), and cloud resource management Mao et al. (2022).

Related oracle-free or learning-while-playing methods avoid repeatedly solving policy evaluation under arbitrarily prescribed population configurations, but focus on stationary or infinite-horizon MFG formulations with a time-independent mean-field state Yardım et al. (2023); Xie et al. (2021). However, in a finite-time MFG, the state distribution is time-varying, and the future value of an agent depends on the time-varying distributions over the remaining episode. To address this time-varying scenario, a naive idea is to store the full data of all N agents across all T time steps in the entire episode for policy evaluation. Nevertheless, this leads to a space complexity of $\mathcal{O}(NT)$, which is significant because N and T are usually very large.

In this work, we aim to answer this main question: **In finite-time MFG with time-varying state distribution, can we propose a policy evaluation method for (S2) that is replay-free and has a space complexity of $\mathcal{O}(1)$ with respect to N and T?** This method can provide a memory-efficient approach to policy evaluation for Nash equilibrium determination in MFGs, removing the need for a real-world simulator. To address this question, first, inspired by trace-based reinforcement learning (e.g., TD(λ), LSTD(λ), and Dutch-trace-based methods Sutton (1988); Boyan (2002); van Seijen & Sutton (2014)), we propose maintaining one trace for each agent in (S1) to summarize the history needed for policy evaluation, making the space complexity independent of the number of time steps T. Second, we further reduce the memory by avoiding storing one trace for every agent. This is challenging because agents in the same state can have different traces because of different past histories such that naively averaging them leads to an incorrect parameter update. To address this, inspired by expected eligibility traces van Hasselt et al. (2021); van Zuijlen & Antunes (2025), we propose state-wise expected traces to summarize the private traces of agents in the mean-field setting. Finally, we theoretically analyze the bounds of the error resulting from mean-field approximation and the expected trace-based policy evaluation, validating our proposed mean-field trace-based method a valid approximation for a large but finite population game.

Our main contributions are summarized as follows:

- Private Population Dutch Trace Algorithm: To enable replay-free memory-efficient Nash equilibrium determination in MFGs, we propose a policy evaluation algorithm that summarizes the history of each agent using traces. It achieves a T-independent space complexity of $\mathcal{O}(Nd + d^2)$, with d being the dimension of critic parameter in the value function.
- The DoubleSWEET Algorithm: We compress the agent-wise private traces to state-wise expected traces to reduce the memory consumption. Unlike conventional expected traces, our method leverages the mean-field approximation to obtain a closed recursive update for state-wise expected traces. It achieves a space complexity of $\mathcal{O}(Sd + d^2)$, with S being the number of states. Under the large-population setting, we prove that the gap between the

compressed and state-aggregated private credit masses vanishes at an $\mathcal{O}(N^{-1/2})$ rate. This compression principle further extends to conventional accumulating eligibility traces, including LSTD(λ) Boyan (2002). Under the episodic $\lambda = 1$ specialization, it yields a ridge-regularized least-squares estimator based on Monte Carlo returns Sutton & Barto (2018), which we refer to as Ridge MC-LS; the corresponding theoretical guarantees continue to hold.

- Theoretical Analysis: We prove that the upper-bounds of the policy evaluation with the finite-N private traces, their compressed state-wise traces, and the mean-field trace recursions are $\mathcal{O}(N^{-1/2})$, which diminishes and approaches zero under large population. Meanwhile, considering MFGs, we prove an execution-order equivalence at the deterministic mean-field level: the same exact-expectation Dutch evaluator produces the same critic trajectory and learned value-function readout whether policy evaluation is performed during (S1) Population Flow Generation or after the same population flow has first been generated and frozen.
- Experiments: Our experiments show that DoubleSWEET maintains nearly the same accuracy as the Private Population Dutch Trace Algorithm while substantially reducing memory consumption. At $N = 20{,}000$, the private trace method requires $69.1\times$ as much memory as DoubleSWEET, and this ratio increases to $344.9\times$ at $N = 100{,}000$. DoubleSWEET also requires less processing time per step. As N grows from $5{,}000$ to $100{,}000$, its processing time increases by only $4.17\times$, compared with $8.96\times$ for the private trace method. Similar observations hold for the extended variants under LSTD(λ) and Ridge MC-LS. We additionally compare against classical policy-evaluation methods, including TD(0) and MC-SGD, together with subsampled variants and model-based tabular/projected evaluators. These comparisons highlight complementary advantages of DoubleSWEET over different baseline families, including improved policy-evaluation accuracy, reduced learner memory, and the ability to exploit the full population data without subsampling.

2 Problem Setup

In MFGs, the determination of the Nash equilibrium iterates for multiple episodes. We focus on one episode and aim at proposing a replay-free and memory-efficient algorithm for evaluating the policy used in this episode. In this episode, suppose there are T finite and discrete time steps, denoted by $t = 0, \dots, T$. There are N homogeneous and indistinguishable agents, where $N \gg 1$. Let $\mathcal{X} \triangleq \{1, \dots, S\}$ and $\mathcal{A} \triangleq \{1, \dots, K\}$ denote the set of the states and actions of agents, respectively. These agents follow the same policy $\boldsymbol{\pi} = (\pi_t, t = 0, 1, \dots, T-1)$, where π_t specifies the probability that an agent chooses action a given its state x. The policy π_t varies across time, the goal is to evaluate $\boldsymbol{\pi}$.

At time $t = 0$, each agent $i = 1, 2, ..., N$ has an initial state $X_0^{i,N} \in \mathcal{X}$ independently and randomly sampled from a distribution u_0, i.e., $X_0^{i,N} \sim u_0$. At time t, each agent has a recent state $X_t^{i,N} \in \mathcal{X}$, and the empirical state distribution of all agents is denoted by $U_t^N(x) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}_{\{X_t^{i,N}=x\}}$. Given its state $X_t^{i,N}$ and distribution U_t^N, each agent i takes an action based on the given policy π_t at time t. Then, it transits to the next state $X_{t+1}^{i,N}$ according to a random process, with transition kernel $P_t(\cdot)$[1], and obtains a reward $R_{t+1}^{i,N}$ based on a deterministic reward function $r_t(\cdot)$ as follows:

$$X_{t+1}^{i,N} \sim P_t\Big(\cdot \mid X_t^{i,N}, U_t^N\Big), \qquad R_{t+1}^{i,N} = r_t\Big(X_t^{i,N}, U_t^N\Big), \tag{1}$$

Evaluating the given policy $\boldsymbol{\pi}$ corresponds to estimating the long-term value $V^N(t, x)$ of an agent conditioned on its current state for all t and x.[2]

$$V^N(t, x) = \mathbb{E}\left[\sum_{s=t}^{T-1} \gamma^{s-t} R_{s+1}^{i,N} \,\middle|\, X_t^{i,N} = x\right]. \tag{2}$$

[1] Since the policy is fixed during policy evaluation, action sampling under π_t is absorbed into the closed-loop transition kernel P_t.

[2] For simplicity, we focus on estimating the state-value function $V^N(t, x)$. Estimating the state-action value $Q^N(t, x, a)$ is a direct extension: one can treat (x, a) as an augmented state and replace $\phi(t, x)$ with $\phi(t, x, a)$.

where the expectation is with respect to the state transition randomness, and $\gamma \in [0, 1]$ is a discount factor. Since the agents are homogeneous, $V^N(t, x)$ is identical for all agents. To estimate $V^N(t, x)$, we approximate the value function using a linear critic:

$$V_w(t, x) = w^\top \phi(t, x), \qquad w \in \mathbb{R}^d, \tag{3}$$

where $\phi : \{0, \dots, T\} \times \mathcal{X} \to \mathbb{R}^d$ is a user-chosen feature representation of each time-state pair (t, x). This choice follows the linear value-function approximation used in temporal-difference learning Sutton (1988) and in True Online TD(λ) van Seijen & Sutton (2014).

In this work, the goal is to estimate $V_w(t, x)$ by estimating w in (3), with which the given policy is evaluated. However, the state transition random process $P_t(\cdot)$ is unknown, so $V_w(t, x)$ must be estimated based on the observed transitions and rewards within the entire episode. A straightforward way is to store all these observations for the policy evaluation at the end of the episode, while this leads to a significant memory consumption of $\mathcal{O}(NT)$, since N and T are very large. To reduce the memory, one may retain trajectories from only a small subset of agents, but this wastes most of the data generated during the population flow generation. Alternatively, one may estimate a time-dependent transition model from the observed transitions and then compute the value function backward using the estimated model. However, this requires estimating and maintaining time-dependent transition models throughout the horizon, which can incur substantial memory costs when the state space is large or the horizon T is long.

This motivates us to propose a trace-based approach. Instead of retaining the entire history, a trace recursively summarizes only the historical information needed for parameter updates. We apply the Dutch-trace construction van Seijen & Sutton (2014) because it admits an exact online forward–backward formulation, i.e., the parameter can be updated as each new population transition arrives without replaying the full history. It is also closely related to standard eligibility traces Sutton (1988), allowing the analysis to cover the broader trace-based methods developed later.

3 METHODOLOGY

We begin with a base algorithm called Private Population Dutch Trace, which summarizes the trace for each agent and achieves a space complexity independent of T. Then, we extend this algorithm to DoubleSWEET Algorithm, which performs further compression by storing state-wise expected traces, making the space complexity irrelevant to N, i.e., $\mathcal{O}(1)$ with respect to N and T.

3.1 BASE ALGORITHM: PRIVATE POPULATION DUTCH TRACE ALGORITHM

Inspired by trace-based reinforcement learning TD(λ) Sutton (1988); van Seijen & Sutton (2014), we propose the Private Population Dutch Trace Algorithm for updating the estimation parameter w in (3). We theoretically show that under synchronized updates over all agents at each time step assumption, this algorithm leads to a w that is equal to that under full data storage, with details in Appendix B, Sec. B.3. In this algorithm, (i) the value of w is updated across time steps, making (S1) Population Flow Generation and (S2) Policy Evaluation operate simultaneously. Let $w_t^{N,D} \in \mathbb{R}^d$ denote the estimated w of (3) at time step t, where D stands for Dutch Trace. (ii) Each agent maintains a private trace that summarizes information from its own history and the population history, which is updated at each time step. Let $e_t^{i,N,D} \in \mathbb{R}^d$ denote the private trace of agent i at time t. We initialize $w_0^{N,D} = w_{\text{init}}$ and $e_{-1}^{i,N,D} = 0$ for all agents.

At time t, we define a population-level transformation matrix $A_t^N = I_d - \frac{\alpha}{N} \sum_{j=1}^N \phi_t^{j,N} (\phi_t^{j,N})^\top$ to summarize the current population state in the feature space, where α is a learning rate and $\phi_t^{i,N} = \phi(t, X_t^{i,N})$ transforms the agent's state $X_t^{i,N}$ to the feature representation. Then, the private trace is updated as follows:

$$e_t^{i,N,D} = \phi_t^{i,N} + \gamma\lambda A_t^N e_{t-1}^{i,N,D}. \tag{4}$$

Intuitively, the trace $e_t^{i,N,D}$ weights the contribution of agent i's past state features to the update of critic in (3). After observing the state transition and reward at time t, parameter w is updated by

$$w_{t+1}^{N,D} = w_t^{N,D} + \frac{\alpha}{N} \sum_{i=1}^N \left(\delta'_{t,i} e_t^{i,N,D} - c_{t,i} \phi_t^{i,N} \right), \tag{5}$$

where $c_{t,i} = (w_t^{N,D} - w_{t-1}^{N,D})^\top \phi_t^{i,N}$, and $\delta'_{t,i} = R_{t+1}^{i,N} + \gamma \mathbf{1}_{\{t+1<T\}} (w_t^{N,D})^\top \phi_{t+1}^{i,N} - (w_t^{N,D})^\top \phi_t^{i,N} + c_{t,i}$, with $w_{-1}^{N,D} = w_0^{N,D}$. The derivation details of correction terms $c_{t,i}$ and $\delta'_{t,i}$ are in Appendix B, Sec. B.2.

Remark 1 (Theoretical Equivalence)**.** *We show that under the formal comparison settings and forward-target convention stated in Appendix B, Sec. B.3, this Private Population Dutch Trace Algorithm leads to an estimation $w_t^{N,D}$ that is equal to the estimation $w_t^{N,F}$ under full data storage, where the states of all agents are stored, for all $t = 1, 2, ..., T$. Due to the space limit, since this is not the main theoretical result, we place the formal theorem and proof in Appendix B, Sec. B.3.*

Remark 2 (Time and Space Complexity)**.** *Recall that d is the dimension of w. The Private Population Dutch Trace Algorithm requires a time complexity of $O(TNd^2)$, dominated by that for the matrix multiplication in computing A_t^N and $e_t^{i,N,D}$. It requires a space complexity of $O(Nd + d^2)$: the $O(Nd)$ term accounts for storing $e_t^{i,N,D}$ for all i's, while the $O(d^2)$ term is for storing A_t^N for the update in the next time $t + 1$. Note that the space complexity is independent of T.*

3.2 THE DOUBLESWEET ALGORITHMS

The Private Population Dutch Trace Algorithm stores one trace for each agent, so its memory cost grows with N, which can be substantial under the large population game. On the other hand, under our discrete-state setting, multiple agents typically occupy the same state at a given time. This motivates us to store one trace for each occupied state rather than one trace for each individual agent, thereby compressing the collection of private traces. As mentioned earlier, agents in the same state can have different traces due to their different past histories, making naively averaging lead to incorrect parameter update. Thus, we propose DoubleSWEET that uses state-wise expected traces for compression. To begin with, inspired by van Hasselt et al. (2021), we show that private traces can be replaced by their conditional expected traces without changing the expected parameter update. However, in finite-N agent game, this conditional expected trace is not state-wise and does not admit a closed recursion based on the agent state alone, because its evolution also depends on the random empirical population distribution. Then, we approximate it by considering the mean-field limit (i.e., letting the population size approach infinity, which is reasonable under large population game), under which the empirical population distribution is replaced by a deterministic population flow, allowing the expected-trace recursion to close at the state level. Finally, we present DoubleSWEET and derive its time and space complexity.

Expected One-Step Update Preservation Theorem. For time $t \in \{0, \dots, T-1\}$, we define the finite-N conditional expected trace as

$$z_t^{N,D}(x, u) = \mathbb{E}\left[e_t^{i,N,D} \mid X_t^{i,N} = x,\, U_t^N = u \right]. \tag{6}$$

This expectation is conditioned on $X_t^{i,N}$ and U_t^N, because the state transition and reward of an agent depends on the agent's state and the population state distribution. Meanwhile, since all agents are homogeneous, $z_t^{N,D}(x, u)$ does not depend on the agent index i.

We show that given any parameter pair $(w, w^-) \in \mathbb{R}^d \times \mathbb{R}^d$, the expected updates of w under private trace $e_t^{i,N,D}$ and conditional expected trace $z_t^{N,D}(x, u)$ are equal. Define $c_{t,i}(w, w^-) = (w - w^-)^\top \phi_t^{i,N}$ and $\delta'_{t,i}(w, w^-) = R_{t+1}^{i,N} + \gamma \mathbf{1}_{\{t+1<T\}} w^\top \phi_{t+1}^{i,N} - (w^-)^\top \phi_t^{i,N}$. Then, the expected update of w using (5) under private and conditional expected traces are

$$\Delta_{\text{priv}}^{N,t}(w, w^-) = \frac{\alpha}{N} \sum_{i=1}^{N} \left[\delta'_{t,i}(w, w^-) e_t^{i,N,D} - c_{t,i}(w, w^-) \phi_t^{i,N} \right], \tag{7}$$

$$\Delta_{\text{cond}}^{N,t}(w, w^-) = \frac{\alpha}{N} \sum_{i=1}^{N} \left[\delta'_{t,i}(w, w^-) z_t^{N,D}(X_t^{i,N}, U_t^N) - c_{t,i}(w, w^-) \phi_t^{i,N} \right]. \tag{8}$$

Theorem 1 (Update Equivalence)**.** *Under the current-context transition, adapted-trace, conditional-label-invariance, and fixed-parameter comparison assumptions stated in Appendix C, Sec. C.2, for every fixed $w, w^- \in \mathbb{R}^d$, replacing each private trace by its conditional expected trace preserves the expected one-step parameter update, i.e.,*

$$\mathbb{E}\left[\Delta_{\text{priv}}^{N,t}(w, w^-) \right] = \mathbb{E}\left[\Delta_{\text{cond}}^{N,t}(w, w^-) \right], \tag{9}$$

where the expectation is with respect to the randomness of state transition.

The proof is provided in Appendix C, Sec. C.2. This theorem motivates replacing private traces by conditional trace means. We next derive a state-wise recursion that makes this replacement implementable in the large-population setting.

Mean-Field Recursion of the Expected Trace. However, the conditional expected trace in (6) is conditioned on both the agent state and the empirical population distribution. This would require maintaining traces for different state-distribution pairs (x, u), defeating the desired state-wise compression. Thus, we propose to approximate it using the mean-field limit, under which the population distribution becomes a deterministic flow, and we can determine a state-wise expected trace.

In the mean-field limit such that $N \to \infty$, since all agents are homogeneous and follow the same policy and transition rule, each agent has the same probability distribution over states at each time, so the following discussion can focus on a representative agent (which could be any agent) without agent index. We use $\bar{X}_t$ to denote the state of the representative agent and use u_t to denote the population state distribution. At $t = 0$, we have $\bar{X}_0 \sim u_0$. For $t = 1, \dots, T$, the representative-agent state evolves as $\bar{X}_t \sim P_{t-1}(\cdot \mid \bar{X}_{t-1}, u_{t-1})$, and the population distribution evolves as $u_t(x) = \sum_{y \in \mathcal{X}} u_{t-1}(y) P_{t-1}(x \mid y, u_{t-1})$, according to the law of total probability. Since u_t depends only on u_{t-1} and the transition kernel P_{t-1}, we say that the population recursion is closed. Thus, given u_0 and the transition model, the population flow $(u_0, \dots, u_T)$ is uniquely determined, while this flow is not directly available for updating the value function due to the unknown transition kernel. Let $A_t = I_d - \alpha \sum_{x \in \mathcal{X}} u_t(x) \phi_t(x) \phi_t(x)^\top$, and define the mean-field Dutch trace by $\bar{e}^D_{-1} = 0$ and $\bar{e}^D_t = \phi_t(\bar{X}_t) + \gamma\lambda A_t \bar{e}^D_{t-1}$. For $u_t(x) > 0$, define the mean-field conditional expected trace as $z^D(t, x) := \mathbb{E}[\bar{e}^D_t \mid \bar{X}_t = x]$, and define the expected population trace (also called a mass credit) as $m^D(t, x) := \mathbb{E}[\bar{e}^D_t \mathbf{1}_{\{\bar{X}_t = x\}}]$, so $m^D(t, x) = u_t(x) z^D(t, x)$. We set $m^D(t, x) = 0$ when $u_t(x) = 0$. We define a transition-pair mass as $\Gamma_t(y, x) := \mathbb{P}(\bar{X}_{t-1} = y, \bar{X}_t = x) = u_{t-1}(y) P_{t-1}(x \mid y, u_{t-1})$. Then, for $t = 1, \dots, T$, the expected population trace $m^D(t, x)$ satisfies

$$m^D(t, x) = u_t(x)\phi_t(x) + \gamma\lambda A_t \sum_{y : u_{t-1}(y) > 0} \Gamma_t(y, x) \frac{m^D(t-1, y)}{u_{t-1}(y)}, \tag{10}$$

with $m^D(0, x) = u_0(x)\phi_0(x)$. The derivation of this forward recursion is in Appendix C, Sec. C.3. Note that we can map the expected population trace $m^D(t, x)$ back to the expected trace of the representative agent $z^D(t, x)$ based on $m^D(t, x) = u_t(x) z^D(t, x)$, with $z^D(t, x)$ being state-wise.

The DoubleSWEET Algorithm. For a large finite population, we can design the state-wise expected trace $\widehat{z}^{N,D}_t(x)$ according to $z^D(t, x)$ in the mean field limit. This is done by replacing the deterministic population distribution u_t in the mean field limit with the empirical distribution U^N_t. Specifically, at time t, define the corresponding empirical transition-pair mass as $\Gamma^N_t(y, x) = \frac{1}{N} \sum_{i=1}^N \mathbf{1}_{\{X^{i,N}_{t-1} = y,\ X^{i,N}_t = x\}}$, and let $\Gamma^N_0(y, x) = 0$. Initialize the empirical population trace $\widehat{m}^{N,D}_0(x) := U^N_0(x)\phi_0(x)$ and update it for $t = 1, \dots, T$ as follows:

$$\widehat{m}^{N,D}_t(x) = U^N_t(x)\phi_t(x) + \gamma\lambda A^N_t \sum_{y : U^N_{t-1}(y) > 0} \Gamma^N_t(y, x) \frac{\widehat{m}^{N,D}_{t-1}(y)}{U^N_{t-1}(y)}. \tag{11}$$

For each occupied state x, the resulting state-wise expected trace is $\widehat{z}^{N,D}_t(x) = \widehat{m}^{N,D}_t(x) / U^N_t(x)$. The detailed pseudocode of DoubleSWEET is provided in Appendix C, Sec. C.4.

Compared with the Private Population Dutch Trace Algorithm, instead of storing a private trace for every agent, DoubleSWEET uses the state-wise trace associated with the agent's current state.

Remark 3 (Time and Space Complexity)**.** *Let $S^N_t = |\{x \in \mathcal{X} : U^N_t(x) > 0\}|$ denote the number of occupied states at time t. DoubleSWEET requires a time complexity of $O(T(Nd + S^N_t d^2))$ and a space complexity of $O((S^N_{t-1} + S^N_t)d + d^2) = O(Sd + d^2)$, which is a function of the number of states S rather than the number of agents N.*

Now, we prove that the expected gap between using the compressed trace in DoubleSWEET and using the private trace in Private Population Dutch Trace Algorithm is upper-bounded, and this gap

approaches zero when N approaches infinity. Specifically, we define the expected population trace under the private trace setting, i.e., $m_t^{\mathrm{priv},N,D}(x) := \frac{1}{N}\sum_{i=1}^{N} e_t^{i,N,D}\mathbf{1}_{\{X_t^{i,N}=x\}}$. We establish the analysis in Appendix D, Sec. D.3 and show the following result.

Theorem 2 (Consistency of Trace Compression)**.** *Under the finite-horizon, idiosyncratic-noise, state-only-feature, and fixed-Dutch-parameter assumptions stated in Appendix D, Sec. D.3, for every fixed $T < \infty$ there exists a constant $C_{\mathrm{credit},T} < \infty$ that is independent of N such that*

$$\max_{0\le t\le T} \mathbb{E}\|\widehat{m}_t^{N,D} - m_t^{\mathrm{priv},N,D}\|_{1,2} \le C_{\mathrm{credit},T} N^{-1/2}, \tag{12}$$

where we define $\|f\|_{1,2} := \sum_{x\in\mathcal{X}} \|f(x)\|_2$ for a vector field $f : \mathcal{X} \to \mathbb{R}^d$, and this bound $C_{\mathrm{credit},T}N^{-1/2}$ approaches zero when $N \to \infty$.

Extension to Conventional Traces and LSTD(λ). The state-wise compression principle is not specific to Dutch traces. For conventional accumulating eligibility traces, the same population-credit compression can be applied directly. The resulting compressed traces can then be used to construct the sufficient statistics of LSTD(λ), yielding *DoubleSWEET-LSTD(λ)*. Appendix E, Sec. E.2 establishes the trace-level extension, and Sec. E.3 together with Sec. E.4 gives the corresponding ridge-regularized LSTD(λ) construction and consistency analysis. Ridge MC-LS is obtained as the episodic $\lambda = 1$ special case in Sec. E.5. In addition, a similar theoretical result as in Theorem 2 holds under the extensions, with the analysis established in Appendix E, Sec. E.2. Specifically, let $m_t^{N,\lambda}$ and m_t^{priv} denote the corresponding credit mass for the compressed trace in DoubleSWEET and private trace, respectively, under these extensions. We can prove that there exists a constant $C_{\lambda,T}$ such that $\max_{0\le t\le T}\mathbb{E}\|\widehat{m}_t^{N,\lambda} - m_t^{\mathrm{priv},N,\lambda}\|_{1,2} \le C_{\lambda,T}N^{-1/2}$, where this expected gap is approximately zero under large population.

3.3 Theoretical Analysis

We analyze the theoretical performance of DoubleSWEET from two levels. (i) Parameter level: The gaps between the critic parameter w's under the private trace in Private Population Dutch Trace Algorithm, the compressed state-wise trace in DoubleSWEET, and that in mean field limit are upper-bounded. (ii) Streaming level: In the deterministic mean-field setting, we compare two execution orders of the same exact-expectation Dutch evaluator: the live-flow execution, which performs policy evaluation while the population flow is generated, and the matched frozen-flow execution, which first generates the same flow and then evaluates against it.

Parameter Level. Let $w_t^{N,C}$ and $w_t^{N,D}$ denote the critic parameter w produced by DoubleSWEET and Private Population Dutch Trace Algorithm, respectively, on the finite-population episode. Let w_t^{MF} denote the critic parameter w produced by the corresponding deterministic mean-field learner.

Theorem 3 (Parameter Consistency)**.** *Under the finite-horizon, iid-initialization, no-common-noise, population-Lipschitz transition/reward, shared-feature and fixed-Dutch-parameter, and common-initialization assumptions stated in Appendix D, Sec. D.4, there exists a finite constant $C_{\mathrm{learner},T}$, which is independent of N, such that*

$$\max_{0\le t\le T} \mathbb{E}\Big[\max\Big\{\|w_t^{N,D} - w_t^{\mathrm{MF}}\|_2,\ \|w_t^{N,C} - w_t^{\mathrm{MF}}\|_2,\ \|w_t^{N,D} - w_t^{N,C}\|_2\Big\}\Big] \le \frac{C_{\mathrm{learner},T}}{\sqrt{N}}. \tag{13}$$

The analysis of this theorem is in Appendix D, Sec. D.4. In addition, similar theorems also hold for extensions of ridge-LSTD(λ) and ridge MC-LS specialization (i.e., set $\lambda = 1$), whose analyses are provided in Appendix E, Sec. E.4 (Theorem 7) and Appendix E, Sec. E.5 (Theorem 8), respectively.

Streaming Level. Let w_t^{live} and w_t^{frozen} denote the critic parameter w obtained by two execution orders of the same exact-expectation Dutch evaluator. The live-flow execution performs (S2) policy evaluation while the deterministic mean-field population flow in (S1) is generated, whereas the matched frozen-flow execution first generates that same deterministic flow and then applies the same evaluator to it using exact representative-agent expectations. Define $V_t^{\mathrm{live}}(x) := (w_t^{\mathrm{live}})^\top \phi_t(x)$ and $V_t^{\mathrm{frozen}}(x) := (w_t^{\mathrm{frozen}})^\top \phi_t(x)$, which are the corresponding estimated value functions.

Theorem 4 (Policy Evaluation Equivalence). *Under the shared deterministic mean-field environment, matched representative-agent construction, exact-expectation, and common-initialization assumptions stated in Appendix D, Sec. D.5,* $w_t^{\text{live}} = w_t^{\text{frozen}}$ *and* $V_t^{\text{live}}(x) = V_t^{\text{frozen}}(x)$ *for all* $0 \le t \le T$ *and* $x \in \mathcal{X}$.

This shows that, at the mean-field level, the live-flow execution and the corresponding matched frozen-flow execution of the same evaluator are equivalent, thereby connecting our problem directly to the standard MFG framework. The detailed proof is in Appendix D, Sec. D.5, where the theoretical statement with formal assumptions is given in Theorem 4.

4 Experiment

In the experiments, we use a synthetic Garnet-inspired finite-state MFG environment, motivated by the Garnet benchmark described in Laurière et al. (2024). The environment contains three types of rooms: ordinary, entrance, and goal rooms. Agents move among ordinary rooms and occasionally enter goal rooms through entrance rooms. Rewards are obtained only in goal rooms and decrease as more agents occupy them. Unless otherwise specified, we use $S = 256$ states, $N = 20{,}000$ agents, a horizon of $T = 500$, and critic dimension $d = 512$. We evaluate Private Population Dutch Trace Algorithm, denoted by Private Dutch, and DoubleSWEET, together with their extensions to LSTD(λ) and MC-LS. Other details are provided in Appendix F, Sec. F.1.

Expected Approximation Gap. Figure 1(a) compares the population trace m, also called credit-mass, of three methods: $\widehat{m}_t^{N,D}(x)$ of DoubleSWEET, $m_t^{\text{priv},N,D}(x)$ of Private Dutch, and $m^D(t,x)$ of Mean Field. Note that for two credit-mass a_t^N and b_t^N, we define their gap as $\max_{0\le t<T} \frac{1}{R}\sum_{r=1}^{R} \|a_{r,t}^N - b_{r,t}^N\|_{1,2}$, where $\|f\|_{1,2} := \sum_{x\in\mathcal{X}} \|f(x)\|_2$, with $R = 100$ population rollouts. Figure 1(b) reports the analogous comparisons for accumulating TD(λ) traces and MC-LS.

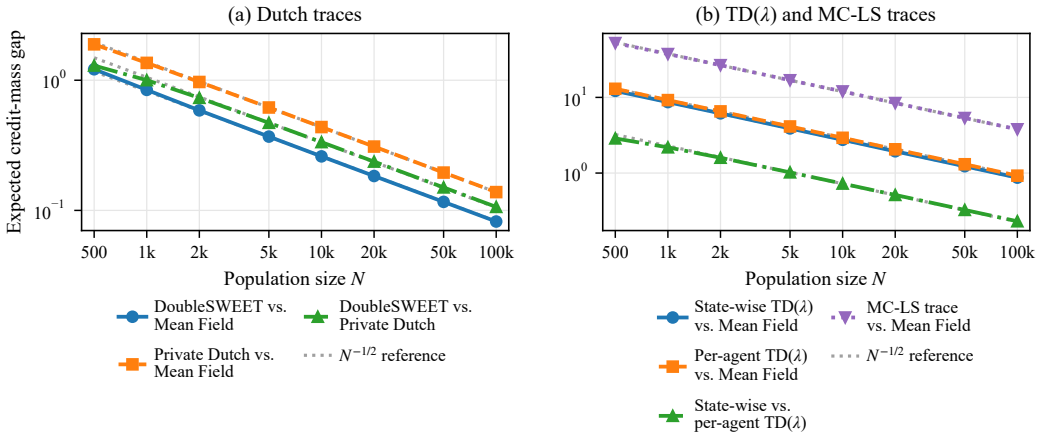

Figure 1: Expected credit-mass gaps among DoubleSWEET, Private Dutch, and Mean Field. The dotted lines indicate the $N^{-1/2}$ reference.

Figure 1 shows that all credit-mass gaps decrease approximately as $N^{-1/2}$, validating that the approximation gap due to compression and mean field limit is approximately zero under large population. For Dutch traces, the fitted slopes are -0.507, -0.495, and -0.478; for TD(λ), they are -0.499, -0.499, and -0.481; and the MC-LS slope is -0.497. In addition, these empirical results match our analytical results. For example, the result of DoubleSWEET vs. Private Dutch matches Theorem 2 that the gap between the private trace and the compressed state-wise trace approaches zero in the order of $\mathcal{O}(N^{-1/2})$.

Accuracy, Memory, and Computation Time in MFG Setting. We compare Private Dutch with DoubleSWEET, MC-LS with DoubleSWEET-MC-LS, and LSTD(λ) with DoubleSWEET-

LSTD(λ). In each pair, the latter replaces per-agent traces with state-wise compression. We vary N in Figure 2, and additional experiments with varying S and T are reported in Appendix F, Sec. F.3.

Figure 2 shows that the per-agent traces and their compressed state-wise counterparts retain nearly the same policy-evaluation accuracy, i.e., similar $V_w(t, x)$, while substantially reducing memory consumption and improving computational scaling. For memory, at $N = 20{,}000$, Private Dutch requires $69.1\times$ as much memory as DoubleSWEET, while MC-LS and LSTD(λ) each require about $25.6\times$ as much memory as their DoubleSWEET-compressed counterparts. At $N = 100{,}000$, these ratios increase to $344.9\times$ and about $125.4\times$, respectively. For processing time, as N increases from 5,000 to 100,000, the processing time grows by $8.96\times$, $7.57\times$, and $10.93\times$ for Private Dutch, MC-LS, and LSTD(λ), respectively, whereas the corresponding growth factors are only $4.17\times$, $4.53\times$, and $3.74\times$ for their DoubleSWEET-compressed counterparts.

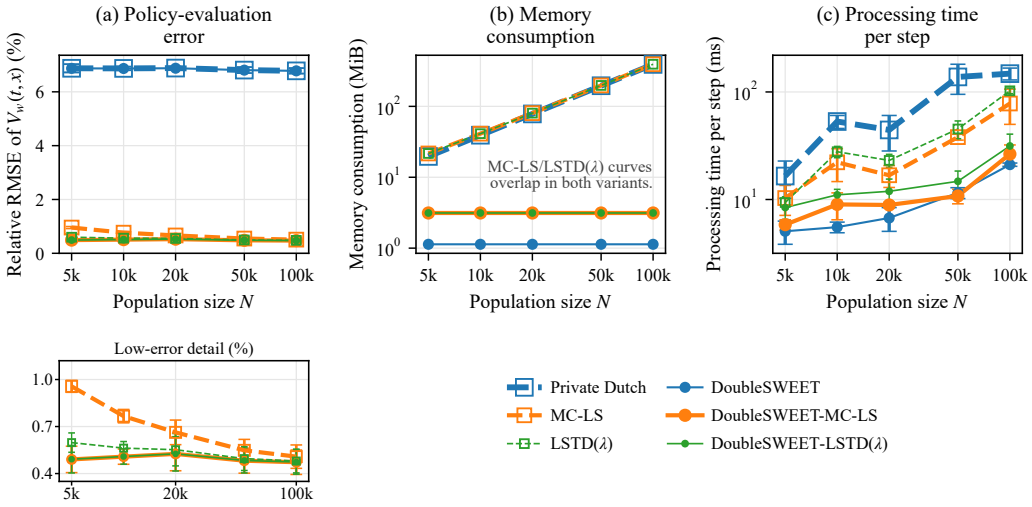

Figure 2: Impact of population size N on policy-evaluation error, memory consumption, and processing time per step. Dashed lines denote the per-agent methods, while solid lines denote their DoubleSWEET-compressed counterparts.

At $N = 100{,}000$, DoubleSWEET achieves a relative RMSE of 6.779%, compared with 26.844% for TD(0) and 26.127% for MC-SGD, while requiring 21.087 ms/step versus 2.117 and 1.775 ms/step, respectively. DoubleSWEET-MC-LS and DoubleSWEET-LSTD(λ) achieve 0.472% and 0.474% error with only 3.133 MiB of learner memory, compared with 0.472%, 127.082 MiB, and 6.466 ms/step for the model-based projected evaluator; their processing times are 26.451 and 31.458 ms/step, respectively. Complete S-, T-, and N-scaling results are reported in Appendix F, Sec. F.3. We further embed the forward-learned critics into an outer MFG policy-improvement loop; Appendix F, Sec. F.4

5 Conclusion

We study replay-free policy evaluation for finite-horizon evolutive mean-field games and develop Private Population Dutch Trace and DoubleSWEET. Our methods perform policy evaluation during the forward population evolution without storing the full population history. Private Population Dutch Trace removes the memory dependence on the horizon T, while DoubleSWEET further compresses agent-wise traces into state-wise traces so that the trace storage does not grow with either N or T. We establish fixed-horizon $\mathcal{O}(N^{-1/2})$ consistency guarantees and show that, at the mean-field level, the live-flow and matched frozen-flow executions of the same exact-expectation Dutch evaluator are equivalent. Experiments demonstrate substantial memory savings while maintaining similar policy-evaluation accuracy.Future work includes extending the state-wise compression to continuous state spaces and studying MFGs with common noise.

AI USE STATEMENT

In this work, we used generative AI tools to assist with mathematical derivations, proof drafting and checking, experimental design and implementation, and interpretation of experimental results. We also used generative AI tools for code editing, literature search, reference formatting, and drafting and editing parts of the manuscript. All AI-assisted theoretical claims, proofs, code, and experimental results were reviewed and verified by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide complete assumptions and proofs for all theoretical results in the appendix. Detailed experimental settings, evaluation metrics, implementation details, and additional scaling results are provided in Appendix F. The code package includes Markdown documentation with instructions for setting up and running the experiments.

REFERENCES

Patrick Benjamin and Alessandro Abate. Networked communication for decentralised agents in mean-field games, 2023. arXiv:2306.02766.

Dimitri P. Bertsekas and John N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, Belmont, Massachusetts, 1996. ISBN 9781886529106.

Justin A. Boyan. Technical update: Least-squares temporal difference learning. *Machine Learning*, 49(2–3):233–246, 2002. doi: 10.1023/A:1017936530646.

Aimé Lachapelle and Marie-Therese Wolfram. On a mean field game approach modeling congestion and aversion in pedestrian crowds. *Transportation Research Part B: Methodological*, 45(10): 1572–1589, 2011. doi: 10.1016/j.trb.2011.07.011.

Mathieu Laurière, Sarah Perrin, Sertan Girgin, Paul Muller, Ayush Jain, Théophile Cabannes, Georgios Piliouras, Julien Pérolat, Romuald Élie, Olivier Pietquin, and Matthieu Geist. Scalable deep reinforcement learning algorithms for mean field games. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 12078–12095. PMLR, 2022.

Mathieu Laurière, Sarah Perrin, Julien Pérolat, Sertan Girgin, Paul Muller, Romuald Élie, Matthieu Geist, and Olivier Pietquin. Learning in mean field games: A survey, 2024. arXiv:2205.12944.

Weichao Mao, Haoran Qiu, Chen Wang, Hubertus Franke, Zbigniew Kalbarczyk, Ravishankar K. Iyer, and Tamer Başar. A mean-field game approach to cloud resource management with function approximation. In *Advances in Neural Information Processing Systems*, volume 35, pp. 36243–36258, 2022. doi: 10.52202/068431-2626.

Richard S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1):9–44, 1988. doi: 10.1007/BF00115009.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, Massachusetts, 2 edition, 2018.

Takashi Tanaka, Ehsan Nekouei, Ali Reza Pedram, and Karl Henrik Johansson. Linearly solvable mean-field traffic routing games. *IEEE Transactions on Automatic Control*, 66(2):880–887, 2021. doi: 10.1109/TAC.2020.2986195.

John N. Tsitsiklis and Benjamin Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42(5):674–690, 1997. doi: 10.1109/9.580874.

Hado van Hasselt, Sephora Madjiheurem, Matteo Hessel, David Silver, André Barreto, and Diana Borsa. Expected eligibility traces. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(11):9997–10005, 2021. doi: 10.1609/aaai.v35i11.17200.

Harm van Seijen and Richard S. Sutton. True online TD(λ). In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 692–700. PMLR, 2014.

Roy van Zuijlen and Duarte Antunes. Least-squares temporal difference with expected eligibility traces. *Machine Learning*, 114, 2025. doi: 10.1007/s10994-025-06912-z. Article 269.

Qiaomin Xie, Zhuoran Yang, Zhaoran Wang, and Andreea Minca. Learning while playing in mean-field games: Convergence and optimality. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 11436–11447. PMLR, 2021.

Batuhan Yardım, Semih Cayci, Matthieu Geist, and Niao He. Policy mirror ascent for efficient and independent learning in mean field games. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 39722–39754. PMLR, 2023.

Ziming Zhu, Sangarapillai Lambotharan, Woon Hau Chin, and Zhong Fan. A mean field game theoretic approach to electric vehicles charging. *IEEE Access*, 4:3501–3510, 2016. doi: 10.1109/ACCESS.2016.2581989.

A APPENDIX

SUPPLEMENTARY MATERIAL ROADMAP

We organize the entire appendix file as follows:

In Sec. B, we introduce the Private Population Dutch Trace Algorithm and prove its correctness.

- In Sec. B.1, we introduce the setting and notation.
- In Sec. B.2, we derive the Private Population Dutch Trace Algorithm from full-history forward replay.
- In Sec. B.3, we state and prove the pathwise correctness lemma.

In Sec. C, we introduce the DoubleSWEET Algorithm and derive its state-wise trace compression.

- In Sec. C.1, we introduce the state-wise compression setting and notation.
- In Sec. C.2, we define the conditional expected Dutch trace and prove Theorem 1.
- In Sec. C.3, we state and prove the mean-field expected-trace and credit-mass closure lemma.
- In Sec. C.4, we present DoubleSWEET and show that its one-pass implementation exactly realizes the empirical recursion.

In Sec. D, we establish the finite-population consistency and mean-field guarantees for DoubleSWEET.

- In Sec. D.1, we define the fixed-window finite-population and mean-field comparison systems and notation.
- In Sec. D.2, we establish consistency of the physical population flow, transition-pair flow, feature moments, and Dutch operators.
- In Sec. D.3, we establish credit-mass consistency and prove Theorem 2.
- In Sec. D.4, we establish learner and critic consistency and prove Theorem 3.
- In Sec. D.5, we prove Theorem 4, i.e., matched live-flow/frozen-flow execution-order equivalence for the same exact-expectation Dutch evaluator.
- In Sec. D.6, we use supporting TD lemmas to prove Theorem 5, i.e., mean-field value convergence and its asymptotic value-error bound.

In Sec. E, we extend DoubleSWEET to conventional eligibility traces, ridge-LSTD(λ), and ridge MC-LS.

- In Sec. E.1, we introduce conventional accumulating eligibility traces, define the trace/credit notation used in the extensions, and review the LSTD(λ) algorithm.
- In Sec. E.2, we derive state-wise compression for conventional accumulating eligibility traces and prove Theorem 6, i.e., fixed-window trace consistency.
- In Sec. E.3, we construct the DoubleSWEET ridge-LSTD(λ) one-pass estimator.
- In Sec. E.4, we establish sufficient-statistic consistency, ridge stability, and Theorem 7, i.e., parameter and learned-value consistency with the private counterpart.
- In Sec. E.5, we specialize the construction to ridge MC-LS at $\lambda = 1$ and prove the corresponding parameter and learned-value consistency.

In Sec. F, we provide the complete experimental details and results.

- In Sec. F.1, we provide the full experimental setup, protocols, metrics, implementation details, and reproducibility information.

- In Sec. F.2, we report the trace- and parameter-convergence results for DoubleSWEET-LSTD(λ) and DoubleSWEET-MC-LS.
- In Sec. F.3, we report the accuracy, computation, and memory trade-offs under the S-, T-, and N-scaling experiments.
- In Sec. F.4, we report the MFG-solving results using forward-learned critics, including policy-distance trajectories and finite-population unilateral-deviation gaps.

B Private Population Dutch Trace and Pathwise Exactness

B.1 Finite-Population Evaluation Setting

This section fixes the notation used by both the full-history forward view and the Private Population Dutch Trace view. The symbols are grouped by their roles so that every object used later in Appendix B is defined before the algorithmic derivation begins.

Population and time indices. There are N agents and a finite episode with physical times $t = 0, \dots, T$. We use $i \in [N] := \{1, \dots, N\}$ for an agent index, s for a historical time, n for an n-step look-ahead length, and τ for a query time at which a learned critic is evaluated. The state space is $\mathcal{X} := \{1, \dots, S\}$. The state of agent i at time t is $X_t^{i,N} \in \mathcal{X}$. Its empirical population law is

$$U_t^N(x) := \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}_{\{X_t^{i,N} = x\}}, \qquad x \in \mathcal{X}. \tag{B.1}$$

The realized reward on transition $t \to t+1$ is denoted by $R_{t+1}^{i,N}$. Under the model in the main paper,

$$R_{t+1}^{i,N} = r_t(X_t^{i,N}, U_t^N). \tag{B.2}$$

For the pathwise forward/backward comparison below, the same realized states and rewards are supplied to both views.

Critic and feature symbols. The critic parameter is shared by the whole population. Its dimension is d, and the time-state feature of agent i is

$$\phi_t^{i,N} := \phi_t(X_t^{i,N}) = \phi(t, X_t^{i,N}) \in \mathbb{R}^d. \tag{B.3}$$

The linear critic is

$$V_w(t, x) = w^\top \phi_t(x), \qquad w \in \mathbb{R}^d. \tag{B.4}$$

The agent index appears on the observed feature $\phi_t^{i,N}$, not on the critic parameter: all agents contribute to one shared w.

Discount, trace, and learning-rate symbols. The scalar parameters are

$$\alpha > 0, \qquad 0 \le \gamma \le 1, \qquad 0 \le \lambda \le 1, \qquad \beta := \gamma\lambda. \tag{B.5}$$

Here α is the critic learning rate, γ is the discount factor, λ is the trace parameter, and β is a shorthand used in the Dutch recursion. We write I_d for the $d \times d$ identity matrix.

Forward-view and Dutch-view symbols. The superscript F denotes the full-history forward-replay construction and D denotes the Private Population Dutch implementation. Thus $w_t^{N,F}$ and $w_t^{N,D}$ are their respective shared critic parameters, while $e_t^{i,N,D} \in \mathbb{R}^d$ is the private Dutch trace of agent i. Both constructions start from the same parameter,

$$w_{-1}^{N,F} = w_0^{N,F} = w_{-1}^{N,D} = w_0^{N,D} = w_{\text{init}}, \qquad e_{-1}^{i,N,D} = 0. \tag{B.6}$$

For concise terminal notation, define

$$\chi_k := \mathbf{1}_{\{k<T\}}, \qquad 0 \le k \le T. \tag{B.7}$$

Hence a bootstrap from time k is multiplied by χ_k and vanishes when $k = T$.

Synchronous population-batch convention. At one physical time t, all N agent contributions are evaluated using the same pre-update critic pair (w_t, w_{t-1}). The population increment is formed only after all agent contributions from that time slice have been evaluated; the shared parameter is then updated once. This synchronous convention is used in both views throughout Appendix B.

Table 1: Core symbols used in Appendix B.

Symbol	Meaning
N, i	population size and agent index $i \in [N]$
T, t, s, n, τ	horizon, current time, historical time, look-ahead length, and critic-query time
$\mathcal{X}, S$	finite state space and its cardinality
$X_t^{i,N}, U_t^N$	state of agent i and empirical population law
$R_{t+1}^{i,N}$	realized reward on transition $t \to t+1$
$d, \phi_t(x), \phi_t^{i,N}$	critic dimension, time-state feature, and observed agent feature
w, V_w	shared critic parameter and linear critic
$\alpha, \gamma, \lambda, \beta$	learning rate, discount, trace parameter, and $\beta = \gamma\lambda$
F, D	full-history forward-replay and Private Population Dutch views
$w_t^{N,F}, w_t^{N,D}$	critic parameters produced by the two views
$e_t^{i,N,D}$	private Dutch trace of agent i
w_{init}	common deterministic critic initialization
χ_k	terminal-bootstrap indicator $\mathbf{1}_{\{k<T\}}$

B.2 Private Population Dutch Trace Algorithm

Explanation. The full-history forward learner is a correctness reference. When one additional population transition becomes available, some historical interim λ-returns change because they now include one more reward and a newer historical online bootstrap. A direct implementation would update these historical targets, reset the critic to w_{init}, and replay all stored population slices in chronological order. The key algebraic observation is that one synchronous population slice acts affinely on the incoming critic parameter. Therefore, a target change created at an old slice is propagated through a chronological product of later population-batch operators. The Private Population Dutch Trace stores the sum of exactly these propagated historical influences recursively. The corrected TD term supplies the new horizon increment, and the Dutch trace supplies its accumulated propagation through the old population slices. This produces an online update with the same parameter trajectory as the full replay while retaining only one d-dimensional trace per agent.

Formal construction. Define the empirical feature second moment and the shared population Dutch operator by

$$C_t^N := \frac{1}{N}\sum_{i=1}^{N} \phi_t^{i,N}(\phi_t^{i,N})^\top, \qquad A_t^N := I_d - \alpha C_t^N. \tag{B.8}$$

For arbitrary scalar targets $G_{t,1}, \dots, G_{t,N}$, define the synchronous population batch map

$$F_t^N(w; G_{t,1:N}) := w + \frac{\alpha}{N}\sum_{i=1}^{N}\big(G_{t,i} - w^\top \phi_t^{i,N}\big)\phi_t^{i,N}. \tag{B.9}$$

Its affine form is

$$F_t^N(w; G_{t,1:N}) = A_t^N w + \frac{\alpha}{N}\sum_{i=1}^{N} G_{t,i}\phi_t^{i,N}. \tag{B.10}$$

For target batches $H_{0:t-1}$, define chronological replay from the common initialization by

$$\mathcal{R}_0 := w_{\text{init}}, \tag{B.11}$$

$$\mathcal{R}_t(H_{0:t-1}) := F_{t-1}^N\Big(\cdots F_1^N\big(F_0^N(w_{\text{init}}; H_{0,1:N}); H_{1,1:N}\big)\cdots; H_{t-1,1:N}\Big), \qquad 1 \le t \le T. \tag{B.12}$$

For $0 \le s < T$ and $1 \le n \le T - s$, define the historical online n-step target

$$G_{s,i}^{(n)} := \sum_{m=1}^{n} \gamma^{m-1} R_{s+m}^{i,N} + \gamma^n \chi_{s+n} (w_{s+n-1}^{N,F})^\top \phi_{s+n}^{i,N}. \tag{B.13}$$

For $0 \le s < t \le T$, define the interim λ-return

$$G_{s,i}^{\lambda|t} := (1-\lambda) \sum_{n=1}^{t-s-1} \lambda^{n-1} G_{s,i}^{(n)} + \lambda^{t-s-1} G_{s,i}^{(t-s)}, \tag{B.14}$$

where the first sum is empty when $t = s+1$. The full-history forward learner at observed horizon t is

$$w_t^{N,F} := \mathcal{R}_t\Big(G_{0,1:N}^{\lambda|t}, \dots, G_{t-1,1:N}^{\lambda|t}\Big), \qquad 1 \le t \le T. \tag{B.15}$$

The Private Population Dutch trace is

$$e_t^{i,N,D} := \phi_t^{i,N} + \beta A_t^N e_{t-1}^{i,N,D}. \tag{B.16}$$

Iterating (B.16) gives

$$e_t^{i,N,D} = \sum_{s=0}^{t} \beta^{t-s} \left(A_t^N A_{t-1}^N \cdots A_{s+1}^N\right) \phi_s^{i,N}, \tag{B.17}$$

where the ordered product is I_d when $s = t$.

For $0 \le t < T$, define

$$\delta_{t,i}^D := R_{t+1}^{i,N} + \gamma \chi_{t+1} (w_t^{N,D})^\top \phi_{t+1}^{i,N} - (w_t^{N,D})^\top \phi_t^{i,N}, \tag{B.18}$$

$$c_{t,i}^D := (w_t^{N,D} - w_{t-1}^{N,D})^\top \phi_t^{i,N}, \tag{B.19}$$

$$\delta_{t,i}^{\prime,D} := \delta_{t,i}^D + c_{t,i}^D \tag{B.20}$$

$$= R_{t+1}^{i,N} + \gamma \chi_{t+1} (w_t^{N,D})^\top \phi_{t+1}^{i,N} - (w_{t-1}^{N,D})^\top \phi_t^{i,N}. \tag{B.21}$$

The Private Population Dutch parameter update is

$$w_{t+1}^{N,D} := w_t^{N,D} + \frac{\alpha}{N} \sum_{i=1}^{N} \left[\delta_{t,i}^{\prime,D} e_t^{i,N,D} - c_{t,i}^D \phi_t^{i,N}\right]. \tag{B.22}$$

Algorithm 1 Private Population Dutch Trace Algorithm

1: Initialize $w_{-1} = w_0 = w_{\text{init}}$ and $e_{-1}^i = 0$ for every $i \in [N]$.
2: **for** $t = 0, \dots, T-1$ **do**
3: Observe the population transition batch $(X_t^{i,N}, R_{t+1}^{i,N}, X_{t+1}^{i,N})_{i=1}^N$.
4: Evaluate $\phi_t^{i,N}$ and $\phi_{t+1}^{i,N}$ and form C_t^N and A_t^N by (B.8).
5: Update every trace synchronously by (B.16).
6: Using the same pre-update pair (w_t, w_{t-1}) for all agents, compute $\delta_{t,i}^D$, $c_{t,i}^D$, and $\delta_{t,i}^{\prime,D}$ from (B.18)–(B.20).
7: Form $\Delta w_t := \frac{\alpha}{N} \sum_{i=1}^N \left[\delta_{t,i}^{\prime,D} e_t^{i,N,D} - c_{t,i}^D \phi_t^{i,N}\right]$.
8: Commit simultaneously $(w_{t-1}, w_t) \leftarrow (w_t, w_t + \Delta w_t)$.
9: **end for**

With dense matrix-vector arithmetic, Algorithm 1 uses

$$\text{time} = O(TNd^2), \qquad \text{learner memory} = O(Nd + d^2). \tag{B.23}$$

The $O(Nd)$ term stores the N private traces; the shared critic pair contributes only $O(d)$ storage, and the dense population second-moment/Dutch operator contributes $O(d^2)$. Full-history replay, in contrast, retains all NT agent-time observations. Thus the Private Population Dutch construction removes the learner-memory dependence on T but not on N.

B.3 Finite-Population Forward–Backward Pathwise Exactness Lemma

Explanation. The two constructions represent the same sequence of population updates in different forms. The forward view explicitly revises historical labels and replays them, whereas the Dutch view recursively stores their propagated influence. For an exact comparison, both views must use the same realized population data, the same feature vectors and scalar hyperparameters, the same initialization and terminal convention, the same synchronous population-batch semantics, and the historical online bootstrap in (B.13). Under these aligned conditions, every horizon extension in the forward replay produces exactly the increment represented by the Dutch update.

Comparison setting B-A1 (Common realized data and critic representation). The forward and Dutch views are evaluated on exactly the same realized finite-population trajectory and realized rewards. They also use the same time–state feature representation and the same shared linear critic architecture.

Mathematical form of B-A1.

$$\mathcal{D}^F = \mathcal{D}^D =: \mathcal{D}, \qquad \mathcal{D} = \mathcal{D}_X \cup \mathcal{D}_R, \tag{B.24}$$

$$\mathcal{D}_X := \left\{ X_t^{i,N}, \phi_t^{i,N} : 1 \le i \le N,\ 0 \le t \le T \right\}, \tag{B.25}$$

$$\mathcal{D}_R := \left\{ R_{t+1}^{i,N} : 1 \le i \le N,\ 0 \le t < T \right\}, \tag{B.26}$$

$$V_w(t,x) = w^\top \phi_t(x), \qquad w \in \mathbb{R}^d. \tag{B.27}$$

Comparison setting B-A2 (Common parameters, initialization, and terminal convention). The two views use identical learning, discount, and trace parameters. They start from the same critic parameter and use the same terminal-bootstrap convention at every time.

Mathematical form of B-A2.

$$\alpha^F = \alpha^D = \alpha, \qquad \gamma^F = \gamma^D = \gamma, \qquad \lambda^F = \lambda^D = \lambda, \tag{B.28}$$

$$w_{-1}^{N,F} = w_0^{N,F} = w_{-1}^{N,D} = w_0^{N,D} = w_{\mathrm{init}}, \qquad e_{-1}^{i,N,D} = 0, \tag{B.29}$$

$$\chi_k = \mathbf{1}_{\{k<T\}}. \tag{B.30}$$

Comparison setting B-A3 (Synchronous population-batch update). Within one physical population slice, every agent contribution is evaluated using one common pre-update critic pair. The shared critic is changed only after all contributions from that slice have been aggregated.

Mathematical form of B-A3.

$$F_t^N(w; G_{t,1:N}) = w + \frac{\alpha}{N} \sum_{i=1}^{N} \big(G_{t,i} - w^\top \phi_t^{i,N}\big) \phi_t^{i,N}. \tag{B.31}$$

Forward-target convention B-A4 (Historical-online forward target). Each historical multi-step forward target bootstraps from the critic parameter that was actually available at the corresponding bootstrap time.

Mathematical form of B-A4.

$$G_{s,i}^{(n)} = \sum_{m=1}^{n} \gamma^{m-1} R_{s+m}^{i,N} + \gamma^n \chi_{s+n} (w_{s+n-1}^{N,F})^\top \phi_{s+n}^{i,N}. \tag{B.32}$$

Lemma 1 (Finite-N Population Forward–Backward Pathwise Exactness)**.** *Under comparison settings B-A1, B-A2, and B-A3, together with forward-target convention B-A4, the full-history forward-replay learner and the Private Population Dutch Trace learner produce identical critic parameters, and therefore identical critic readouts, at every time in the finite horizon.*

$$\begin{aligned} w_t^{N,F} &= w_t^{N,D}, \qquad 0 \le t \le T, \\ V_{w_t^{N,F}}(\tau, x) &= V_{w_t^{N,D}}(\tau, x), \qquad 0 \le t, \tau \le T,\ x \in \mathcal{X}. \end{aligned} \tag{B.33}$$

Remark 4. *The equality in Lemma 1 is finite-N, finite-horizon, pathwise, and exact. It is an algebraic property of the shared linear critic, synchronous population-batch map, historical online bootstrap, and common realized data stream. Sequential within-slice parameter updates define a different ordered update and are outside this lemma.*

Proof. The proof has three steps. First, when the observed horizon grows from t to $t+1$, every old interim λ-return changes by one corrected TD innovation multiplied by β^{t-s}. Second, the affine replay map propagates a target perturbation created at historical slice s through the chronological product of all later population operators; summing these propagated perturbations produces the Dutch trace. Third, under the induction hypothesis, the forward and backward TD quantities are identical, so the horizon increment of the full replay is exactly the Private Population Dutch update.

For $0 \le t < T$, define

$$\delta^F_{t,i} := R^{i,N}_{t+1} + \gamma\chi_{t+1}(w^{N,F}_t)^\top \phi^{i,N}_{t+1} - (w^{N,F}_t)^\top \phi^{i,N}_t, \tag{B.34}$$

$$c^F_{t,i} := (w^{N,F}_t - w^{N,F}_{t-1})^\top \phi^{i,N}_t, \tag{B.35}$$

$$\delta'^{,F}_{t,i} := \delta^F_{t,i} + c^F_{t,i} \tag{B.36}$$

$$= R^{i,N}_{t+1} + \gamma\chi_{t+1}(w^{N,F}_t)^\top \phi^{i,N}_{t+1} - (w^{N,F}_{t-1})^\top \phi^{i,N}_t. \tag{B.37}$$

1. Horizon increment. For $0 \le s < t < T$, let $n = t - s$. Since $s + n = t < T$, we have $\chi_{s+n} = \chi_t = 1$. Then

$$G^{\lambda|t+1}_{s,i} - G^{\lambda|t}_{s,i} = \lambda^n \big(G^{(n+1)}_{s,i} - G^{(n)}_{s,i}\big) \tag{B.38}$$

$$= \lambda^n\gamma^n \Big[R^{i,N}_{t+1} + \gamma\chi_{t+1}(w^{N,F}_t)^\top \phi^{i,N}_{t+1} - (w^{N,F}_{t-1})^\top \phi^{i,N}_t\Big] \tag{B.39}$$

$$= \lambda^n\gamma^n \delta'^{,F}_{t,i} \tag{B.40}$$

$$= \beta^{t-s}\delta'^{,F}_{t,i}. \tag{B.41}$$

2. Affine replay and Dutch trace. Repeated substitution of (B.10) gives

$$\mathcal{R}_t(H_{0:t-1}) = A^N_{t-1}\cdots A^N_0 w_{\text{init}} \tag{B.42}$$

$$+ \sum_{s=0}^{t-1} \big(A^N_{t-1}\cdots A^N_{s+1}\big)\, \frac{\alpha}{N} \sum_{i=1}^N H_{s,i}\phi^{i,N}_s, \tag{B.43}$$

with the product for $s = t-1$ equal to I_d. For $t \ge 1$, define the horizon-$(t+1)$ replay after the old slices only by

$$\widetilde{w}^{N,F}_t := \mathcal{R}_t\Big(G^{\lambda|t+1}_{0,1:N}, \ldots, G^{\lambda|t+1}_{t-1,1:N}\Big), \tag{B.44}$$

and set $\widetilde{w}^{N,F}_0 := w_{\text{init}}$. By (B.42), (B.41), and (B.17),

$$A^N_t(\widetilde{w}^{N,F}_t - w^{N,F}_t) = \frac{\alpha}{N}\sum_{i=1}^N \delta'^{,F}_{t,i} \sum_{s=0}^{t-1} \beta^{t-s}\left(A^N_t\cdots A^N_{s+1}\right)\phi^{i,N}_s \tag{B.45}$$

$$= \frac{\alpha}{N}\sum_{i=1}^N \delta'^{,F}_{t,i}\left(e^{i,N,D}_t - \phi^{i,N}_t\right). \tag{B.46}$$

3. Induction. The claim holds at $t = 0$ by (B.6). Assume

$$w^{N,F}_t = w^{N,D}_t, \qquad w^{N,F}_{t-1} = w^{N,D}_{t-1}$$

for $t \ge 1$, with $w_{-1} = w_0 = w_{\text{init}}$ at $t = 0$. Then

$$\delta^F_{t,i} = \delta^D_{t,i}, \qquad c^F_{t,i} = c^D_{t,i}, \qquad \delta'^{,F}_{t,i} = \delta'^{,D}_{t,i}.$$

Write the common quantities as $\delta_{t,i}$, $c_{t,i}$, and $\delta'_{t,i}$. Since

$$w^{N,F}_{t+1} = F^N_t\Big(\widetilde{w}^{N,F}_t; G^{(1)}_{t,1:N}\Big),$$

$$w_{t+1}^{N,F} - w_t^{N,F} = A_t^N(\widetilde{w}_t^{N,F} - w_t^{N,F}) \tag{B.47}$$

$$+ F_t^N\left(w_t^{N,F}; G_{t,1:N}^{(1)}\right) - w_t^{N,F} \tag{B.48}$$

$$= \frac{\alpha}{N}\sum_{i=1}^{N}\left[\delta'_{t,i}(e_t^{i,N,D} - \phi_t^{i,N}) + \delta_{t,i}\phi_t^{i,N}\right] \tag{B.49}$$

$$= \frac{\alpha}{N}\sum_{i=1}^{N}\left[\delta'_{t,i}e_t^{i,N,D} - c_{t,i}\phi_t^{i,N}\right]. \tag{B.50}$$

By (B.22),

$$w_{t+1}^{N,F} = w_{t+1}^{N,D}.$$

Thus (B.33) follows by induction, and the critic-readout equality follows from (B.4).

The forward-view horizon correction is exactly the corrected TD innovation, and the affine replay operators propagate that correction into the same accumulated history represented by the Dutch trace. The induction therefore identifies the two parameter trajectories at every time. □

C State-Wise Trace Compression and DoubleSWEET

C.1 State-Wise Trace-Compression Setting

Appendix C studies how the collection of agent-wise Dutch traces can be represented by state-wise quantities. This section introduces the finite-population information structure, the mean-field reference objects, and the distinct trace/credit quantities used in the construction.

Finite-population information structure. Let $\mathcal{F}_t^N$ contain the population states up to time t and all rewards observed before the fresh transition $t \to t+1$. For agent i, define the current state-population context

$$\mathcal{C}_{t,i}^N := \sigma(X_t^{i,N}, U_t^N) \subseteq \mathcal{F}_t^N. \tag{C.1}$$

The private Dutch trace to be compressed is the recursion

$$e_{-1}^{i,N,D} = 0, \qquad e_t^{i,N,D} = \phi_t^{i,N} + \beta A_t^N e_{t-1}^{i,N,D}, \tag{C.2}$$

where A_t^N is the synchronous population operator in (B.8). We use the same terminal indicator χ_{t+1} as in (B.7).

Mean-field reference objects. For the mean-field reference notation, let u_0 denote the prescribed initial law. With the fixed policy absorbed into the closed-loop kernel, define

$$K_t(x' \mid x) := P_t(x' \mid x, u_t), \tag{C.3}$$

and the deterministic mean-field flow

$$u_{t+1}(x') = \sum_{x\in\mathcal{X}} u_t(x)K_t(x' \mid x). \tag{C.4}$$

The representative state satisfies

$$\bar{X}_0 \sim u_0, \qquad \mathcal{L}(\bar{X}_{t+1} \mid \bar{X}_0, \ldots, \bar{X}_t) = K_t(\cdot \mid \bar{X}_t). \tag{C.5}$$

Define

$$C_t := \sum_{x\in\mathcal{X}} u_t(x)\phi_t(x)\phi_t(x)^\top, \qquad A_t := I_d - \alpha C_t, \tag{C.6}$$

and the occupied mean-field support

$$\mathcal{S}_t := \{x \in \mathcal{X} : u_t(x) > 0\}. \tag{C.7}$$

Table 2: Trace and credit objects used in Appendix C.

Object	Level	Meaning
$e_t^{i,N,D}$	finite N, agent-wise	private Dutch trace of agent i
$z_t^{N,D}(x,u)$	finite N, oracle	conditional mean of a private trace given current state and empirical population law
$\bar{e}_t^D$	mean field, trajectory-wise	Dutch trace of the representative mean-field agent
$z_t^D(x)$	mean field, state-wise	conditional expected Dutch trace given $\bar{X}_t = x$
$m_t^D(x)$	mean field, state-wise	unnormalized Dutch credit mass
$\widehat{m}_t^{N,D}(x)$	finite N, compressed	empirical DoubleSWEET credit mass
$\widehat{z}_t^{N,D}(x)$	finite N, compressed	occupied-state readout $\widehat{m}_t^{N,D}(x)/U_t^N(x)$

Map of trace and credit objects. The superscript C on a critic parameter, as in $w_t^{N,C}$ below, denotes the compressed DoubleSWEET learner and is distinct from the feature second-moment matrices C_t^N and C_t.

C.2 Conditional Expected Dutch Trace Theorem 1

Explanation. Two agents can occupy the same current state while carrying different private traces because their histories differ. A state-wise compression therefore cannot identify their realized traces directly. The parameter update, however, multiplies the private trace by a fresh transition innovation. Once the current agent state and empirical population law are fixed, the conditional law of that fresh innovation is determined by the current context, while the private trace is determined by the past. This screening property makes the conditional mean of the private trace the correct oracle replacement for a one-step comparison at a fixed parameter pair: replacing the private trace by its conditional expected trace preserves the expected update.

Formal construction. For a supported pair (x,u) with $\mathbb{P}(X_t^{i,N} = x, U_t^N = u) > 0$, define the finite-population oracle Expected-Dutch field

$$z_t^{N,D}(x,u) := \mathbb{E}\Big[e_t^{i,N,D} \mid X_t^{i,N} = x,\ U_t^N = u\Big]. \tag{C.8}$$

Assumption C-A3 below is exactly the label-invariance condition that makes this supported field independent of the chosen agent index i. Fix a deterministic comparison pair $(w, w_-) \in \mathbb{R}^d \times \mathbb{R}^d$ and define

$$\delta_{t,i}(w) := R_{t+1}^{i,N} + \gamma\chi_{t+1} w^\top \phi_{t+1}^{i,N} - w^\top \phi_t^{i,N}, \tag{C.9}$$

$$c_{t,i}(w, w_-) := (w - w_-)^\top \phi_t^{i,N}, \tag{C.10}$$

$$\delta'_{t,i}(w, w_-) := \delta_{t,i}(w) + c_{t,i}(w, w_-) \tag{C.11}$$

$$= R_{t+1}^{i,N} + \gamma\chi_{t+1} w^\top \phi_{t+1}^{i,N} - w_-^\top \phi_t^{i,N}. \tag{C.12}$$

The private and oracle Expected-Dutch one-agent contributions are

$$\Delta_{t,i}^{\mathrm{priv}}(w, w_-) := \alpha\left[\delta'_{t,i}(w, w_-) e_t^{i,N,D} - c_{t,i}(w, w_-)\phi_t^{i,N}\right], \tag{C.13}$$

$$\Delta_{t,i}^{\mathrm{ED}}(w, w_-) := \alpha\left[\delta'_{t,i}(w, w_-) z_t^{N,D}(X_t^{i,N}, U_t^N) - c_{t,i}(w, w_-)\phi_t^{i,N}\right]. \tag{C.14}$$

Define the corresponding population-batch increments, using the notation of Theorem 1 in the main paper, by

$$\Delta_{\mathrm{priv}}^{N,t}(w, w_-) := \frac{1}{N}\sum_{i=1}^N \Delta_{t,i}^{\mathrm{priv}}(w, w_-), \qquad \Delta_{\mathrm{cond}}^{N,t}(w, w_-) := \frac{1}{N}\sum_{i=1}^N \Delta_{t,i}^{\mathrm{ED}}(w, w_-). \tag{C.15}$$

Reward setting. As in the main model, the stage reward is deterministic given the current state and empirical population context:

$$R_{t+1}^{i,N} = r_t(X_t^{i,N}, U_t^N). \tag{C.16}$$

Transition-model setting C-A1 (Current-context fresh transition law). This is the finite-population transition semantics already specified by the model, not an additional regularity assumption. At the update time under consideration, once the current agent state and empirical population distribution are fixed, the fresh next-state law is determined by that current context.

Mathematical form of C-A1.

$$\mathcal{C}_{t,i}^N := \sigma(X_t^{i,N}, U_t^N) \subseteq \mathcal{F}_t^N, \tag{C.17}$$

$$\mathcal{L}(X_{t+1}^{i,N} \mid \mathcal{F}_t^N) = P_t(\cdot \mid X_t^{i,N}, U_t^N). \tag{C.18}$$

Derived property C-A2 (Adapted private trace). The current private Dutch trace is completely determined by information available before the fresh transition is sampled. This is a consequence of the stated trace recursion and filtration, rather than an additional stochastic assumption.

Mathematical form of C-A2.

$$\sigma(e_t^{i,N,D}) \subseteq \mathcal{F}_t^N, \tag{C.19}$$

$$e_{-1}^{i,N,D} = 0, \qquad e_t^{i,N,D} = \phi_t^{i,N} + \beta A_t^N e_{t-1}^{i,N,D}. \tag{C.20}$$

Because $\phi_t^{i,N}$ and A_t^N are functions of the population information available at time t, they are $\mathcal{F}_t^N$-measurable. Since $e_{-1}^{i,N,D} = 0$ and $\mathcal{F}_{t-1}^N \subseteq \mathcal{F}_t^N$, induction in the displayed recursion gives $e_t^{i,N,D} \in \mathcal{F}_t^N$ for every t.

Assumption C-A3 (Conditional label invariance). The proof only requires the supported conditional trace mean to be independent of the agent label; full joint exchangeability of the state–trace population is sufficient but not required. In the homogeneous exchangeable model this condition is automatic, but Theorem 1 uses only the weaker statement below.

Mathematical form of C-A3. For every supported (x, u) and every pair of agent labels i, j,

$$\mathbb{E}\Big[e_t^{i,N,D} \mid X_t^{i,N} = x, U_t^N = u\Big] = \mathbb{E}\Big[e_t^{j,N,D} \mid X_t^{j,N} = x, U_t^N = u\Big]. \tag{C.21}$$

Thus the supported conditional trace mean in Eq. (C.8) is well defined independently of the chosen agent label.

Comparison setting C-A4 (Fixed same-parameter comparison). The private-trace and expected-trace updates are compared at one deterministic critic-parameter pair that is held fixed during the one-step comparison. Both rules use the same terminal convention and the same synchronous population batch.

Mathematical form of C-A4.

$$(w, w_-) \in \mathbb{R}^d \times \mathbb{R}^d, \tag{C.22}$$

$$\chi_{t+1} = \mathbf{1}_{\{t+1<T\}}. \tag{C.23}$$

Theorem 1. *Under the reward setting above, transition-model setting C-A1, derived property C-A2, Assumption C-A3, and comparison setting C-A4, replacing each private trace by its conditional expected trace preserves the expected one-step population parameter update at every fixed deterministic comparison pair.*

$$\mathbb{E}\Big[\Delta_{\mathrm{priv}}^{N,t}(w, w_-)\Big] = \mathbb{E}\Big[\Delta_{\mathrm{cond}}^{N,t}(w, w_-)\Big], \qquad (w, w_-) \in \mathbb{R}^d \times \mathbb{R}^d. \tag{C.24}$$

Proof. Conditioning on $(X_t^{i,N}, U_t^N)$ separates the past trace from the fresh next-state innovation. The difference between the two update contributions is then the fresh innovation multiplied by a centered conditional-trace residual, whose conditional mean is zero.

For bounded measurable g, define

$$q_{t,g}(x, u) := \sum_{x' \in \mathcal{X}} P_t(x' \mid x, u) g\big(r_t(x, u) + \gamma \chi_{t+1} w^\top \phi_{t+1}(x') - w_-^\top \phi_t(x)\big). \tag{C.25}$$

Then

$$\mathbb{E}\big[g(\delta'_{t,i}(w,w_-)) \mid \mathcal{F}_t^N\big] = q_{t,g}(X_t^{i,N}, U_t^N) = \mathbb{E}\big[g(\delta'_{t,i}(w,w_-)) \mid \mathcal{C}_{t,i}^N\big]. \tag{C.26}$$

For bounded measurable f,

$$\mathbb{E}\Big[f(e_t^{i,N,D}) g(\delta'_{t,i}(w,w_-)) \mid \mathcal{C}_{t,i}^N\Big] \tag{C.27}$$

$$= \mathbb{E}\Big[f(e_t^{i,N,D}) \mid \mathcal{C}_{t,i}^N\Big]\, \mathbb{E}\big[g(\delta'_{t,i}(w,w_-)) \mid \mathcal{C}_{t,i}^N\big]. \tag{C.28}$$

Hence

$$\delta'_{t,i}(w,w_-) \perp e_t^{i,N,D} \mid \mathcal{C}_{t,i}^N. \tag{C.29}$$

Also,

$$\Delta_{t,i}^{\mathrm{priv}}(w,w_-) - \Delta_{t,i}^{\mathrm{ED}}(w,w_-) = \alpha \delta'_{t,i}(w,w_-) \Big[e_t^{i,N,D} - z_t^{N,D}(X_t^{i,N}, U_t^N)\Big]. \tag{C.30}$$

For every supported (x,u),

$$\mathbb{E}\Big[\Delta_{t,i}^{\mathrm{priv}}(w,w_-) - \Delta_{t,i}^{\mathrm{ED}}(w,w_-) \mid X_t^{i,N} = x, U_t^N = u\Big] \tag{C.31}$$

$$= \alpha\, \mathbb{E}\big[\delta'_{t,i}(w,w_-) \mid x,u\big]\, \mathbb{E}\Big[e_t^{i,N,D} - z_t^{N,D}(x,u) \mid x,u\Big] = 0. \tag{C.32}$$

$$\mathbb{E}\Big[\Delta_{t,i}^{\mathrm{priv}}(w,w_-) - \Delta_{t,i}^{\mathrm{ED}}(w,w_-)\Big] = \mathbb{E}\Big[\mathbb{E}\Big[\Delta_{t,i}^{\mathrm{priv}}(w,w_-) - \Delta_{t,i}^{\mathrm{ED}}(w,w_-) \mid X_t^{i,N}, U_t^N\Big]\Big] = 0. \tag{C.33}$$

Therefore

$$\mathbb{E}\Big[\Delta_{\mathrm{priv}}^{N,t}(w,w_-) \mid U_t^N\Big] = \mathbb{E}\Big[\Delta_{\mathrm{cond}}^{N,t}(w,w_-) \mid U_t^N\Big] \quad \text{a.s.}, \tag{C.34}$$

and taking expectations yields (C.24).

The argument reduces the update difference to a fresh innovation times a centered conditional-trace residual. Conditional screening factorizes these two terms, and the residual has zero conditional mean by definition of $z_t^{N,D}$. □

Remark 5. *Theorem 1 is a same-parameter expected-update identity. Here $z_t^{N,D}(x,u)$ is indexed by the current state and empirical population law. The state-only DoubleSWEET readout is $\widehat{z}_t^{N,D}(x)$, introduced in Sec. C.4.*

C.3 Mean-Field Expected-Trace and Credit-Mass Closure Lemma

Explanation. The finite-population oracle field in (C.8) still depends on the random empirical population law, so it is not a state-only object. In the mean-field system, the population flow u_t is deterministic, and the representative-agent expected trace can therefore be indexed only by the current state. Its natural recursion is predecessor-oriented: to evaluate the expected trace at a current state, one averages the previous expected traces over possible predecessor states. For forward streaming, it is more convenient to multiply the normalized expected trace by the state occupancy. The resulting unnormalized credit mass removes the Bayes normalization and obeys a forward transport recursion under the same transition kernel that transports population probability.

Setting C-M1 (Finite-horizon deterministic mean-field Markov flow). The state space and horizon are finite. Under the fixed policy, the population flow is deterministic and the representative agent evolves as the corresponding time-inhomogeneous Markov chain.

Mathematical form of C-M1.

$$|\mathcal{X}| = S < \infty, \qquad T < \infty, \tag{C.35}$$

$$u_{t+1}(x') = \sum_{x \in \mathcal{X}} u_t(x) P_t(x' \mid x, u_t), \qquad K_t(x' \mid x) = P_t(x' \mid x, u_t), \tag{C.36}$$

$$\bar{X}_0 \sim u_0, \qquad \mathcal{L}(\bar{X}_{t+1} \mid \bar{X}_0, \ldots, \bar{X}_t) = K_t(\cdot \mid \bar{X}_t), \qquad 0 \le t < T. \tag{C.37}$$

By induction in the forward equation and the representative-agent transition law,

$$\mathcal{L}(\bar{X}_t) = u_t, \qquad 0 \le t \le T. \tag{C.38}$$

Setting C-M2 (Mean-field Dutch construction). The mean-field trace uses fixed Dutch parameters and the deterministic population Dutch operator generated by the same mean-field flow.

Mathematical form of C-M2. Fix

$$\alpha > 0, \qquad 0 \le \gamma \le 1, \qquad 0 \le \lambda \le 1, \qquad \beta = \gamma\lambda, \tag{C.39}$$

and define

$$A_t = I_d - \alpha \sum_{x \in \mathcal{X}} u_t(x)\phi_t(x)\phi_t(x)^\top. \tag{C.40}$$

The representative Dutch trace is

$$\bar{e}^D_{-1} = 0, \qquad \bar{e}^D_t = \phi_t(\bar{X}_t) + \beta A_t \bar{e}^D_{t-1}. \tag{C.41}$$

Derived finite-window integrability property C-M3. Because C-M1 fixes a finite state-time domain, no separate bounded-feature or trace-integrability assumption is needed. Define

$$B_\phi := \max_{0 \le t \le T} \max_{x \in \mathcal{X}} \|\phi_t(x)\|_2 < \infty, \tag{C.42}$$

so that

$$\|A_t\|_{\mathrm{op}} \le 1 + \alpha B_\phi^2 =: B_A, \tag{C.43}$$

and hence

$$\|\bar{e}^D_t\|_2 \le B_\phi \sum_{k=0}^{t} (\beta B_A)^k < \infty. \tag{C.44}$$

Therefore all conditional expectations below are well defined on their supported states.

Derived state-wise quantities. For $x \in \mathcal{S}_t$, define

$$z^D_t(x) := \mathbb{E}[\bar{e}^D_t \mid \bar{X}_t = x]. \tag{C.45}$$

For $1 \le t \le T$, define

$$\Gamma_t(y, x) := \mathbb{P}(\bar{X}_{t-1} = y, \bar{X}_t = x) = u_{t-1}(y) K_{t-1}(x \mid y), \tag{C.46}$$

and, for $x \in \mathcal{S}_t$,

$$B_t(y \mid x) := \mathbb{P}(\bar{X}_{t-1} = y \mid \bar{X}_t = x) = \frac{\Gamma_t(y, x)}{u_t(x)}. \tag{C.47}$$

Define the unnormalized Dutch credit mass on the full state space by

$$m^D_t(x) := \mathbb{E}\left[\bar{e}^D_t \mathbf{1}_{\{\bar{X}_t = x\}}\right]. \tag{C.48}$$

Lemma 2 (Mean-Field Expected-Trace and Credit-Mass Closure)**.** *Under Setting C-M1, Setting C-M2, and the derived finite-window integrability property C-M3, the normalized mean-field Expected-Dutch trace admits an exact predecessor-state recursion, while the corresponding unnormalized credit mass equals occupancy times the normalized trace and obeys an exact denominator-free forward recursion under the same mean-field transition kernel.*

$$z^D_0(x) = \phi_0(x), \qquad x \in \mathcal{S}_0, \tag{C.49}$$

$$z^D_t(x) = \phi_t(x) + \beta A_t \sum_{y \in \mathcal{S}_{t-1}} B_t(y \mid x) z^D_{t-1}(y), \qquad 1 \le t \le T,\ x \in \mathcal{S}_t. \tag{C.50}$$

Moreover,

$$\begin{aligned} m^D_t(x) &= u_t(x) z^D_t(x), && 0 \le t \le T,\ x \in \mathcal{S}_t, \\ m^D_t(x) &= 0, && 0 \le t \le T,\ x \in \mathcal{X} \setminus \mathcal{S}_t, \end{aligned} \tag{C.51}$$

and

$$\begin{aligned} m^D_0(x) &= u_0(x)\phi_0(x), && x \in \mathcal{X}, \\ m^D_t(x) &= u_t(x)\phi_t(x) + \beta A_t \sum_{y \in \mathcal{X}} K_{t-1}(x \mid y) m^D_{t-1}(y), && 1 \le t \le T,\ x \in \mathcal{X}. \end{aligned} \tag{C.52}$$

Equivalently,

$$m^D_t(x) = u_t(x)\phi_t(x) + \beta A_t \sum_{y \in \mathcal{S}_{t-1}} \Gamma_t(y, x) z^D_{t-1}(y), \qquad 1 \le t \le T,\ x \in \mathcal{X}. \tag{C.53}$$

Proof. Condition the Dutch recursion on the current state and use the Markov property to screen the previous trace from the fresh next-state draw given the predecessor state. Multiplying the normalized conditional expectation by the current occupancy gives the credit-mass identity. For the forward recursion, expand the previous-trace contribution over predecessor states and transport each predecessor mass through the same kernel K_{t-1} that transports population probability.

For $t = 0$, (C.41) and $\bar{e}^D_{-1} = 0$ give

$$\bar{e}^D_0 = \phi_0(\bar{X}_0),$$

so conditioning on $\bar{X}_0 = x$ yields (C.49).

Fix $1 \le t \le T$ and $x \in \mathcal{S}_t$. By (C.41),

$$z^D_t(x) = \phi_t(x) + \beta A_t \, \mathbb{E}[\bar{e}^D_{t-1} \mid \bar{X}_t = x]. \tag{C.54}$$

The trace $\bar{e}^D_{t-1}$ is measurable with respect to the representative-agent history through time $t-1$. By the Markov property in C-M1,

$$\bar{e}^D_{t-1} \perp \bar{X}_t \mid \bar{X}_{t-1}.$$

Therefore, for every supported predecessor state $y \in \mathcal{S}_{t-1}$,

$$\mathbb{E}[\bar{e}^D_{t-1} \mid \bar{X}_{t-1} = y, \bar{X}_t = x] = \mathbb{E}[\bar{e}^D_{t-1} \mid \bar{X}_{t-1} = y] = z^D_{t-1}(y). \tag{C.55}$$

Applying the conditional law of total expectation with the predecessor kernel $B_t(\cdot \mid x)$ gives

$$\mathbb{E}[\bar{e}^D_{t-1} \mid \bar{X}_t = x] = \sum_{y \in \mathcal{S}_{t-1}} B_t(y \mid x) z^D_{t-1}(y). \tag{C.56}$$

Substituting this into (C.54) proves (C.50).

Next, for $x \in \mathcal{S}_t$, the definition of conditional expectation gives

$$m^D_t(x) = \mathbb{E}\big[\bar{e}^D_t \mathbf{1}_{\{\bar{X}_t = x\}}\big] \tag{C.57}$$

$$= \mathbb{P}(\bar{X}_t = x) \, \mathbb{E}[\bar{e}^D_t \mid \bar{X}_t = x] = u_t(x) z^D_t(x). \tag{C.58}$$

If $x \notin \mathcal{S}_t$, then $u_t(x) = 0$ and $\mathbf{1}_{\{\bar{X}_t = x\}} = 0$ almost surely, so $m^D_t(x) = 0$. This proves (C.51). In particular, at $t = 0$, (C.49) yields

$$m^D_0(x) = u_0(x)\phi_0(x),$$

which is the initial condition in (C.52).

For $1 \le t \le T$, multiply the trace recursion by $\mathbf{1}_{\{\bar{X}_t = x\}}$ and take expectations:

$$m^D_t(x) = u_t(x)\phi_t(x) + \beta A_t \, \mathbb{E}\big[\bar{e}^D_{t-1} \mathbf{1}_{\{\bar{X}_t = x\}}\big]. \tag{C.59}$$

Expanding over predecessor states and conditioning on the history through time $t-1$,

$$\mathbb{E}\big[\bar{e}^D_{t-1} \mathbf{1}_{\{\bar{X}_t = x\}}\big] = \sum_{y \in \mathcal{X}} \mathbb{E}\Big[\bar{e}^D_{t-1} \mathbf{1}_{\{\bar{X}_{t-1} = y\}} \mathbf{1}_{\{\bar{X}_t = x\}}\Big] \tag{C.60}$$

$$= \sum_{y \in \mathcal{X}} \mathbb{E}\Big[\bar{e}^D_{t-1} \mathbf{1}_{\{\bar{X}_{t-1} = y\}} K_{t-1}(x \mid y)\Big] \tag{C.61}$$

$$= \sum_{y \in \mathcal{X}} K_{t-1}(x \mid y) m^D_{t-1}(y). \tag{C.62}$$

Substituting (C.62) into (C.59) proves (C.52). Finally, using (C.51) at time $t-1$ and (C.46),

$$K_{t-1}(x \mid y) m^D_{t-1}(y) = \Gamma_t(y, x) z^D_{t-1}(y)$$

for $y \in \mathcal{S}_{t-1}$, while the terms outside $\mathcal{S}_{t-1}$ vanish. Substitution into (C.52) gives (C.53).

The Markov property converts the backward conditional expectation into a predecessor-state average, the conditional-expectation identity gives $m^D_t = u_t z^D_t$ on occupied states, and the same transition kernel transports the unnormalized credit mass forward without any Bayes denominator. □

Remark 6. *The normalized field z^D_t is defined only on occupied mean-field states, while the unnormalized mass m^D_t is defined on the full state space. The forward mass recursion* (C.52) *is therefore the convenient form for the streaming construction below.*

C.4 DOUBLESWEET STREAMING ALGORITHM

Explanation. The mean-field recursion (C.53) transports state-wise historical credit through population transition flow. A large finite population provides empirical analogues of both ingredients directly: the current empirical occupancy and the observed transition-pair frequencies. DoubleSWEET replaces the mean-field quantities by these empirical quantities and carries the resulting unnormalized credit mass forward. During the same scan of the N observed transitions, the learner can also accumulate the compressed critic update. Each transition therefore contributes once to the current critic increment, once to the destination count, and once to the destination old-credit accumulator. After the scan, the next empirical occupancy, next population Dutch operator, and next compressed credit mass are finalized together.

Formal construction. Define the occupied empirical support

$$\mathcal{S}_t^N := \{x \in \mathcal{X} : U_t^N(x) > 0\}, \qquad S_t^N := |\mathcal{S}_t^N|. \tag{C.63}$$

For the observed transition batch $t \to t+1$, define the empirical transition-pair law

$$\Gamma_{t+1}^N(y,x) := \frac{1}{N}\sum_{i=1}^N \mathbf{1}_{\{X_t^{i,N}=y,\ X_{t+1}^{i,N}=x\}}. \tag{C.64}$$

Its exact marginals are

$$\sum_{x\in\mathcal{X}} \Gamma_{t+1}^N(y,x) = U_t^N(y), \qquad \sum_{y\in\mathcal{X}} \Gamma_{t+1}^N(y,x) = U_{t+1}^N(x). \tag{C.65}$$

Initialize the compressed empirical credit mass by

$$\widehat{m}_0^{N,D}(x) := U_0^N(x)\phi_0(x). \tag{C.66}$$

On every occupied state $x \in \mathcal{S}_t^N$, define the state-wise DoubleSWEET readout

$$\widehat{z}_t^{N,D}(x) := \frac{\widehat{m}_t^{N,D}(x)}{U_t^N(x)}. \tag{C.67}$$

The empirical forward credit recursion is

$$\widehat{m}_{t+1}^{N,D}(x) := U_{t+1}^N(x)\phi_{t+1}(x) + \beta A_{t+1}^N \sum_{y\in\mathcal{S}_t^N} \Gamma_{t+1}^N(y,x)\widehat{z}_t^{N,D}(y), \tag{C.68}$$

where

$$A_{t+1}^N := I_d - \alpha \sum_{x\in\mathcal{X}} U_{t+1}^N(x)\phi_{t+1}(x)\phi_{t+1}(x)^\top. \tag{C.69}$$

The compressed critic is initialized by

$$w_{-1}^{N,C} = w_0^{N,C} = w_{\text{init}}. \tag{C.70}$$

For $0 \le t < T$, define

$$c_{t,i}^C := (w_t^{N,C} - w_{t-1}^{N,C})^\top \phi_t^{i,N}, \tag{C.71}$$

$$\delta_{t,i}^{\prime,C} := R_{t+1}^{i,N} + \gamma\chi_{t+1}(w_t^{N,C})^\top \phi_{t+1}^{i,N} - (w_{t-1}^{N,C})^\top \phi_t^{i,N}. \tag{C.72}$$

The DoubleSWEET critic update is

$$w_{t+1}^{N,C} := w_t^{N,C} + \frac{\alpha}{N}\sum_{i=1}^N \left[\delta_{t,i}^{\prime,C}\widehat{z}_t^{N,D}(X_t^{i,N}) - c_{t,i}^C\phi_t^{i,N}\right]. \tag{C.73}$$

Every observed predecessor $X_t^{i,N}$ belongs to $\mathcal{S}_t^N$, so the readout in (C.73) is well defined.

For each destination x, define the count and old-credit accumulator

$$n_{t+1}^N(x) := \sum_{i=1}^N \mathbf{1}_{\{X_{t+1}^{i,N}=x\}}, \tag{C.74}$$

$$\mathsf{Acc}_{t+1}^N(x) := \sum_{i=1}^N \mathbf{1}_{\{X_{t+1}^{i,N}=x\}}\widehat{z}_t^{N,D}(X_t^{i,N}). \tag{C.75}$$

Then

$$U_{t+1}^N(x) = \frac{n_{t+1}^N(x)}{N}, \qquad \sum_{y \in \mathcal{S}_t^N} \Gamma_{t+1}^N(y,x)\widehat{z}_t^{N,D}(y) = \frac{1}{N}\mathsf{Acc}_{t+1}^N(x), \tag{C.76}$$

so (C.68) is equivalently

$$\widehat{m}_{t+1}^{N,D}(x) = U_{t+1}^N(x)\phi_{t+1}(x) + \frac{\beta}{N}A_{t+1}^N\mathsf{Acc}_{t+1}^N(x). \tag{C.77}$$

Algorithm 2 DoubleSWEET: one-pass compressed Dutch evaluator

Initialize $w_{-1} = w_0 = w_{\text{init}}$, compute U_0^N, cache $\phi_0(x)$ on $\mathcal{S}_0^N$, and set $\widehat{m}_0^{N,D}(x) = U_0^N(x)\phi_0(x)$.
for $t = 0, \dots, T-1$ **do**
  Treat U_t^N, $\widehat{m}_t^{N,D}$, the current feature cache, w_{t-1}, and w_t as read-only during the transition scan.
  Initialize sparse destination counts $n[x] = 0$, credit accumulators $a[x] = 0 \in \mathbb{R}^d$, an empty next-feature cache, and $\Delta w = 0 \in \mathbb{R}^d$.
  for each observed record $(y, R, x) = (X_t^{i,N}, R_{t+1}^{i,N}, X_{t+1}^{i,N})$ **do**
    If x is new, evaluate and cache $\phi_{t+1}(x)$ once.
    $z \leftarrow \widehat{m}_t^{N,D}(y)/U_t^N(y)$, $p \leftarrow \phi_t(y)$ from the current cache, and $q \leftarrow \phi_{t+1}(x)$ from the next cache.
    $\delta' \leftarrow R + \gamma\chi_{t+1}w_t^\top q - w_{t-1}^\top p$ and $c \leftarrow (w_t - w_{t-1})^\top p$.
    $\Delta w \leftarrow \Delta w + \frac{\alpha}{N}(\delta' z - cp)$.
    $n[x] \leftarrow n[x] + 1$ and $a[x] \leftarrow a[x] + z$.
  end for
  Set $U_{t+1}^N(x) \leftarrow n[x]/N$ on the occupied destination states.
  Form A_{t+1}^N by (C.69) using the next-feature cache.
  Set $\widehat{m}_{t+1}^{N,D}(x) \leftarrow U_{t+1}^N(x)\phi_{t+1}(x) + \frac{\beta}{N}A_{t+1}^N a[x]$.
  Commit simultaneously $(w_{t-1}, w_t) \leftarrow (w_t, w_t + \Delta w)$ and advance $(U_t^N, \widehat{m}_t^{N,D}, \phi_t)$ to the next cached state.
end for

In the sparse representation, any state absent from the destination map has $U_{t+1}^N(x) = 0$ and $\widehat{m}_{t+1}^{N,D}(x) = 0$. Hence Algorithm 2 represents the full-state fields even though only occupied states are materialized.

Exact one-pass realization. For any realized transition batch, one complete pass of Algorithm 2 exactly realizes the formal DoubleSWEET critic update and empirical credit recursion:

$$\begin{aligned} w_{t+1}^{N,C} &= w_t^{N,C} + \frac{\alpha}{N}\sum_{i=1}^N \left[\delta_{t,i}^{\prime,C}\widehat{z}_t^{N,D}(X_t^{i,N}) - c_{t,i}^C\phi_t^{i,N}\right], \\ \widehat{m}_{t+1}^{N,D}(x) &= U_{t+1}^N(x)\phi_{t+1}(x) + \beta A_{t+1}^N \sum_{y \in \mathcal{S}_t^N} \Gamma_{t+1}^N(y,x)\widehat{z}_t^{N,D}(y), \qquad x \in \mathcal{X}. \end{aligned} \tag{C.78}$$

For every destination x,

$$\sum_{y \in \mathcal{S}_t^N} \Gamma_{t+1}^N(y,x)\widehat{z}_t^{N,D}(y) = \frac{1}{N}\sum_{i=1}^N \mathbf{1}_{\{X_{t+1}^{i,N}=x\}}\widehat{z}_t^{N,D}(X_t^{i,N}) \tag{C.79}$$

$$= \frac{1}{N}\mathsf{Acc}_{t+1}^N(x), \tag{C.80}$$

while $n_{t+1}^N(x)/N = U_{t+1}^N(x)$. Thus the destination accumulators reproduce (C.68), and the critic accumulator reproduces (C.73). Because all within-slice scan updates are additive and the current fields are read-only, the exact-arithmetic output is independent of the order of the transition records. No additional approximation is introduced by the one-pass implementation.

Let c_ϕ denote the cost of evaluating one time-state feature. With sparse state maps and dense vector/matrix arithmetic, the end-to-end one-step learner-side arithmetic cost is

$$O\big(Nd + S_{t+1}^N c_\phi + S_{t+1}^N d^2\big)\,. \tag{C.81}$$

If $c_\phi = O(d)$, this becomes $O(Nd + S_{t+1}^N d^2) \le O(Nd + Sd^2)$. This per-step bound uses the persistent current feature cache and evaluates each newly occupied destination feature once; initialization requires $O(S_0^N c_\phi)$ feature-evaluation work. The peak learner-side storage, including the current/next state-wise feature and credit caches, is

$$O\big((S_t^N + S_{t+1}^N)d + d^2\big) \le O(Sd + d^2). \tag{C.82}$$

D FINITE-POPULATION CONSISTENCY AND MEAN-FIELD GUARANTEES FOR DOUBLESWEET

D.1 FIXED-WINDOW COMPARISON SETTING

Appendix D compares the finite-population DoubleSWEET and Private Population Dutch constructions with a deterministic mean-field reference on a fixed time window. This section collects only the comparison objects and norms used throughout the appendix. Assumptions are introduced locally immediately before the formal statements that require them.

Finite-population physical quantities. For $0 \le t \le T$, the random empirical population law is

$$U_t^N(x) := \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}_{\{X_t^{i,N}=x\}}, \qquad x \in \mathcal{X}. \tag{D.1}$$

For $1 \le t \le T$, define the empirical transition-pair law

$$\Gamma_t^N(y,x) := \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}_{\{X_{t-1}^{i,N}=y,\, X_t^{i,N}=x\}}, \qquad y,x \in \mathcal{X}. \tag{D.2}$$

The empirical feature second moment and Dutch operator are

$$C_t^N := \sum_{x\in\mathcal{X}} U_t^N(x)\phi_t(x)\phi_t(x)^\top, \qquad A_t^N := I_d - \alpha C_t^N. \tag{D.3}$$

Here U_t^N, Γ_t^N, C_t^N, and A_t^N are random because they are constructed from the interacting finite population.

Mean-field physical quantities. Let u_0 be the deterministic initial population law. Under the fixed policy, define the closed-loop mean-field kernel and population flow by

$$K_t(x' \mid x) := P_t(x' \mid x, u_t), \qquad u_{t+1}(x') := \sum_{x\in\mathcal{X}} u_t(x)K_t(x' \mid x). \tag{D.4}$$

The representative process satisfies

$$\bar{X}_0 \sim u_0, \qquad \mathcal{L}(\bar{X}_{t+1} \mid \bar{X}_0, \dots, \bar{X}_t) = K_t(\cdot \mid \bar{X}_t). \tag{D.5}$$

For $1 \le t \le T$, its transition-pair law is

$$\Gamma_t(y,x) := \mathbb{P}(\bar{X}_{t-1}=y, \bar{X}_t = x) = u_{t-1}(y)K_{t-1}(x \mid y). \tag{D.6}$$

The deterministic feature second moment and Dutch operator are

$$C_t := \sum_{x\in\mathcal{X}} u_t(x)\phi_t(x)\phi_t(x)^\top, \qquad A_t := I_d - \alpha C_t. \tag{D.7}$$

The quantities u_t, Γ_t, C_t, and A_t are deterministic once u_0, the fixed policy, and the closed-loop kernels are fixed.

Credit-mass quantities. The DoubleSWEET compressed credit mass is denoted by $\widehat{m}_t^{N,D}$. The state-aggregated Private Dutch credit mass is

$$m_t^{\mathrm{priv},N,D}(x) := \frac{1}{N}\sum_{i=1}^{N} e_t^{i,N,D}\mathbf{1}_{\{X_t^{i,N}=x\}}. \tag{D.8}$$

The Dutch quantities use the fixed algorithmic scalars $\alpha > 0$ and $\beta := \gamma\lambda$. For the deterministic mean-field reference, let

$$\bar{e}_{-1}^D = 0, \qquad \bar{e}_t^D = \phi_t(\bar{X}_t) + \beta A_t \bar{e}_{t-1}^D, \tag{D.9}$$

and define

$$m_t^D(x) := \mathbb{E}\big[\bar{e}_t^D \mathbf{1}_{\{\bar{X}_t=x\}}\big]. \tag{D.10}$$

Thus $\widehat{m}_t^{N,D}$ and $m_t^{\mathrm{priv},N,D}$ are random finite-population vector fields, whereas m_t^D is deterministic.

Learner parameters. We use

$$w_t^{N,C}, \qquad w_t^{N,D}, \qquad w_t^{MF} \tag{D.11}$$

for the shared critic parameters produced by DoubleSWEET, Private Population Dutch, and the deterministic mean-field learner, respectively. The superscript C on $w_t^{N,C}$ means *compressed*; it is distinct from the second-moment matrices C_t^N and C_t. The corresponding update maps are defined locally in Sec. D.4.

Norms. For a vector field $f : \mathcal{X} \to \mathbb{R}^d$ and a signed table $q : \mathcal{X}^2 \to \mathbb{R}$, define

$$\|f\|_{1,2} := \sum_{x\in\mathcal{X}} \|f(x)\|_2, \qquad \|q\|_1 := \sum_{y,x\in\mathcal{X}} |q(y,x)|. \tag{D.12}$$

Matrix norms $\|\cdot\|_{\mathrm{op}}$ are Euclidean operator norms.

Fixed-window convention. The horizon $T < \infty$ is fixed throughout the finite-population consistency analysis. Constants carrying a subscript T may depend on T, the fixed model parameters, and the fixed state space, but not on N. For any nonnegative random quantities $Z_0, \dots, Z_T$,

$$\max_{0\le t\le T} \mathbb{E}Z_t \le \mathbb{E}\left[\max_{0\le t\le T} Z_t\right] \le (T+1) \max_{0\le t\le T} \mathbb{E}Z_t. \tag{D.13}$$

D.2 Physical-Flow and Dutch-Operator Consistency Lemma

Explanation. The compressed and mean-field credit recursions are driven by different physical population objects. In the finite population, the current occupancy, realized transition-pair frequencies, feature second moment, and Dutch operator fluctuate with the interacting particle system. Their mean-field counterparts are deterministic. We first control these physical discrepancies on a fixed horizon. The proof couples every interacting particle with an independent nonlinear copy following the mean-field law, transfers the particle mismatch to empirical state and transition-pair histograms, and then pushes the state-law bound through the feature second moment and the Dutch operator.

Formal construction. Let $\mathcal{F}_t^N$ contain the finite-population states up to time t and all transition randomness revealed before the fresh step $t \to t+1$. Enlarge the probability space with nonlinear copies $\bar{X}^1, \dots, \bar{X}^N$ and define

$$\mathcal{G}_t^N := \mathcal{F}_t^N \vee \sigma\big(\bar{X}_s^i : 0 \le s \le t,\ 1 \le i \le N\big). \tag{D.14}$$

For the coupling error, set

$$I_t^{i,N} := \mathbf{1}_{\{X_t^{i,N} \ne \bar{X}_t^i\}}, \qquad a_t^N := \frac{1}{N}\sum_{i=1}^N \mathbb{E}I_t^{i,N}. \tag{D.15}$$

Define the nonlinear-copy empirical state and transition-pair laws by

$$\bar{U}_t^N := \frac{1}{N}\sum_{i=1}^N \delta_{\bar{X}_t^i}, \tag{D.16}$$

$$\bar{\Gamma}_t^N := \frac{1}{N}\sum_{i=1}^N \delta_{(\bar{X}_{t-1}^i, \bar{X}_t^i)}, \qquad 1 \le t \le T. \tag{D.17}$$

Setting D-P1 (Finite fixed-policy population system). The physical state space and time horizon are finite. A single fixed policy is evaluated throughout the window, so the closed-loop transition kernels are fixed over the comparison window.

Mathematical form of D-P1.

$$|\mathcal{X}| = S < \infty, \qquad T < \infty, \qquad P_t : \mathcal{X} \times \mathcal{P}(\mathcal{X}) \to \mathcal{P}(\mathcal{X}), \quad 0 \le t < T. \tag{D.18}$$

Assumption D-P2 (Consistent iid initialization). The finite population starts from independent samples of the same deterministic initial law that initializes the representative mean-field system.

Mathematical form of D-P2.

$$X_0^{1,N}, \dots, X_0^{N,N} \overset{\text{iid}}{\sim} u_0, \qquad u_0 \in \mathcal{P}(\mathcal{X}). \tag{D.19}$$

Assumption D-P3 (Idiosyncratic transition noise without common noise). Conditioned on the current finite-population information, different agents draw their next states independently from their state- and population-dependent closed-loop kernels. No additional shared transition noise drives the whole population.

Mathematical form of D-P3.

$$\mathcal{L}\Big((X_{t+1}^{i,N})_{i=1}^N \mid \mathcal{F}_t^N\Big) = \bigotimes_{i=1}^N P_t\Big(\cdot \mid X_t^{i,N}, U_t^N\Big), \qquad 0 \le t < T. \tag{D.20}$$

Assumption D-P4 (Population regularity of the transition kernel). At a fixed individual state, changing the population distribution changes the next-state law in a Lipschitz-continuous way, uniformly over the finite time window.

Mathematical form of D-P4.

$$\|P_t(\cdot \mid x, u) - P_t(\cdot \mid x, v)\|_{\mathrm{TV}} \le L_P \|u - v\|_1, \qquad {}^{0\le t<T,\ x\in\mathcal{X},}_{u,v\in\mathcal{P}(\mathcal{X})}. \tag{D.21}$$

Derived finite-domain feature bound D-P5. The finite-population and mean-field constructions use the same time-state feature map. Because D-P1 fixes a finite state-time domain, no separate bounded-feature assumption is needed; define

$$B_\phi := \max_{0\le t\le T} \max_{x\in\mathcal{X}} \|\phi_t(x)\|_2 < \infty. \tag{D.22}$$

Lemma 3 (Fixed-Window Physical-Flow and Dutch-Operator Consistency)**.** *Under Setting D-P1, Assumptions D-P2, D-P3, and D-P4, with the derived finite-domain bound D-P5, the empirical population law, transition-pair law, feature second moment, and Dutch operator approach their deterministic mean-field counterparts at the inverse-square-root population rate on every fixed horizon; under the same coupling, the average particle mismatch has the same rate and the nonlinear-copy trajectories are iid representative mean-field trajectories, with finite constants independent of the population size.*

$$\begin{aligned}
\max_{0\le t\le T} a_t^N &\le \frac{C_{a,T}}{\sqrt{N}},\\
\big((\bar X_0^i, \dots, \bar X_T^i)\big)_{i=1}^N &\overset{\text{iid}}{\sim} \mathcal{L}(\bar X_0, \dots, \bar X_T),\\
\max_{0\le t\le T} \mathbb{E}\|U_t^N - u_t\|_1 &\le \frac{C_{U,T}}{\sqrt{N}},\\
\max_{1\le t\le T} \mathbb{E}\|\Gamma_t^N - \Gamma_t\|_1 &\le \frac{C_{\Gamma,T}}{\sqrt{N}},\\
\max_{0\le t\le T} \mathbb{E}\|C_t^N - C_t\|_{\mathrm{op}} &\le \frac{C_{C,T}}{\sqrt{N}},\\
\max_{0\le t\le T} \mathbb{E}\|A_t^N - A_t\|_{\mathrm{op}} &\le \frac{C_{A,T}}{\sqrt{N}}.
\end{aligned} \tag{D.23}$$

Proof. The finite population is coupled with nonlinear copies that follow the deterministic mean-field law. A one-step maximal-coupling estimate turns the population-law discrepancy into a recurrence for the average particle mismatch. Finite-alphabet empirical-histogram bounds then control

the state and transition-pair laws. Because the feature map depends only on time and state, the second-moment and Dutch-operator errors follow directly from the state-law error.

Step 1: two finite-alphabet bounds. Let $Y_1, \dots, Y_N$ be iid on a finite alphabet E with $|E| = M$, law p, and empirical law $\widehat{p}_N$. Then

$$\mathbb{E}\|\widehat{p}_N - p\|_1 \le \sum_{a\in E} \sqrt{\mathrm{Var}(\widehat{p}_N(a))} \le \frac{1}{\sqrt{N}} \sum_{a\in E} \sqrt{p(a)} \le \sqrt{\frac{M}{N}}. \tag{D.24}$$

For arbitrary $x_i, y_i \in E$,

$$\left\| \frac{1}{N}\sum_{i=1}^N \delta_{x_i} - \frac{1}{N}\sum_{i=1}^N \delta_{y_i} \right\|_1 \le \frac{2}{N}\sum_{i=1}^N \mathbf{1}_{\{x_i \neq y_i\}}. \tag{D.25}$$

Step 2: nonlinear-copy coupling and iid preservation. Choose $\bar{X}_0^i = X_0^{i,N}$. For $0 \le t < T$, conditionally on $\mathcal{G}_t^N$, couple

$$X_{t+1}^{i,N} \sim P_t(\cdot \mid X_t^{i,N}, U_t^N), \qquad \bar{X}_{t+1}^i \sim K_t(\cdot \mid \bar{X}_t^i) \tag{D.26}$$

independently across i. By maximal coupling and (D.21),

$$\mathbb{P}(I_{t+1}^{i,N} = 1 \mid \mathcal{G}_t^N) \le I_t^{i,N} + L_P \|U_t^N - u_t\|_1. \tag{D.27}$$

Because the couplings are conditionally independent across agents and each nonlinear-copy marginal is $K_t(\cdot \mid \bar{X}_t^i)$,

$$\mathcal{L}\big(\bar{X}_{t+1}^{1:N} \mid \mathcal{G}_t^N\big) = \bigotimes_{i=1}^N K_t(\cdot \mid \bar{X}_t^i). \tag{D.28}$$

The right-hand side is measurable with respect to $\sigma(\bar{X}_t^{1:N})$. Therefore, the tower property gives

$$\mathbb{P}\big(\bar{X}_{t+1}^{1:N} = x'_{1:N} \mid \bar{X}_t^{1:N}\big) = \prod_{i=1}^N K_t(x'_i \mid \bar{X}_t^i). \tag{D.29}$$

Thus, by induction,

$$(\bar{X}_0^i, \dots, \bar{X}_T^i), \qquad i = 1, \dots, N, \tag{D.30}$$

are iid copies of the representative mean-field trajectory.

Step 3: population-law consistency. By (D.24) on $\mathcal{X}$ and (D.25),

$$\mathbb{E}\|U_t^N - u_t\|_1 \le 2a_t^N + \sqrt{\frac{S}{N}}. \tag{D.31}$$

From (D.27),

$$a_{t+1}^N \le (1 + 2L_P) a_t^N + L_P\sqrt{\frac{S}{N}}, \qquad a_0^N = 0. \tag{D.32}$$

Set

$$L := 1 + 2L_P, \qquad c := L_P\sqrt{S}, \qquad C_{a,T} := c \max_{0\le t\le T} \sum_{r=0}^{t-1} L^r, \tag{D.33}$$

where the sum is zero at $t = 0$. Hence

$$\max_{0\le t\le T} a_t^N \le \frac{C_{a,T}}{\sqrt{N}}. \tag{D.34}$$

With

$$C_{U,T} := 2C_{a,T} + \sqrt{S}, \tag{D.35}$$

Eq. (D.31) gives

$$\max_{0\le t\le T} \mathbb{E}\|U_t^N - u_t\|_1 \le \frac{C_{U,T}}{\sqrt{N}}. \tag{D.36}$$

Step 4: transition-pair consistency. By (D.30) and (D.24) on $\mathcal{X}^2$,

$$\mathbb{E}\|\bar{\Gamma}_t^N - \Gamma_t\|_1 \le \frac{S}{\sqrt{N}}. \tag{D.37}$$

Since

$$\mathbf{1}_{\{(X_{t-1}^{i,N}, X_t^{i,N}) \neq (\bar{X}_{t-1}^i, \bar{X}_t^i)\}} \le I_{t-1}^{i,N} + I_t^{i,N}, \tag{D.38}$$

Eq. (D.25) yields

$$\mathbb{E}\|\Gamma_t^N - \bar{\Gamma}_t^N\|_1 \le 2(a_{t-1}^N + a_t^N). \tag{D.39}$$

Therefore, with

$$C_{\Gamma,T} := 4C_{a,T} + S, \tag{D.40}$$

$$\max_{1\le t\le T} \mathbb{E}\|\Gamma_t^N - \Gamma_t\|_1 \le \frac{C_{\Gamma,T}}{\sqrt{N}}. \tag{D.41}$$

Step 5: feature-second-moment consistency. By (D.3), (D.7), and (D.22),

$$\|C_t^N - C_t\|_{\mathrm{op}} \le \sum_{x\in\mathcal{X}} |U_t^N(x) - u_t(x)| \|\phi_t(x)\phi_t(x)^\top\|_{\mathrm{op}} \tag{D.42}$$

$$\le B_\phi^2 \|U_t^N - u_t\|_1. \tag{D.43}$$

Thus, with

$$C_{C,T} := B_\phi^2 C_{U,T}, \tag{D.44}$$

$$\max_{0\le t\le T} \mathbb{E}\|C_t^N - C_t\|_{\mathrm{op}} \le \frac{C_{C,T}}{\sqrt{N}}. \tag{D.45}$$

Step 6: Dutch-operator consistency.

$$A_t^N - A_t = -\alpha(C_t^N - C_t), \qquad \|A_t^N - A_t\|_{\mathrm{op}} = \alpha\|C_t^N - C_t\|_{\mathrm{op}}. \tag{D.46}$$

With

$$C_{A,T} := \alpha C_{C,T}, \tag{D.47}$$

Eq. (D.45) gives

$$\max_{0\le t\le T} \mathbb{E}\|A_t^N - A_t\|_{\mathrm{op}} \le \frac{C_{A,T}}{\sqrt{N}}. \tag{D.48}$$

Equations (D.30), (D.34), (D.36), (D.41), (D.45), and (D.48) give (D.23).

The coupling reduces the population error to an empirical histogram fluctuation and a recursively propagated mismatch. The resulting state-law bound then controls the transition-pair law, feature moments, and Dutch operator. □

Remark 7. *Lemma 3 is a fixed-state-space, fixed-window expected-norm result. The constants may grow with T and with the fixed alphabet size S. It does not assert a deterministic pathwise $N^{-1/2}$ rate or a bound uniform as $T \to \infty$. Under common noise, the natural analogue is conditional on the common noise, because the nonlinear copies are then generally conditionally iid rather than unconditionally iid.*

D.3 Finite-Population Credit-Mass Consistency Theorem 2

Explanation. The physical consistency result in Sec. D.2 controls the finite-population quantities that drive the Dutch credit recursions. We first propagate those errors to a deterministic mean-field credit mass, because those stronger one-sided bounds are useful later. For Theorem 2 itself, however, no mean-field detour is needed: subtracting the compressed and state-aggregated private recursions gives a direct finite-population credit-gap recursion. Its only stochastic term is a within-state centered fresh-transition fluctuation, so the theorem requires fresh idiosyncratic transition noise but not iid initialization or population-Lipschitz regularity.

Formal construction. For every $0 \le t \le T$, define the occupied supports

$$\mathcal{S}_t^N := \{x \in \mathcal{X} : U_t^N(x) > 0\}, \qquad \mathcal{S}_t := \{x \in \mathcal{X} : u_t(x) > 0\}. \tag{D.49}$$

Use the full-state zero extensions

$$\widehat{z}_t^{N,D,0}(x) := \begin{cases} \widehat{m}_t^{N,D}(x)/U_t^N(x), & U_t^N(x) > 0, \\ 0, & U_t^N(x) = 0, \end{cases} \qquad z_t^{D,0}(x) := \begin{cases} m_t^D(x)/u_t(x), & u_t(x) > 0, \\ 0, & u_t(x) = 0. \end{cases} \tag{D.50}$$

For $1 \le t \le T$, define the old-credit transport fields

$$\mathcal{T}_t^N(x) := \sum_{y \in \mathcal{X}} \Gamma_t^N(y,x)\widehat{z}_{t-1}^{N,D,0}(y), \qquad \mathcal{T}_t(x) := \sum_{y \in \mathcal{X}} \Gamma_t(y,x) z_{t-1}^{D,0}(y). \tag{D.51}$$

The compressed and mean-field masses are initialized and propagated by

$$\begin{aligned} \widehat{m}_0^{N,D}(x) &= U_0^N(x)\phi_0(x), \quad \widehat{m}_t^{N,D}(x) = U_t^N(x)\phi_t(x) + \beta A_t^N \mathcal{T}_t^N(x), \\ m_0^D(x) &= u_0(x)\phi_0(x), \qquad m_t^D(x) = u_t(x)\phi_t(x) + \beta A_t \mathcal{T}_t(x). \end{aligned} \tag{D.52}$$

The finite-population private trace used by Eq. (D.8) satisfies

$$e_{-1}^{i,N,D} = 0, \qquad e_t^{i,N,D} := \phi_t(X_t^{i,N}) + \beta A_t^N e_{t-1}^{i,N,D}. \tag{D.53}$$

For each nonlinear copy from Sec. D.2, define

$$\bar{e}_{-1}^{i,D} = 0, \qquad \bar{e}_t^{i,D} := \phi_t(\bar{X}_t^i) + \beta A_t \bar{e}_{t-1}^{i,D}, \tag{D.54}$$

and the nonlinear-copy empirical mass

$$\bar{m}_t^N(x) := \frac{1}{N} \sum_{i=1}^N \bar{e}_t^{i,D} \mathbf{1}_{\{\bar{X}_t^i = x\}}. \tag{D.55}$$

The Private Dutch mass is the quantity already defined in Eq. (D.8).

Fixed-window setting. The state space and horizon are finite, one fixed policy generates the physical trajectory, and

$$|\mathcal{X}| = S < \infty, \qquad T < \infty, \qquad P_t : \mathcal{X} \times \mathcal{P}(\mathcal{X}) \to \mathcal{P}(\mathcal{X}), \quad 0 \le t < T.$$

Assumption D-C1 (Consistent iid initialization). The finite agents start independently from the same deterministic initial law as the mean-field representative process.

Mathematical form of D-C1.

$$X_0^{1,N}, \ldots, X_0^{N,N} \overset{\text{iid}}{\sim} u_0. \tag{D.56}$$

Assumption D-C2 (Idiosyncratic transition noise without common noise). Conditioned on the current finite-population information, the agents draw their fresh next states independently from the prescribed state- and population-dependent kernels, with no additional common transition noise.

Mathematical form of D-C2.

$$\mathcal{L}\Big((X_{t+1}^{i,N})_{i=1}^N \mid \mathcal{F}_t^N\Big) = \bigotimes_{i=1}^N P_t\Big(\cdot \mid X_t^{i,N}, U_t^N\Big), \qquad 0 \le t < T. \tag{D.57}$$

Assumption D-C3 (Population regularity of the transition kernel). At a fixed physical state, the closed-loop transition kernel is uniformly Lipschitz continuous in the population distribution over the fixed horizon.

Mathematical form of D-C3.

$$\|P_t(\cdot \mid x,u) - P_t(\cdot \mid x,v)\|_{\mathrm{TV}} \le L_P \|u - v\|_1, \qquad {}^{0 \le t < T,\ x \in \mathcal{X},}_{u,v \in \mathcal{P}(\mathcal{X})}. \tag{D.58}$$

Algorithmic setting D-C4 (Shared features and fixed Dutch parameters). The finite and mean-field constructions use the same time-state feature map. Since the state-time domain is finite, define the induced bound

$$B_\phi := \max_{0\le t\le T}\max_{x\in\mathcal{X}}\|\phi_t(x)\|_2 < \infty.$$

The Dutch parameters are fixed throughout the episode.

Mathematical form of D-C4.

$$\alpha > 0, \qquad 0 \le \gamma \le 1, \qquad 0 \le \lambda \le 1, \qquad \beta := \gamma\lambda. \tag{D.59}$$

Lemma 4 (Common Mean-Field Credit Consistency)**.** *In the fixed-window setting above, under Assumptions D-C1, D-C2, and D-C3, together with algorithmic setting D-C4, the DoubleSWEET empirical credit mass and the state-aggregated Private Dutch credit mass both approach the same deterministic mean-field Dutch credit mass at the inverse-square-root population rate on every fixed horizon, with finite constants independent of the population size.*

$$\begin{aligned}\max_{0\le t\le T}\mathbb{E}\|\widehat{m}_t^{N,D} - m_t^D\|_{1,2} &\le \frac{C_{m,T}}{\sqrt{N}},\\ \max_{0\le t\le T}\mathbb{E}\|m_t^{\mathrm{priv},N,D} - m_t^D\|_{1,2} &\le \frac{C_{\mathrm{priv},T}}{\sqrt{N}},\end{aligned} \tag{D.60}$$

Proof. The proof has two branches with one common target. For DoubleSWEET, bounded state-wise readouts yield a denominator-free transport estimate; the physical errors from Lemma 3 then give a finite-horizon recursion for the compressed credit-mass error. For Private Dutch, the same nonlinear-copy coupling controls the additional trace coordinate, after which an iid empirical-mass bound controls the sampling error of the nonlinear-copy credit mass.

The fixed-window setting together with D-C1–D-C3 supplies the initialization, fresh-noise, and population-regularity conditions of D-P1, D-P2, D-P3, and D-P4, while D-C4 supplies the same shared feature map and fixed Dutch parameters; the feature bound D-P5 is automatic on the finite state-time domain. Hence Lemma 3 applies throughout this proof.

Step 1: fixed-window Dutch envelopes. Define

$$B_A := 1 + \alpha B_\phi^2, \qquad B_{z,T} := B_\phi \sum_{k=0}^{T}(\beta B_A)^k, \qquad B_{e,T} := B_{z,T}. \tag{D.61}$$

Then

$$\max_{0\le t\le T}\{\|A_t^N\|_{\mathrm{op}}, \|A_t\|_{\mathrm{op}}\} \le B_A. \tag{D.62}$$

The private and nonlinear-copy trace recursions give

$$\max_{0\le t\le T}\max_{1\le i\le N}\left\{\|e_t^{i,N,D}\|_2, \|\bar{e}_t^{i,D}\|_2\right\} \le B_{e,T}. \tag{D.63}$$

For $x \in \mathcal{S}_t^N$, set

$$\widehat{B}_t^N(y \mid x) := \frac{\Gamma_t^N(y,x)}{U_t^N(x)}. \tag{D.64}$$

Using the exact destination marginal and Eq. (D.52),

$$\widehat{z}_t^{N,D}(x) = \phi_t(x) + \beta A_t^N \sum_{y\in\mathcal{X}} \widehat{B}_t^N(y \mid x)\widehat{z}_{t-1}^{N,D,0}(y). \tag{D.65}$$

Hence

$$\|\widehat{z}_t^{N,D}\|_{\infty,\mathrm{occ}} \le B_\phi + \beta B_A \|\widehat{z}_{t-1}^{N,D}\|_{\infty,\mathrm{occ}}. \tag{D.66}$$

For every $x \in \mathcal{S}_t$, conditional Jensen and Eq. (D.63) give

$$\|z_t^{D,0}(x)\|_2 \le \mathbb{E}[\|\bar{e}_t^D\|_2 \mid \bar{X}_t = x] \le B_{e,T} = B_{z,T}. \tag{D.67}$$

Therefore

$$\max_{0\le t\le T}\left\{\|\widehat{z}_t^{N,D}\|_{\infty,\mathrm{occ}},\|z_t^{D,0}\|_\infty\right\}\le B_{z,T}. \tag{D.68}$$

By the predecessor marginals,

$$\max_{1\le t\le T}\left\{\|\mathcal{T}_t^N\|_{1,2},\|\mathcal{T}_t\|_{1,2}\right\}\le B_{z,T}. \tag{D.69}$$

Step 2: denominator-free transport stability. For $1\le t\le T$, add and subtract $\sum_y \Gamma_t^N(y,x)z_{t-1}^{D,0}(y)$ in Eq. (D.51). Then

$$\|\mathcal{T}_t^N-\mathcal{T}_t\|_{1,2}\le\sum_y U_{t-1}^N(y)\|\widehat{z}_{t-1}^{N,D,0}(y)-z_{t-1}^{D,0}(y)\|_2+B_{z,T}\|\Gamma_t^N-\Gamma_t\|_1 \tag{D.70}$$

$$\le\|\widehat{m}_{t-1}^{N,D}-m_{t-1}^D\|_{1,2}+B_{z,T}\|U_{t-1}^N-u_{t-1}\|_1+B_{z,T}\|\Gamma_t^N-\Gamma_t\|_1. \tag{D.71}$$

Step 3: DoubleSWEET-to-mean-field credit consistency. Set

$$\Delta_t^{m,N}:=\|\widehat{m}_t^{N,D}-m_t^D\|_{1,2},\qquad d_t^{m,N}:=\mathbb{E}\Delta_t^{m,N}. \tag{D.72}$$

From Eq. (D.52),

$$\Delta_t^{m,N}\le B_\phi\|U_t^N-u_t\|_1+\beta B_A\|\mathcal{T}_t^N-\mathcal{T}_t\|_{1,2}+\beta B_{z,T}\|A_t^N-A_t\|_{\mathrm{op}} \tag{D.73}$$

$$\le\beta B_A\Delta_{t-1}^{m,N}+\rho_t^N, \tag{D.74}$$

where

$$\rho_t^N:=B_\phi\|U_t^N-u_t\|_1+\beta B_AB_{z,T}\|U_{t-1}^N-u_{t-1}\|_1+\beta B_AB_{z,T}\|\Gamma_t^N-\Gamma_t\|_1+\beta B_{z,T}\|A_t^N-A_t\|_{\mathrm{op}}. \tag{D.75}$$

Moreover,

$$\Delta_0^{m,N}\le B_\phi\|U_0^N-u_0\|_1. \tag{D.76}$$

By Lemma 3, with

$$L_m:=\beta B_A,\qquad D_T:=B_\phi C_{U,T}+\beta B_AB_{z,T}(C_{U,T}+C_{\Gamma,T})+\beta B_{z,T}C_{A,T}, \tag{D.77}$$

we obtain

$$d_t^{m,N}\le L_md_{t-1}^{m,N}+\frac{D_T}{\sqrt{N}}. \tag{D.78}$$

Thus

$$C_{m,T}:=\max_{0\le t\le T}\left(L_m^tB_\phi C_{U,T}+D_T\sum_{k=0}^{t-1}L_m^k\right)<\infty, \tag{D.79}$$

with the sum zero at $t=0$, and

$$\max_{0\le t\le T}\mathbb{E}\|\widehat{m}_t^{N,D}-m_t^D\|_{1,2}\le\frac{C_{m,T}}{\sqrt{N}}. \tag{D.80}$$

Step 4: coupled private-trace consistency. By Eqs. (D.53) and (D.54),

$$e_t^{i,N,D}-\bar{e}_t^{i,D}=\phi_t(X_t^{i,N})-\phi_t(\bar{X}_t^i)+\beta A_t^N(e_{t-1}^{i,N,D}-\bar{e}_{t-1}^{i,D}) \tag{D.81}$$

$$+\beta(A_t^N-A_t)\bar{e}_{t-1}^{i,D}. \tag{D.82}$$

Define

$$b_t^N:=\frac{1}{N}\sum_{i=1}^N\mathbb{E}\|e_t^{i,N,D}-\bar{e}_t^{i,D}\|_2. \tag{D.83}$$

Using Eq. (D.62), the trace envelope $B_{e,T}$, and

$$\|\phi_t(X_t^{i,N})-\phi_t(\bar{X}_t^i)\|_2\le 2B_\phi I_t^{i,N}, \tag{D.84}$$

we have

$$b_t^N\le\beta B_Ab_{t-1}^N+2B_\phi a_t^N+\beta B_{e,T}\mathbb{E}\|A_t^N-A_t\|_{\mathrm{op}}. \tag{D.85}$$

By Lemma 3, $b_0^N = 0$ and

$$\max_{0\le t\le T} b_t^N \le \frac{C_{e,T}}{\sqrt{N}}, \tag{D.86}$$

where

$$C_{e,T} := (2B_\phi C_{a,T} + \beta B_{e,T} C_{A,T}) \max_{0\le t\le T} \sum_{k=0}^{t-1} (\beta B_A)^k. \tag{D.87}$$

Step 5: Private-Dutch-to-mean-field credit consistency. For each $x \in \mathcal{X}$,

$$m_t^{\mathrm{priv},N,D} - m_t^D = (m_t^{\mathrm{priv},N,D} - \bar{m}_t^N) + (\bar{m}_t^N - m_t^D). \tag{D.88}$$

For each copy index,

$$\left\| e_t^{i,N,D} \mathbf{1}_{\{X_t^{i,N}=\cdot\}} - \bar{e}_t^{i,D} \mathbf{1}_{\{\bar{X}_t^i=\cdot\}} \right\|_{1,2} \le \|e_t^{i,N,D} - \bar{e}_t^{i,D}\|_2 + 2B_{e,T} I_t^{i,N}. \tag{D.89}$$

Hence

$$\mathbb{E}\|m_t^{\mathrm{priv},N,D} - \bar{m}_t^N\|_{1,2} \le b_t^N + 2B_{e,T} a_t^N. \tag{D.90}$$

For

$$Y_{t,i}^x := \bar{e}_t^{i,D} \mathbf{1}_{\{\bar{X}_t^i = x\}}, \tag{D.91}$$

Lemma 3 implies that $(Y_{t,i}^x)_{i=1}^N$ are iid, with

$$\mathbb{E}Y_{t,i}^x = m_t^D(x), \qquad \mathbb{E}\|Y_{t,i}^x\|_2^2 \le B_{e,T}^2 u_t(x). \tag{D.92}$$

Independence and centering give, for every $x \in \mathcal{X}$,

$$\mathbb{E}\|\bar{m}_t^N(x) - m_t^D(x)\|_2^2 = \frac{1}{N}\mathbb{E}\|Y_{t,1}^x - m_t^D(x)\|_2^2 \le \frac{B_{e,T}^2 u_t(x)}{N}. \tag{D.93}$$

Hence

$$\mathbb{E}\|\bar{m}_t^N - m_t^D\|_{1,2} \le \frac{B_{e,T}}{\sqrt{N}} \sum_{x\in\mathcal{X}} \sqrt{u_t(x)} \le \frac{\sqrt{S}\, B_{e,T}}{\sqrt{N}}. \tag{D.94}$$

With

$$C_{\mathrm{priv},T} := C_{e,T} + 2B_{e,T} C_{a,T} + \sqrt{S}\, B_{e,T}, \tag{D.95}$$

Eqs. (D.86)–(D.94) give

$$\max_{0\le t\le T} \mathbb{E}\|m_t^{\mathrm{priv},N,D} - m_t^D\|_{1,2} \le \frac{C_{\mathrm{priv},T}}{\sqrt{N}}. \tag{D.96}$$

Equations (D.80) and (D.96) prove Eq. (D.60).

The DoubleSWEET branch propagates the physical errors through a mass recursion without occupancy division. The Private Dutch branch lifts the nonlinear-copy coupling to traces and then uses iid empirical averaging. Both reach the same mean-field credit mass. □

Fixed-window setting for Theorem 2. The theorem uses a finite state space and horizon and one fixed policy:

$$|\mathcal{X}| = S < \infty, \qquad T < \infty, \qquad P_t : \mathcal{X} \times \mathcal{P}(\mathcal{X}) \to \mathcal{P}(\mathcal{X}), \quad 0 \le t < T.$$

Assumption D-T2-A1 (Idiosyncratic transition noise without common noise). Given the current finite-population information, the fresh agent transitions are conditionally independent and no shared transition noise is present. This is the only stochastic assumption needed for the direct compressed/private credit comparison.

Mathematical form of D-T2-A1.

$$\mathcal{L}\Big((X_{t+1}^{i,N})_{i=1}^N \mid \mathcal{F}_t^N\Big) = \bigotimes_{i=1}^N P_t\Big(\cdot \mid X_t^{i,N}, U_t^N\Big), \qquad 0 \le t < T. \tag{D.97}$$

Algorithmic setting D-T2-A2 (State-only features and fixed Dutch parameters). The feature map depends only on time and physical state. Its bound is automatic on the finite state-time domain,

$$B_\phi := \max_{0\le t\le T}\max_{x\in\mathcal{X}} \|\phi_t(x)\|_2 < \infty,$$

and the Dutch parameters are fixed on the episode.

Mathematical form of D-T2-A2.

$$\alpha > 0, \qquad 0 \le \gamma \le 1, \qquad 0 \le \lambda \le 1, \qquad \beta = \gamma\lambda. \tag{D.98}$$

Theorem 2. *In the fixed-window setting above, under Assumption D-T2-A1 and algorithmic setting D-T2-A2, the DoubleSWEET credit mass and the state-aggregated Private Dutch credit mass differ by at most an inverse-square-root population rate over every fixed horizon, with no iid initialization or population-Lipschitz condition required for this direct finite-population comparison.*

$$C_{\mathrm{credit},T} < \infty, \qquad \max_{0\le t\le T} \mathbb{E}\|\widehat{m}_t^{N,D} - m_t^{\mathrm{priv},N,D}\|_{1,2} \le C_{\mathrm{credit},T} N^{-1/2}. \tag{D.99}$$

Proof. Subtract the two finite-population credit recursions directly. The transported old-credit discrepancy is nonexpansive in the unnormalized $\|\cdot\|_{1,2}$ mass norm up to the common Dutch operator. The remaining term is a within-current-state centered trace residual multiplied by fresh destination indicators. Assumption D-T2-A1 makes these destination draws conditionally independent, giving an $O(N^{-1/2})$ fluctuation without any comparison to a mean-field process.

Let

$$D_t^N(x) := \widehat{m}_t^{N,D}(x) - m_t^{\mathrm{priv},N,D}(x). \tag{D.100}$$

For occupied states define

$$z_t^{\mathrm{priv},N,0}(x) := \frac{m_t^{\mathrm{priv},N,D}(x)}{U_t^N(x)}, \tag{D.101}$$

and set it to zero on empty states. Because both constructions start from $U_0^N(x)\phi_0(x)$,

$$D_0^N = 0 \tag{D.102}$$

pathwise, for an arbitrary realized initial population.

For $1 \le t \le T$, write the within-state centered predecessor residual

$$\rho_{t-1,i} := z_{t-1}^{\mathrm{priv},N,0}(X_{t-1}^{i,N}) - e_{t-1}^{i,N,D}. \tag{D.103}$$

By definition of the within-state private average,

$$\sum_{i:X_{t-1}^{i,N}=y} \rho_{t-1,i} = 0, \qquad y \in \mathcal{X}. \tag{D.104}$$

Using Eq. (D.52) for the compressed branch and aggregating Eq. (D.53) over agents arriving at x gives the exact difference recursion

$$D_t^N(x) = \beta A_t^N\left[\sum_y \Gamma_t^N(y,x)\big(\widehat{z}_{t-1}^{N,D,0}(y) - z_{t-1}^{\mathrm{priv},N,0}(y)\big) + R_t^N(x)\right], \tag{D.105}$$

where

$$R_t^N(x) := \frac{1}{N}\sum_{i=1}^N \rho_{t-1,i}\mathbf{1}_{\{X_t^{i,N}=x\}}. \tag{D.106}$$

The transport term is controlled pathwise by the exact predecessor marginal $\sum_x \Gamma_t^N(y,x) = U_{t-1}^N(y)$:

$$\begin{aligned}\sum_x \left\|\sum_y \Gamma_t^N(y,x)\big(\widehat{z}_{t-1}^{N,D,0}(y) - z_{t-1}^{\mathrm{priv},N,0}(y)\big)\right\|_2 \\ \le \sum_y U_{t-1}^N(y)\|\widehat{z}_{t-1}^{N,D,0}(y) - z_{t-1}^{\mathrm{priv},N,0}(y)\|_2 = \|D_{t-1}^N\|_{1,2}.\end{aligned} \tag{D.107}$$

Under algorithmic setting D-T2-A2, define the local finite-window envelopes

$$B_A^{(2)} := 1 + \alpha B_\phi^2, \qquad B_{e,T}^{(2)} := B_\phi \sum_{j=0}^{T} (\beta B_A^{(2)})^j. \tag{D.108}$$

The private trace recursion and the definition of the within-state average then give

$$\|\rho_{t-1,i}\|_2 \le 2B_{e,T}^{(2)}, \qquad \|A_t^N\|_{\mathrm{op}} \le B_A^{(2)}. \tag{D.109}$$

Conditioned on $\mathcal{F}_{t-1}^N$, define

$$p_{t-1,i}^N(x) := P_{t-1}\Big(x \mid X_{t-1}^{i,N}, U_{t-1}^N\Big). \tag{D.110}$$

For agents sharing the same predecessor state y, this probability is the common value $P_{t-1}(x \mid y, U_{t-1}^N)$. Hence the within-state cancellation in Eq. (D.104) gives

$$\sum_{i=1}^{N} \rho_{t-1,i} p_{t-1,i}^N(x) = \sum_{y \in \mathcal{X}} P_{t-1}(x \mid y, U_{t-1}^N) \sum_{i: X_{t-1}^{i,N} = y} \rho_{t-1,i} \tag{D.111}$$

$$= 0. \tag{D.112}$$

Therefore Assumption D-T2-A1 implies

$$\mathbb{E}[R_t^N(x) \mid \mathcal{F}_{t-1}^N] = 0, \tag{D.113}$$

and, using Eq. (D.112),

$$R_t^N(x) = \frac{1}{N} \sum_{i=1}^{N} \rho_{t-1,i} \left(\mathbf{1}_{\{X_t^{i,N} = x\}} - p_{t-1,i}^N(x) \right). \tag{D.114}$$

Using this zero total conditional mean together with conditional independence of the fresh destination draws yields

$$\mathbb{E}\big[\|R_t^N(x)\|_2^2 \mid \mathcal{F}_{t-1}^N\big] \le \frac{1}{N^2} \sum_{i=1}^{N} \|\rho_{t-1,i}\|_2^2 \le \frac{4(B_{e,T}^{(2)})^2}{N}, \tag{D.115}$$

so Jensen's inequality gives

$$\mathbb{E}\|R_t^N\|_{1,2} \le \frac{2SB_{e,T}^{(2)}}{\sqrt{N}}. \tag{D.116}$$

Combining Eqs. (D.105)–(D.116),

$$\mathbb{E}\|D_t^N\|_{1,2} \le \beta B_A^{(2)} \mathbb{E}\|D_{t-1}^N\|_{1,2} + \frac{2\beta B_A^{(2)} S B_{e,T}^{(2)}}{\sqrt{N}}. \tag{D.117}$$

Iterating from $D_0^N = 0$ gives

$$\max_{0 \le t \le T} \mathbb{E}\|D_t^N\|_{1,2} \le \frac{C_{\mathrm{credit},T}}{\sqrt{N}}, \qquad C_{\mathrm{credit},T} := 2SB_{e,T}^{(2)} \sum_{j=1}^{T} (\beta B_A^{(2)})^j < \infty. \tag{D.118}$$

This is Eq. (D.99).

Theorem 2 is a direct finite-population bridge. Within-state centering removes the conditional mean of the private residual, fresh idiosyncratic transition noise gives the $N^{-1/2}$ fluctuation, and unnormalized mass transport propagates the discrepancy without any occupancy lower bound or mean-field regularity assumption. □

Remark 8. *Theorem 2 is a fixed-window expected-norm result for unnormalized state-aggregated credit masses. It does not assert pathwise $N^{-1/2}$ closeness of individual private traces and state-wise readouts, and it uses no global positive lower bound on state occupancy. Pointwise normalized-readout bounds require an additional well-occupied-state condition and are not assumptions of Theorem 2.*

D.4 Finite-Population Learner Consistency Theorem 3

Explanation. The credit results in Sec. D.3 control the state-wise historical information used by the finite-population learners. We now transfer those credit-level errors to the critic parameters. The first comparison places DoubleSWEET against a deterministic mean-field learner: after conditioning on the current finite-population history, the remaining fresh-transition fluctuation averages across agents, while the conditional-mean discrepancy is controlled by the physical and credit errors already established. The second comparison places DoubleSWEET directly against Private Dutch on the same realized population episode: their correction terms cancel, their state-aggregated credit fields are close by Theorem 2, and the residual within-state trace fluctuation is centered with respect to the fresh transition noise. Finite-time Lipschitz recursions then convert both one-step bounds into parameter-trajectory bounds, from which Theorem 3 follows by a triangle inequality.

Formal construction. Let

$$\mathcal{T}_T^- := \{0, 1, \dots, T-1\}, \qquad \chi_{t+1} := \mathbf{1}_{\{t+1<T\}}. \tag{D.119}$$

For a deterministic comparison pair $(w, w_-) \in \mathbb{R}^d \times \mathbb{R}^d$, define

$$\delta'^{,N}_{t,i}(w, w_-) := r_t(X_t^{i,N}, U_t^N) + \gamma\chi_{t+1} w^\top \phi_{t+1}(X_{t+1}^{i,N}) - w_-^\top \phi_t(X_t^{i,N}), \tag{D.120}$$

$$c^N_{t,i}(w, w_-) := (w - w_-)^\top \phi_t(X_t^{i,N}), \tag{D.121}$$

$$q_t(x; w, w_-) := \big[(w - w_-)^\top \phi_t(x)\big]\phi_t(x). \tag{D.122}$$

The finite-population and mean-field conditional innovation coefficients are

$$h_t^N(x; w, w_-) := r_t(x, U_t^N) + \gamma\chi_{t+1} \sum_{x' \in \mathcal{X}} P_t(x' \mid x, U_t^N) w^\top \phi_{t+1}(x') - w_-^\top \phi_t(x), \tag{D.123}$$

$$h_t(x; w, w_-) := r_t(x, u_t) + \gamma\chi_{t+1} \sum_{x' \in \mathcal{X}} P_t(x' \mid x, u_t) w^\top \phi_{t+1}(x') - w_-^\top \phi_t(x). \tag{D.124}$$

Using the zero-extended readouts in Eq. (D.50), define the three one-step update maps

$$\Delta_t^{N,C}(w, w_-) := \frac{\alpha}{N} \sum_{i=1}^N \left[\delta'^{,N}_{t,i}(w, w_-) \widehat{z}_t^{N,D,0}(X_t^{i,N}) - c^N_{t,i}(w, w_-)\phi_t(X_t^{i,N})\right], \tag{D.125}$$

$$\Delta_t^{N,\mathrm{priv}}(w, w_-) := \frac{\alpha}{N} \sum_{i=1}^N \left[\delta'^{,N}_{t,i}(w, w_-) e_t^{i,N,D} - c^N_{t,i}(w, w_-)\phi_t(X_t^{i,N})\right], \tag{D.126}$$

$$\Delta_t^{MF}(w, w_-) := \alpha \sum_{x \in \mathcal{X}} u_t(x) \left[h_t(x; w, w_-) z_t^{D,0}(x) - q_t(x; w, w_-)\right]. \tag{D.127}$$

The corresponding learner trajectories satisfy

$$w_{t+1}^{N,C} = w_t^{N,C} + \Delta_t^{N,C}(w_t^{N,C}, w_{t-1}^{N,C}), \tag{D.128}$$

$$w_{t+1}^{N,D} = w_t^{N,D} + \Delta_t^{N,\mathrm{priv}}(w_t^{N,D}, w_{t-1}^{N,D}), \tag{D.129}$$

$$w_{t+1}^{MF} = w_t^{MF} + \Delta_t^{MF}(w_t^{MF}, w_{t-1}^{MF}), \qquad t \in \mathcal{T}_T^-. \tag{D.130}$$

For the direct compressed/private comparison, define the zero-extended within-state Private Dutch average

$$z_t^{\mathrm{priv},N,0}(x) := \begin{cases} m_t^{\mathrm{priv},N,D}(x)/U_t^N(x), & U_t^N(x) > 0, \\ 0, & U_t^N(x) = 0, \end{cases} \tag{D.131}$$

and the two population-weighted readout discrepancies

$$G_t^{CM,N} := \sum_{x \in \mathcal{X}} U_t^N(x) \|\widehat{z}_t^{N,D,0}(x) - z_t^{D,0}(x)\|_2, \tag{D.132}$$

$$G_t^{CP,N} := \sum_{x \in \mathcal{X}} U_t^N(x) \|\widehat{z}_t^{N,D,0}(x) - z_t^{\mathrm{priv},N,0}(x)\|_2 = \|\widehat{m}_t^{N,D} - m_t^{\mathrm{priv},N,D}\|_{1,2}. \tag{D.133}$$

Fixed-policy finite-window setting. The state space and horizon are finite and one fixed policy generates the whole comparison episode:

$$|\mathcal{X}| = S < \infty, \qquad 1 \le T < \infty.$$

Assumption D-T3-A1 (Consistent iid initialization). The finite population starts from independent draws of the same deterministic initial law as the representative mean-field system.

Mathematical form of D-T3-A1.

$$X_0^{1,N}, \dots, X_0^{N,N} \overset{\text{iid}}{\sim} u_0. \tag{D.134}$$

Assumption D-T3-A2 (Idiosyncratic transition noise without common noise). Conditioned on the current finite-population history, the fresh next-state draws are independent across agents and follow the prescribed state- and population-dependent kernels, with no shared transition noise.

Mathematical form of D-T3-A2.

$$\mathcal{L}\Big((X_{t+1}^{i,N})_{i=1}^N \mid \mathcal{F}_t^N\Big) = \bigotimes_{i=1}^N P_t(\cdot \mid X_t^{i,N}, U_t^N), \qquad 0 \le t < T. \tag{D.135}$$

Assumption D-T3-A3 (Population-regular transition kernel). At a fixed physical state, the closed-loop next-state law varies Lipschitz-continuously with the population distribution over the fixed horizon.

Mathematical form of D-T3-A3.

$$\|P_t(\cdot \mid x, u) - P_t(\cdot \mid x, v)\|_{\mathrm{TV}} \le L_P \|u - v\|_1, \qquad {\scriptstyle 0 \le t < T,\ x \in \mathcal{X},\ u, v \in \mathcal{P}(\mathcal{X})}. \tag{D.136}$$

Algorithmic setting D-T3-A4 (Shared features and fixed Dutch parameters). All three learners use the same time-state feature map, the same fixed Dutch parameters, and the same terminal-bootstrap convention. The finite state-time domain gives the derived bound

$$B_\phi := \max_{0 \le t \le T} \max_{x \in \mathcal{X}} \|\phi_t(x)\|_2 < \infty.$$

Mathematical form of D-T3-A4.

$$\begin{gathered} \alpha > 0, \qquad 0 \le \gamma \le 1, \qquad 0 \le \lambda \le 1, \qquad \beta = \gamma\lambda, \\ \chi_{t+1} = \mathbf{1}_{\{t+1<T\}}. \end{gathered} \tag{D.137}$$

Reward setting. As in the main model, rewards are deterministic functions of the current state and population distribution,

$$R_{t+1}^{i,N} = r_t(X_t^{i,N}, U_t^N), \qquad i \in [N], \quad 0 \le t < T.$$

Assumption D-T3-A5 (Population-regular rewards). The reward function is uniformly Lipschitz in the population distribution.

Mathematical form of D-T3-A5.

$$|r_t(x, u) - r_t(x, v)| \le L_r \|u - v\|_1, \qquad {\scriptstyle 0 \le t < T,\ x \in \mathcal{X},\ u, v \in \mathcal{P}(\mathcal{X})}. \tag{D.138}$$

Fix any $\bar{u} \in \mathcal{P}(\mathcal{X})$. Since t and x range over finite sets and $\|u - \bar{u}\|_1 \le 2$, D-T3-A5 implies the derived uniform bound

$$B_r := \max_{0 \le t < T} \max_{x \in \mathcal{X}} |r_t(x, \bar{u})| + 2L_r < \infty, \qquad |r_t(x, u)| \le B_r. \tag{D.139}$$

Comparison setting D-T3-A6 (Common deterministic learner initialization). The DoubleSWEET, Private Dutch, and deterministic mean-field learners start from the same deterministic critic parameter and use the same previous-parameter convention.

Mathematical form of D-T3-A6.

$$w_{-1}^{N,C} = w_0^{N,C} = w_{-1}^{N,D} = w_0^{N,D} = w_{-1}^{MF} = w_0^{MF} =: w_0 \in \mathbb{R}^d. \tag{D.140}$$

Theorem 3. *In the fixed-policy finite-window setting and reward setting above, under Assumptions D-T3-A1, D-T3-A2, D-T3-A3, and D-T3-A5, together with algorithmic setting D-T3-A4 and comparison setting D-T3-A6, the Private Dutch learner, the DoubleSWEET learner, and the deterministic mean-field learner are mutually consistent at the inverse-square-root population rate on every fixed horizon, with a finite constant independent of population size.*

$$C_{\mathrm{learner},T} < \infty,$$
$$\max_{0\le t\le T} \mathbb{E}\Big[\max\Big\{\|w_t^{N,D} - w_t^{MF}\|_2, \|w_t^{N,C} - w_t^{MF}\|_2, \|w_t^{N,D} - w_t^{N,C}\|_2\Big\}\Big] \le \frac{C_{\mathrm{learner},T}}{\sqrt{N}}. \tag{D.141}$$

Proof. First control the DoubleSWEET/mean-field and DoubleSWEET/Private-Dutch one-step discrepancies at a common parameter pair. Finite-window radius and Lipschitz estimates propagate those two bounds through the adaptive learner recursions. The remaining Private-Dutch/mean-field discrepancy is then bounded by their sum, so the same two primary errors establish all three pairwise gaps in Theorem 3.

The fixed-policy finite-window setting together with D-T3-A1, D-T3-A2, and D-T3-A3, together with D-T3-A4, implies the physical and mean-field credit conditions used in Lemmas 3 and 4; in particular, D-T3-A2 and D-T3-A4 imply the weaker direct finite-population conditions of Theorem 2. D-T3-A5 supplies the additional reward regularity required for learner updates, and D-T3-A6 aligns the three initial critic pairs.

Step 1: common Dutch envelopes and readout bridges. Set

$$B_A := 1 + \alpha B_\phi^2, \qquad B_{\mathrm{tr},T} := B_\phi \sum_{k=0}^{T} (\beta B_A)^k. \tag{D.142}$$

By Eqs. (D.53), (D.54), and the compressed normalized recursion induced by Eq. (D.52),

$$\max_{0\le t\le T} \left\{ \|z_t^{D,0}\|_\infty, \|\widehat{z}_t^{N,D,0}\|_\infty, \max_{1\le i\le N} \|e_t^{i,N,D}\|_2 \right\} \le B_{\mathrm{tr},T} \tag{D.143}$$

pathwise. Moreover,

$$G_t^{CM,N} \le \|\widehat{m}_t^{N,D} - m_t^D\|_{1,2} + B_{\mathrm{tr},T}\|U_t^N - u_t\|_1, \tag{D.144}$$
$$G_t^{CP,N} = \|\widehat{m}_t^{N,D} - m_t^{\mathrm{priv},N,D}\|_{1,2}. \tag{D.145}$$

Thus, by Lemmas 3 and 4, together with Theorem 2,

$$\max_{0\le t\le T} \mathbb{E}G_t^{CM,N} \le \frac{C_{z,T}}{\sqrt{N}}, \qquad \max_{0\le t\le T} \mathbb{E}G_t^{CP,N} \le \frac{C_{\mathrm{credit},T}}{\sqrt{N}}, \tag{D.146}$$

where

$$C_{z,T} := C_{m,T} + B_{\mathrm{tr},T} C_{U,T}, \qquad C_{\mathrm{credit},T} < \infty \tag{D.147}$$

denotes the finite direct compressed/private credit constant supplied by Theorem 2.

Step 2: same-parameter DoubleSWEET/mean-field update. Fix $B_w < \infty$ and $(w, w_-) \in \mathcal{W}_{B_w}^2$, where

$$\mathcal{W}_{B_w} := \{v \in \mathbb{R}^d : \|v\|_2 \le B_w\}. \tag{D.148}$$

Define

$$\begin{aligned} B_\delta(B_w) &:= B_r + (1+\gamma) B_w B_\phi, \\ B_h(B_w) &:= B_\delta(B_w), \\ B_q(B_w) &:= 2 B_w B_\phi^2, \\ L_h(B_w) &:= L_r + 2\gamma B_w B_\phi L_P. \end{aligned} \tag{D.149}$$

Conditioning on $\mathcal{F}_t^N$ gives

$$\begin{aligned}\bar{\Delta}_t^{N,C}(w,w_-) &:= \mathbb{E}[\Delta_t^{N,C}(w,w_-) \mid \mathcal{F}_t^N] \\ &= \alpha \sum_x U_t^N(x)[h_t^N(x;w,w_-)\widehat{z}_t^{N,D,0}(x) - q_t(x;w,w_-)].\end{aligned} \tag{D.150}$$

With

$$\eta_{t,i}^N := \delta_{t,i}'^{,N}(w,w_-) - h_t^N(X_t^{i,N};w,w_-), \tag{D.151}$$

Assumption D-T3-A2 gives conditional centering and independence. Hence

$$\mathbb{E}\|\Delta_t^{N,C} - \bar{\Delta}_t^{N,C}\|_2 \le \frac{\alpha B_\delta(B_w) B_{\mathrm{tr},T}}{\sqrt{N}}. \tag{D.152}$$

Also,

$$\|h_t^N - h_t\|_\infty \le L_h(B_w)\|U_t^N - u_t\|_1. \tag{D.153}$$

Using Eqs. (D.132), (D.146), and (D.153),

$$\begin{aligned}&\mathbb{E}\|\bar{\Delta}_t^{N,C}(w,w_-) - \Delta_t^{MF}(w,w_-)\|_2 \\ &\quad\le \frac{\alpha}{\sqrt{N}}\left[B_h(B_w)C_{z,T} + B_{\mathrm{tr},T}L_h(B_w)C_{U,T} + B_h(B_w)B_{\mathrm{tr},T}C_{U,T} + B_q(B_w)C_{U,T}\right].\end{aligned} \tag{D.154}$$

Therefore

$$\sup_{(w,w_-)\in\mathcal{W}_{B_w}^2} \max_{t\in\mathcal{T}_T^-} \mathbb{E}\|\Delta_t^{N,C}(w,w_-) - \Delta_t^{MF}(w,w_-)\|_2 \le \frac{C_{\Delta,T}(B_w)}{\sqrt{N}}, \tag{D.155}$$

where

$$\begin{aligned}C_{\Delta,T}(B_w) := {}& \alpha B_\delta(B_w)B_{\mathrm{tr},T} + \alpha B_h(B_w)C_{z,T} \\ &+ \alpha B_{\mathrm{tr},T}L_h(B_w)C_{U,T} + \alpha B_h(B_w)B_{\mathrm{tr},T}C_{U,T} + \alpha B_q(B_w)C_{U,T}.\end{aligned} \tag{D.156}$$

Step 3: update Lipschitzness and the mean-field radius. For every realized finite-population batch,

$$\|\Delta_t^{N,C}(w,w_-) - \Delta_t^{N,C}(v,v_-)\|_2 \le L_C(\|w-v\|_2 + \|w_- - v_-\|_2), \tag{D.157}$$

with

$$L_C := \alpha(B_\phi B_{\mathrm{tr},T} + B_\phi^2). \tag{D.158}$$

For the deterministic mean-field learner, set

$$a_0 := \alpha B_r B_{\mathrm{tr},T}, \quad a_1 := \alpha(\gamma B_\phi B_{\mathrm{tr},T} + B_\phi^2), \quad a_2 := \alpha(B_\phi B_{\mathrm{tr},T} + B_\phi^2), \quad \rho_w := 1 + a_1 + a_2. \tag{D.159}$$

Then

$$\max_{-1\le t\le T}\|w_t^{MF}\|_2 \le B_{w,T}^{MF} := (\rho_w)^T\|w_0\|_2 + a_0\sum_{k=0}^{T-1}(\rho_w)^k. \tag{D.160}$$

Step 4: DoubleSWEET/mean-field trajectory. Let

$$d_t^{CM,N} := \mathbb{E}\|w_t^{N,C} - w_t^{MF}\|_2. \tag{D.161}$$

By Eqs. (D.155)–(D.160),

$$d_{t+1}^{CM,N} \le (1+L_C)d_t^{CM,N} + L_C d_{t-1}^{CM,N} + \frac{C_{\Delta,T}(B_{w,T}^{MF})}{\sqrt{N}}, \qquad d_{-1}^{CM,N} = d_0^{CM,N} = 0. \tag{D.162}$$

Hence

$$\max_{0\le t\le T} d_t^{CM,N} \le \frac{C_{CM,T}}{\sqrt{N}}, \qquad C_{CM,T} := C_{\Delta,T}(B_{w,T}^{MF})\sum_{k=0}^{T-1}(1+2L_C)^k. \tag{D.163}$$

Step 5: compressed/private one-step comparison. For every occupied state, Eq. (D.131) is an average of private traces; hence

$$\|z_t^{\mathrm{priv},N,0}\|_\infty \le B_{\mathrm{tr},T}. \tag{D.164}$$

For a fixed pair $(w, w_-) \in \mathcal{W}^2_{B_w}$, the correction terms cancel:

$$\Delta_t^{N,C} - \Delta_t^{N,\mathrm{priv}} = \frac{\alpha}{N} \sum_{i=1}^{N} \delta'^{,N}_{t,i}(w, w_-)[\widehat{z}_t^{N,D,0}(X_t^{i,N}) - e_t^{i,N,D}]. \tag{D.165}$$

Insert $z_t^{\mathrm{priv},N,0}(X_t^{i,N})$. The bridge part obeys

$$\|\mathcal{E}^N_{t,\mathrm{bridge}}\|_2 \le \alpha B_\delta(B_w) G_t^{CP,N}. \tag{D.166}$$

For

$$d^N_{t,i} := z_t^{\mathrm{priv},N,0}(X_t^{i,N}) - e_t^{i,N,D}, \tag{D.167}$$

the state-wise identity

$$\sum_{i:X_t^{i,N}=x} d^N_{t,i} = 0 \tag{D.168}$$

holds on every occupied state. Therefore the h_t^N component cancels and

$$\mathcal{E}^N_{t,\mathrm{noise}} = \frac{\alpha}{N} \sum_{i=1}^{N} \eta^N_{t,i} d^N_{t,i}. \tag{D.169}$$

By Assumption D-T3-A2, conditional centering and independence give

$$\mathbb{E}[\|\mathcal{E}^N_{t,\mathrm{noise}}\|_2 \mid \mathcal{F}_t^N] \le \frac{2\alpha B_\delta(B_w) B_{\mathrm{tr},T}}{\sqrt{N}}. \tag{D.170}$$

Consequently, the bound is uniform for every $\mathcal{F}_t^N$-measurable pair in the radius-B_w ball, and

$$\max_{t \in \mathcal{T}_T^-} \mathbb{E}\|\Delta_t^{N,C}(W_t, W_{t,-}) - \Delta_t^{N,\mathrm{priv}}(W_t, W_{t,-})\|_2 \le \frac{C_{CP,T}(B_w)}{\sqrt{N}}, \tag{D.171}$$

where

$$C_{CP,T}(B_w) := \alpha B_\delta(B_w)(C_{\mathrm{credit},T} + 2B_{\mathrm{tr},T}). \tag{D.172}$$

Step 6: DoubleSWEET/Private Dutch trajectory. For either finite update map $\Delta_t^{N,\star}$, $\star \in \{C, \mathrm{priv}\}$, Eqs. (D.143) and the derived reward bound (D.139) give, pathwise,

$$\|\Delta_t^{N,\star}(w, w_-)\|_2 \le a_0 + a_1\|w\|_2 + a_2\|w_-\|_2, \tag{D.173}$$

with the same constants a_0, a_1, a_2 as in Eq. (D.159). Indeed, the TD term is bounded by $B_r + \gamma B_\phi\|w\|_2 + B_\phi\|w_-\|_2$, the trace/readout by $B_{\mathrm{tr},T}$, and the correction term by $B_\phi^2(\|w\|_2 + \|w_-\|_2)$. Since $w_{-1}^{N,\star} = w_0^{N,\star} = w_0$, a finite-horizon induction using $\rho_w = 1 + a_1 + a_2$ yields, pathwise,

$$\max_{-1 \le t \le T} \max\{\|w_t^{N,C}\|_2, \|w_t^{N,D}\|_2\} \le B^{CP}_{w,T}, \tag{D.174}$$

where

$$B^{CP}_{w,T} := (\rho_w)^T\|w_0\|_2 + a_0 \sum_{k=0}^{T-1} (\rho_w)^k, \tag{D.175}$$

with the constants from Eq. (D.159). Let

$$d_t^{CD,N} := \mathbb{E}\|w_t^{N,C} - w_t^{N,D}\|_2. \tag{D.176}$$

Using Eq. (D.157) and Eq. (D.171) at the adapted Private Dutch parameter pair,

$$d_{t+1}^{CD,N} \le (1 + L_C) d_t^{CD,N} + L_C d_{t-1}^{CD,N} + \frac{C_{CP,T}(B^{CP}_{w,T})}{\sqrt{N}}, \qquad d_{-1}^{CD,N} = d_0^{CD,N} = 0. \tag{D.177}$$

Thus

$$\max_{0 \le t \le T} d_t^{CD,N} \le \frac{C_{CD,T}}{\sqrt{N}}, \qquad C_{CD,T} := C_{CP,T}(B^{CP}_{w,T}) \sum_{k=0}^{T-1} (1 + 2L_C)^k. \tag{D.178}$$

Combining Eqs. (D.163) and (D.178),

$$\max_{0\le t\le T}\mathbb{E}\|w_t^{N,C}-w_t^{MF}\|_2\le\frac{C_{CM,T}}{\sqrt{N}},\qquad \max_{0\le t\le T}\mathbb{E}\|w_t^{N,C}-w_t^{N,D}\|_2\le\frac{C_{CD,T}}{\sqrt{N}}. \tag{D.179}$$

Step 7: joint main-paper bound. For every $0\le t\le T$, set

$$b_t^N:=\|w_t^{N,C}-w_t^{MF}\|_2,\qquad c_t^N:=\|w_t^{N,D}-w_t^{N,C}\|_2. \tag{D.180}$$

Then

$$\|w_t^{N,D}-w_t^{MF}\|_2\le b_t^N+c_t^N, \tag{D.181}$$

and therefore

$$\max\Big\{\|w_t^{N,D}-w_t^{MF}\|_2,b_t^N,c_t^N\Big\}\le b_t^N+c_t^N. \tag{D.182}$$

Using Eq. (D.179),

$$\max_{0\le t\le T}\mathbb{E}\Big[\max\Big\{\|w_t^{N,D}-w_t^{MF}\|_2,b_t^N,c_t^N\Big\}\Big]\le\frac{C_{CM,T}+C_{CD,T}}{\sqrt{N}}. \tag{D.183}$$

For Theorem 3, take

$$C_{\mathrm{learner},T}:=C_{CM,T}+C_{CD,T}. \tag{D.184}$$

The physical and credit bounds established earlier yield two learner-level comparisons: DoubleSWEET versus mean field and DoubleSWEET versus Private Dutch. Their finite-horizon recursions give $O(N^{-1/2})$ parameter errors, and one final triangle inequality supplies the third pairwise error. No alternative Private-Dutch-to-mean-field proof route is needed. □

Learned-critic consequence. Define

$$V_t^{N,C}(x):=(w_t^{N,C})^\top\phi_t(x),\quad V_t^{N,D}(x):=(w_t^{N,D})^\top\phi_t(x),\quad V_t^{MF}(x):=(w_t^{MF})^\top\phi_t(x). \tag{D.185}$$

Then setting D-T3-A4 and Theorem 3 give

$$\max_{0\le t\le T}\mathbb{E}\Big[\max\Big\{\|V_t^{N,D}-V_t^{MF}\|_\infty,\|V_t^{N,C}-V_t^{MF}\|_\infty,\|V_t^{N,D}-V_t^{N,C}\|_\infty\Big\}\Big]\le\frac{B_\phi C_{\mathrm{learner},T}}{\sqrt{N}}. \tag{D.186}$$

Remark 9. *Theorem 3 is a fixed-window expected-norm consistency result for learned critic parameters. It does not assert pathwise $N^{-1/2}$ closeness, a bound uniform as $T\to\infty$, or convergence of the common mean-field critic to the exact Bellman policy value. The direct compressed/private branch uses the same realized finite-population data stream and the conditional independence of the fresh idiosyncratic transitions; no alternative mean-field-mediated Private Dutch proof route is needed.*

D.5 Live-Flow and Frozen-Flow Equivalence Theorem 4

Explanation. The standard finite-horizon mean-field workflow first generates the deterministic population flow and then evaluates the fixed policy against that frozen flow. The live-flow formulation instead evaluates the same fixed policy while the same deterministic population flow is generated. The comparison below is deliberately between two execution orders of the same exact-expectation Dutch evaluator, rather than between different policy-evaluation algorithms. Once the representative process, Dutch operator, trace construction, terminal convention, initialization, and expectation operator are matched, the two procedures expose the evaluator to exactly the same time-inhomogeneous mean-field environment, have the same state–trace law, and therefore have the same expected update map at every common parameter pair. Their critic trajectories consequently coincide by induction. This establishes the execution-order equivalence stated as Theorem 4 in the main paper.

Formal construction. Reuse the deterministic mean-field flow and kernel from Eq. (D.4). For the live-flow protocol, let the representative process be the mean-field process from Eq. (D.5); write its Dutch trace as

$$e_{-1}^{\mathrm{live}} = 0, \qquad e_t^{\mathrm{live}} := \phi_t(\bar{X}_t) + \beta A_t e_{t-1}^{\mathrm{live}}, \qquad 0 \le t \le T. \tag{D.187}$$

For the frozen-flow protocol, first regard the same deterministic sequence $u_{0:T}$ as fixed and then define

$$\widetilde{X}_0 \sim u_0, \qquad \mathcal{L}(\widetilde{X}_{t+1} \mid \widetilde{X}_0, \dots, \widetilde{X}_t) = K_t(\cdot \mid \widetilde{X}_t), \qquad 0 \le t < T, \tag{D.188}$$

and

$$e_{-1}^{\mathrm{frozen}} = 0, \qquad e_t^{\mathrm{frozen}} := \phi_t(\widetilde{X}_t) + \beta A_t e_{t-1}^{\mathrm{frozen}}, \qquad 0 \le t \le T. \tag{D.189}$$

Define the two credit masses

$$m_t^{\mathrm{live}}(x) := \mathbb{E}\big[e_t^{\mathrm{live}} \mathbf{1}_{\{\bar{X}_t = x\}}\big], \qquad m_t^{\mathrm{frozen}}(x) := \mathbb{E}\Big[e_t^{\mathrm{frozen}} \mathbf{1}_{\{\widetilde{X}_t = x\}}\Big]. \tag{D.190}$$

For a deterministic comparison pair $(w, w_-) \in \mathbb{R}^d \times \mathbb{R}^d$, define

$$q_t(x; w, w_-) := \big[(w - w_-)^\top \phi_t(x)\big]\phi_t(x), \tag{D.191}$$

$$\delta_t'^{,\mathrm{live}}(w, w_-) := r_t(\bar{X}_t, u_t) + \gamma \chi_{t+1} w^\top \phi_{t+1}(\bar{X}_{t+1}) - w_-^\top \phi_t(\bar{X}_t), \tag{D.192}$$

$$\delta_t'^{,\mathrm{frozen}}(w, w_-) := r_t(\widetilde{X}_t, u_t) + \gamma \chi_{t+1} w^\top \phi_{t+1}(\widetilde{X}_{t+1}) - w_-^\top \phi_t(\widetilde{X}_t), \qquad 0 \le t < T. \tag{D.193}$$

The exact expected update maps are

$$\Delta_t^{\mathrm{live}}(w, w_-) := \alpha\, \mathbb{E}\Big[\delta_t'^{,\mathrm{live}}(w, w_-) e_t^{\mathrm{live}} - q_t(\bar{X}_t; w, w_-)\Big], \tag{D.194}$$

$$\Delta_t^{\mathrm{frozen}}(w, w_-) := \alpha\, \mathbb{E}\Big[\delta_t'^{,\mathrm{frozen}}(w, w_-) e_t^{\mathrm{frozen}} - q_t(\widetilde{X}_t; w, w_-)\Big]. \tag{D.195}$$

Moreover, the deterministic mean-field update map used in Sec. D.4 is exactly the live-flow update map. By Eqs. (D.9) and (D.187), $e_t^{\mathrm{live}} = \bar{e}_t^D$ under the live representative construction. Conditioning on $\bar{X}_t = x$ and using the Markov screening of this past-dependent trace from the fresh draw $\bar{X}_{t+1}$ given $\bar{X}_t$ therefore gives

$$\begin{aligned} \Delta_t^{\mathrm{live}}(w, w_-) &= \alpha \sum_{x \in \mathcal{X}} u_t(x) \Big[h_t(x; w, w_-) z_t^{D,0}(x) - q_t(x; w, w_-)\Big] \\ &= \Delta_t^{MF}(w, w_-), \end{aligned} \qquad (w, w_-) \in \mathbb{R}^d \times \mathbb{R}^d, \quad 0 \le t < T. \tag{D.196}$$

Thus, under the common initialization in setting D-T4-A4, the mean-field learner w_t^{MF} in Theorem 3 is exactly the live-flow learner. Their learner trajectories are

$$w_{t+1}^{\mathrm{live}} := w_t^{\mathrm{live}} + \Delta_t^{\mathrm{live}}(w_t^{\mathrm{live}}, w_{t-1}^{\mathrm{live}}), \tag{D.197}$$

$$w_{t+1}^{\mathrm{frozen}} := w_t^{\mathrm{frozen}} + \Delta_t^{\mathrm{frozen}}(w_t^{\mathrm{frozen}}, w_{t-1}^{\mathrm{frozen}}), \qquad 0 \le t < T, \tag{D.198}$$

and the associated learned values are

$$V_t^{\mathrm{live}}(x) := (w_t^{\mathrm{live}})^\top \phi_t(x), \qquad V_t^{\mathrm{frozen}}(x) := (w_t^{\mathrm{frozen}})^\top \phi_t(x). \tag{D.199}$$

Setting D-T4-A1 (Shared deterministic mean-field environment). The two procedures use the same deterministic population flow generated by the fixed policy. Their transition kernel and reward depend only on the physical time–state–population context, while the feature representation depends only on time and state; none depends on the critic parameter, so the order of policy evaluation does not alter the mean-field environment.

Mathematical form of D-T4-A1.

$$\begin{gathered} u_0 \in \mathcal{P}(\mathcal{X}), \qquad K_t(x' \mid x) = P_t(x' \mid x, u_t), \\ u_{t+1}(x') = \sum_{x \in \mathcal{X}} u_t(x) K_t(x' \mid x), \\ P_t : \mathcal{X} \times \mathcal{P}(\mathcal{X}) \to \mathcal{P}(\mathcal{X}), \qquad r_t : \mathcal{X} \times \mathcal{P}(\mathcal{X}) \to \mathbb{R}, \qquad 0 \le t < T, \\ \phi_t : \mathcal{X} \to \mathbb{R}^d, \qquad 0 \le t \le T. \end{gathered} \tag{D.200}$$

Setting D-T4-A2 (Matched representative-agent and Dutch constructions). The live and frozen representatives start from the same initial law, evolve with the same time-inhomogeneous closed-loop kernels, and use the same deterministic population Dutch operator, feature map, trace parameters, trace initialization, and terminal-bootstrap convention.

Mathematical form of D-T4-A2.

$$\begin{gathered}\bar{X}_0 \sim u_0, \qquad \widetilde{X}_0 \sim u_0,\\ \mathcal{L}(\bar{X}_{t+1} \mid \bar{X}_{0:t}) = K_t(\cdot \mid \bar{X}_t), \qquad \mathcal{L}(\widetilde{X}_{t+1} \mid \widetilde{X}_{0:t}) = K_t(\cdot \mid \widetilde{X}_t),\\ \alpha > 0, \qquad A_t = I_d - \alpha \sum_{x \in \mathcal{X}} u_t(x)\phi_t(x)\phi_t(x)^\top,\\ e_{-1}^{\mathrm{live}} = e_{-1}^{\mathrm{frozen}} = 0, \qquad \beta = \gamma\lambda, \qquad 0 \le \gamma \le 1, \qquad 0 \le \lambda \le 1, \qquad \chi_{t+1} = \mathbf{1}_{\{t+1<T\}}.\end{gathered} \tag{D.201}$$

Evaluation setting D-T4-A3 (Exact representative-agent expectations). Both procedures use exact expectations under their representative-agent path laws when forming the population critic increment. No single sampled representative trajectory is substituted for either expectation in the theorem.

Mathematical form of D-T4-A3.

$$\Delta_t^{\mathrm{live}}(w, w_-) = \alpha\, \mathbb{E}\Big[\delta_t'^{,\mathrm{live}}(w, w_-) e_t^{\mathrm{live}} - q_t(\bar{X}_t; w, w_-)\Big], \tag{D.202}$$

$$\Delta_t^{\mathrm{frozen}}(w, w_-) = \alpha\, \mathbb{E}\Big[\delta_t'^{,\mathrm{frozen}}(w, w_-) e_t^{\mathrm{frozen}} - q_t(\widetilde{X}_t; w, w_-)\Big]. \tag{D.203}$$

Comparison setting D-T4-A4 (Common learner initialization). The live and frozen critic recursions start from the same deterministic parameter and use the same previous-parameter convention.

Mathematical form of D-T4-A4.

$$w_{-1}^{\mathrm{live}} = w_0^{\mathrm{live}} = w_{-1}^{\mathrm{frozen}} = w_0^{\mathrm{frozen}} = w_0 \in \mathbb{R}^d. \tag{D.204}$$

Theorem 4. *Under settings D-T4-A1, D-T4-A2, D-T4-A3, and D-T4-A4, running the same matched exact-expectation Dutch evaluator while the deterministic mean-field population flow is generated or after freezing that same flow produces exactly the same critic trajectory and value-function readout.*

$$\begin{aligned} w_t^{\mathrm{live}} &= w_t^{\mathrm{frozen}}, \qquad 0 \le t \le T,\\ V_t^{\mathrm{live}}(x) &= V_t^{\mathrm{frozen}}(x), \qquad 0 \le t \le T, \quad x \in \mathcal{X}. \end{aligned} \tag{D.205}$$

Proof. The two representative processes have the same finite-horizon path law because they share the initial law and every time-inhomogeneous transition kernel. Their Dutch traces are the same deterministic functionals of those path laws and the common operator sequence, so their state–trace laws and credit masses coincide. Consequently the two exact expected update maps agree at every common parameter pair. The common initialization then propagates this equality through the two critic recursions by induction.

Step 1: representative path law.

$$\mathcal{L}(\bar{X}_0) = \mathcal{L}(\widetilde{X}_0) = u_0, \tag{D.206}$$

$$\mathcal{L}(\bar{X}_{t+1} \mid \bar{X}_{0:t}) = K_t(\cdot \mid \bar{X}_t), \qquad \mathcal{L}(\widetilde{X}_{t+1} \mid \widetilde{X}_{0:t}) = K_t(\cdot \mid \widetilde{X}_t). \tag{D.207}$$

Hence, by induction,

$$\mathcal{L}(\bar{X}_0, \dots, \bar{X}_t) = \mathcal{L}(\widetilde{X}_0, \dots, \widetilde{X}_t), \qquad 0 \le t \le T. \tag{D.208}$$

Step 2: state–trace and credit identity. By Eqs. (D.187), (D.189), and (D.208),

$$\mathcal{L}(\bar{X}_0, \dots, \bar{X}_t, e_t^{\mathrm{live}}) = \mathcal{L}(\widetilde{X}_0, \dots, \widetilde{X}_t, e_t^{\mathrm{frozen}}), \qquad 0 \le t \le T. \tag{D.209}$$

$$\mathcal{L}(\bar{X}_0, \dots, \bar{X}_{t+1}, e_t^{\text{live}}) = \mathcal{L}(\widetilde{X}_0, \dots, \widetilde{X}_{t+1}, e_t^{\text{frozen}}), \qquad 0 \le t < T. \tag{D.210}$$

$$m_t^{\text{live}}(x) = m_t^{\text{frozen}}(x), \qquad x \in \mathcal{X}. \tag{D.211}$$

Step 3: same-parameter update identity. By Eqs. (D.192)–(D.195) and (D.210),

$$\Delta_t^{\text{live}}(w, w_-) = \Delta_t^{\text{frozen}}(w, w_-), \qquad (w, w_-) \in \mathbb{R}^d \times \mathbb{R}^d, \quad 0 \le t < T. \tag{D.212}$$

Step 4: learner induction. By comparison setting D-T4-A4,

$$w_{-1}^{\text{live}} = w_{-1}^{\text{frozen}}, \qquad w_0^{\text{live}} = w_0^{\text{frozen}}. \tag{D.213}$$

Using Eqs. (D.197) and (D.212),

$$w_t^{\text{live}} = w_t^{\text{frozen}}, \qquad 0 \le t \le T. \tag{D.214}$$

By Eq. (D.199),

$$V_t^{\text{live}}(x) = V_t^{\text{frozen}}(x), \qquad 0 \le t \le T, \quad x \in \mathcal{X}. \tag{D.215}$$

The two execution orders share one deterministic mean-field environment. Their matched path laws and Dutch constructions give identical exact expected updates; common initialization then gives identical critic trajectories. □

Remark 10. *Theorem 4 is an exact execution-order equivalence at the deterministic mean-field level for the matched exact-expectation Dutch evaluator defined above; it is not an equivalence claim between arbitrary policy-evaluation algorithms. It does not assert that the common learned critic equals the exact Bellman policy value. A single sampled representative trajectory is only a Monte Carlo estimator of the frozen expected update and need not generate the same random adaptive parameter path. The equality also relies on using the same deterministic population Dutch operator A_t in both protocols; replacing it by a random single-trajectory rank-one operator defines a different learner.*

D.6 Repeated-Episode Mean-Field Value-Convergence Theorem 5

Explanation. The fixed-window results in Secs. D.2–D.4 compare finite-population and mean-field learners while the episode horizon is held fixed. We now study a different limit: the deterministic mean-field environment is repeated across training episodes, the policy remains fixed, and the episodewise learning rate decreases with the episode index. The goal is to connect the exact-expectation mean-field Dutch evaluator to the classical linear TD(λ) fixed point. The proof first identifies the ordinary TD(λ) drift on a finite time-augmented state space, then shows that one Dutch episode differs from one mean TD step only by a second-order term in the episode stepsize, and finally transfers the classical stable recursion to the repeated Dutch learner.

Formal construction. Fix an episode horizon $T \ge 1$ and let $k = 0, 1, \dots$ index repeated evaluation episodes. The deterministic within-episode mean-field flow is denoted by $u_{0:T}$ and the closed-loop representative kernel by

$$K_t(x' \mid x) := P_t(x' \mid x, u_t), \qquad 0 \le t < T. \tag{D.216}$$

Let $\bar{X}_0 \sim u_0$ and

$$\mathcal{L}(\bar{X}_{t+1} \mid \bar{X}_0, \dots, \bar{X}_t) = K_t(\cdot \mid \bar{X}_t), \qquad 0 \le t < T. \tag{D.217}$$

Write

$$\varphi_t := \phi_t(\bar{X}_t), \qquad C_t := \sum_{x \in \mathcal{X}} u_t(x) \phi_t(x) \phi_t(x)^\top. \tag{D.218}$$

$$r_t^u(x) := r_t(x, u_t), \qquad 0 \le t < T. \tag{D.219}$$

Define the visited time-state support and its normalized occupancy weights by

$$\mathcal{Y} := \{(t, x) : 0 \le t < T,\ u_t(x) > 0\}, \qquad \nu(t, x) := \frac{u_t(x)}{T}. \tag{D.220}$$

Let $n_{\text{ep}} := |\mathcal{Y}|$, let $\Phi \in \mathbb{R}^{n_{\text{ep}} \times d}$ have row $\phi_t(x)^\top$ at index $(t, x) \in \mathcal{Y}$, let $D_\nu := \operatorname{diag}(\nu)$, and set

$$G_\nu := \Phi^\top D_\nu \Phi = \frac{1}{T} \sum_{t=0}^{T-1} C_t. \tag{D.221}$$

Classical TD(λ) input. We use the standard on-policy linear TD(λ) projected-fixed-point framework from Sutton and Barto Sutton & Barto (2018), particularly Sec. 12.2 and the diminishing-stepsize conditions in Eq. (2.7). The projected-operator formulation and episodic weighted analysis are classical Tsitsiklis & Van Roy (1997); Bertsekas & Tsitsiklis (1996). We verify below that the present terminal time-augmented model satisfies the required weighted contraction and identifiability properties. The value-error bound is derived below from the same contraction and projection geometry.

Time-augmented TD construction. On $\mathcal{Y}$, define the substochastic time-augmented transition matrix

$$\mathsf{P}\big((t,x),(s,y)\big) := \begin{cases} K_t(y \mid x), & s = t+1 < T, \\ 0, & \text{otherwise}, \end{cases} \tag{D.222}$$

and define $\mathbf{r}(t,x) := r_t^u(x)$. Since $\mathsf{P}^T = 0$,

$$\mathcal{T}_\lambda f := f + (I - \beta\mathsf{P})^{-1}(\mathbf{r} + \gamma\mathsf{P}f - f) = (I - \beta\mathsf{P})^{-1}[\mathbf{r} + \gamma(1-\lambda)\mathsf{P}f]. \tag{D.223}$$

The true value vector on $\mathcal{Y}$ is

$$v^\pi = (I - \gamma\mathsf{P})^{-1}\mathbf{r}. \tag{D.224}$$

Define the ordinary accumulating trace

$$\ell_{-1}^\lambda = 0, \qquad \ell_t^\lambda = \varphi_t + \beta\ell_{t-1}^\lambda, \qquad 0 \le t < T, \tag{D.225}$$

and the fixed-parameter whole-episode mean TD drift

$$\begin{aligned} h_\lambda(\theta) &:= \sum_{t=0}^{T-1} \mathbb{E}[\ell_t^\lambda \delta_t(\theta)] = b_\lambda - H_\lambda\theta, \\ H_\lambda &:= T\Phi^\top D_\nu (I - \beta\mathsf{P})^{-1}(I - \gamma\mathsf{P})\Phi, \\ b_\lambda &:= T\Phi^\top D_\nu (I - \beta\mathsf{P})^{-1}\mathbf{r}, \end{aligned} \tag{D.226}$$

where

$$\delta_t(\theta) := r_t^u(\bar{X}_t) + \gamma\chi_{t+1}\theta^\top\varphi_{t+1} - \theta^\top\varphi_t. \tag{D.227}$$

Let

$$\Pi_\nu := \Phi G_\nu^{-1}\Phi^\top D_\nu. \tag{D.228}$$

Repeated-episode setting D-R1 (Deterministic fixed-policy mean-field environment). Every training episode repeats the same deterministic initial law, fixed policy, time-inhomogeneous transition kernels, deterministic stage rewards, and time-state feature representation. The critic does not alter the physical population flow, and every learner increment is formed from the exact representative-agent expectation rather than from a single sampled representative trajectory.

Mathematical form of D-R1.

$$\begin{aligned} u_0 &\in \mathcal{P}(\mathcal{X}), \\ u_{t+1}(x') &= \sum_{x\in\mathcal{X}} u_t(x) K_t(x' \mid x), \\ K_t(x' \mid x) &= P_t(x' \mid x, u_t), \\ \mathbb{P}_u := \mathcal{L}(\bar{X}_{0:T}), &\qquad \mathbb{E}[\,\cdot\,] := \int (\cdot)\,\mathrm{d}\mathbb{P}_u. \end{aligned} \tag{D.229}$$

Because the episode has finite state-time support, the feature and realized mean-field reward bounds are automatic; define

$$B_\phi := \max_{0\le t\le T}\max_{x\in\mathcal{X}} \|\phi_t(x)\|_2 < \infty, \qquad B_r := \max_{0\le t<T}\max_{x\in\mathcal{X}} |r_t^u(x)| < \infty. \tag{D.230}$$

Repeated-episode setting D-R2 (Terminal and trace convention). The episode has a genuine terminal time, the terminal value bootstrap is zero, and the discount and trace parameters are fixed across all repeated episodes. The terminal convention is the same indicator convention used in the main paper; no additional condition on the terminal feature vector is imposed.

Mathematical form of D-R2.

$$\begin{gathered}0 \le \gamma < 1, \qquad 0 \le \lambda \le 1, \qquad \beta := \gamma\lambda, \qquad \chi_{t+1} := \mathbf{1}_{\{t+1<T\}}, \\ v_T^\pi(x) = 0, \\ v_t^\pi(x) = r_t^u(x) + \gamma \sum_{x' \in \mathcal{X}} K_t(x' \mid x) v_{t+1}^\pi(x'), \qquad 0 \le t < T.\end{gathered} \tag{D.231}$$

Assumption D-R3 (Identifiable linear features on the visited support). Across the time-state pairs that have positive mean-field occupancy, the aggregate feature Gram matrix is positive definite. Only this aggregate matrix is required to be nonsingular; individual time-slice second-moment matrices may be singular.

Mathematical form of D-R3.

$$G_\nu \succ 0. \tag{D.232}$$

Lemma 5 (Finite-Episode TD Interface)**.** *Under repeated-episode settings D-R1 and D-R2, together with Assumption D-R3, the time-augmented model admits a unique projected lambda-return temporal-difference fixed point, and the corresponding mean temporal-difference matrix has a uniformly positive quadratic form.*

$$\begin{gathered}\kappa_\lambda := \frac{\gamma(1-\lambda)}{1-\gamma\lambda} < 1, \qquad \mu_\lambda := T(1-\kappa_\lambda)\lambda_{\min}(G_\nu) > 0, \\ \theta^\top H_\lambda \theta \ge \mu_\lambda \|\theta\|_2^2, \qquad \theta \in \mathbb{R}^d, \\ \theta_\lambda^{\mathrm{TD}} := H_\lambda^{-1} b_\lambda, \qquad \Phi\theta_\lambda^{\mathrm{TD}} = \Pi_\nu \mathcal{T}_\lambda(\Phi\theta_\lambda^{\mathrm{TD}}).\end{gathered} \tag{D.233}$$

Proof. The time-augmented transition is nonexpansive in the occupancy-weighted norm because its weighted destination mass is dominated by the current visitation weights. This yields a contraction modulus for the λ-return operator. Expanding the accumulating trace identifies the mean TD drift with the projected operator residual; the positive-definite aggregate Gram matrix then converts weighted stability into a Euclidean parameter bound.

Step 1: weighted transition bound. For a destination $(s, y) \in \mathcal{Y}$ with $1 \le s < T$, the mean-field flow equation gives

$$\sum_{(t,x)\in\mathcal{Y}} \nu(t,x)\mathsf{P}((t,x),(s,y)) = \frac{1}{T}\sum_x u_{s-1}(x)K_{s-1}(y \mid x) = \frac{u_s(y)}{T} = \nu(s,y). \tag{D.234}$$

For $s = 0$, the left-hand side is zero because there is no predecessor layer. Therefore, componentwise,

$$\nu^\top \mathsf{P} \le \nu^\top. \tag{D.235}$$

For every $f \in \mathbb{R}^{n_{\mathrm{ep}}}$, Jensen's inequality for each substochastic row then gives

$$\|\mathsf{P}f\|_\nu^2 \le \sum_i \nu_i \sum_j \mathsf{P}_{ij} f_j^2 \le \sum_j \nu_j f_j^2 = \|f\|_\nu^2. \tag{D.236}$$

Hence, with

$$\mathsf{M}_\lambda := \gamma(1-\lambda)(I - \beta\mathsf{P})^{-1}\mathsf{P}, \tag{D.237}$$

$$\|\mathsf{M}_\lambda\|_{\nu\to\nu} \le \frac{\gamma(1-\lambda)}{1-\beta} = \kappa_\lambda < 1. \tag{D.238}$$

Step 2: TD drift identity. Let

$$g_\theta := \mathbf{r} - (I - \gamma\mathsf{P})\Phi\theta. \tag{D.239}$$

Then

$$\mathbb{E}[\delta_t(\theta) \mid \bar{X}_s = x] = (\mathsf{P}^{\,t-s} g_\theta)(s,x), \qquad 0 \le s \le t < T, \tag{D.240}$$

so

$$h_\lambda(\theta) = \sum_{t=0}^{T-1} \sum_{s=0}^{t} \beta^{t-s} \mathbb{E}[\varphi_s \delta_t(\theta)] \tag{D.241}$$

$$= T\Phi^\top D_\nu \left(\sum_{j=0}^{T-1} \beta^j \mathsf{P}^j \right) g_\theta \tag{D.242}$$

$$= T\Phi^\top D_\nu (I - \beta\mathsf{P})^{-1} g_\theta = b_\lambda - H_\lambda \theta. \tag{D.243}$$

Step 3: positive quadratic form. Since

$$(I - \beta\mathsf{P})^{-1}(I - \gamma\mathsf{P}) = I - \mathsf{M}_\lambda, \tag{D.244}$$

$$\theta^\top H_\lambda \theta = T\langle \Phi\theta, (I - \mathsf{M}_\lambda)\Phi\theta \rangle_\nu \tag{D.245}$$

$$\geq T(1 - \kappa_\lambda)\|\Phi\theta\|_\nu^2 \tag{D.246}$$

$$\geq \mu_\lambda \|\theta\|_2^2. \tag{D.247}$$

Thus H_λ is nonsingular and

$$\theta_\lambda^{\mathrm{TD}} = H_\lambda^{-1} b_\lambda. \tag{D.248}$$

Moreover,

$$\Phi^\top D_\nu \left[\mathcal{T}_\lambda(\Phi\theta_\lambda^{\mathrm{TD}}) - \Phi\theta_\lambda^{\mathrm{TD}} \right] = 0, \tag{D.249}$$

which is equivalent to Eq. (D.233).

The time-augmented process is a contraction in the occupancy-weighted geometry, and the aggregate Gram condition transfers that contraction to a unique linear parameter fixed point. □

Dutch-to-TD bridge. The TD interface identifies the first-order target drift. We next compare one complete Dutch episode with one mean TD step at a generic constant stepsize. This is the bridge needed later when the episodewise stepsize becomes small.

Formal construction. Fix $\bar{\alpha} < \infty$. For $a \in [0, \bar{\alpha}]$, define

$$A_{a,t} := I_d - aC_t, \qquad \bar{e}_{a,-1}^D := 0, \qquad \bar{e}_{a,t}^D := \varphi_t + \beta A_{a,t} \bar{e}_{a,t-1}^D. \tag{D.250}$$

Starting from

$$w_{a,-1} = w_{a,0} = \theta, \tag{D.251}$$

define for $0 \leq t < T$

$$c_t(w, w_-) := (w - w_-)^\top \varphi_t, \tag{D.252}$$

$$\begin{aligned} w_{a,t+1} = w_{a,t} + a\,\mathbb{E}\Big[& \delta_t(w_{a,t}) \bar{e}_{a,t}^D \\ & + c_t(w_{a,t}, w_{a,t-1})(\bar{e}_{a,t}^D - \varphi_t) \Big]. \end{aligned} \tag{D.253}$$

Let

$$\Psi_a(\theta) := w_{a,T}. \tag{D.254}$$

Lemma 6 (Episodewise Dutch-to-TD Bridge)**.** *Under repeated-episode settings D-R1 and D-R2, one Dutch episode equals one mean lambda-return temporal-difference step up to a uniformly second-order stepsize remainder.*

$$\begin{aligned} \max_{0 \leq t < T} \|\bar{e}_{a,t}^D - \ell_t^\lambda\|_2 &\leq K_{\mathrm{tr}} a, \\ \max_{-1 \leq t \leq T} \|w_{a,t} - \theta\|_2 &\leq K_{\mathrm{loc}} a(1 + \|\theta\|_2), \\ \Psi_a(\theta) &= \theta + a h_\lambda(\theta) + \mathcal{R}_a(\theta), \\ \|\mathcal{R}_a(\theta)\|_2 &\leq K_{\mathrm{br}} a^2 (1 + \|\theta\|_2), \qquad 0 \leq a \leq \bar{\alpha}. \end{aligned} \tag{D.255}$$

Proof. The Dutch trace differs from the ordinary accumulating trace only through the factor $I - aC_t$, so the trace discrepancy is first order in a over the fixed episode. Because the path law is independent of the critic and of a, the finite within-episode learner is affine in its starting parameter with coefficients that are polynomial in a. Differentiating the full episode map at $a = 0$ gives exactly the ordinary TD(λ) drift, while Taylor's formula supplies a second-order remainder.

Step 1: trace discrepancy. From Eqs. (D.250) and (D.225),

$$\bar{e}^D_{a,t} - \ell^\lambda_t = \beta(\bar{e}^D_{a,t-1} - \ell^\lambda_{t-1}) - a\beta C_t \bar{e}^D_{a,t-1}. \tag{D.256}$$

Set

$$B_{e,\bar{\alpha}} := B_\phi \sum_{j=0}^{T-1} [\beta(1 + \bar{\alpha} B_\phi^2)]^j. \tag{D.257}$$

Then

$$\max_{0 \le t < T} \|\bar{e}^D_{a,t}\|_2 \le B_{e,\bar{\alpha}}, \tag{D.258}$$

$$\max_{0 \le t < T} \|\bar{e}^D_{a,t} - \ell^\lambda_t\|_2 \le K_{\mathrm{tr}} a, \tag{D.259}$$

where

$$K_{\mathrm{tr}} := \beta B_\phi^2 B_{e,\bar{\alpha}} \sum_{j=0}^{T-2} \beta^j, \tag{D.260}$$

with the sum equal to zero when $T = 1$.

Step 2: affine finite-episode map. The claim is proved by induction over the finite episode. At $t = -1, 0$, Eq. (D.251) gives $w_{a,t} = I_d\theta + 0$. The trace recursion (D.250) contains only additions and multiplications by $I_d - aC_t$, so every $\bar{e}^D_{a,t}$ is a finite-degree polynomial in a independent of θ. If $w_{a,t} = L_t(a)\theta + v_t(a)$ and $w_{a,t-1} = L_{t-1}(a)\theta + v_{t-1}(a)$, then both $\delta_t(w_{a,t})$ and $c_t(w_{a,t}, w_{a,t-1})$ are affine in θ with polynomial coefficients in a. Multiplying them by the polynomial trace and by the outer factor a in Eq. (D.253) preserves this form. Hence, for every $-1 \le t \le T$,

$$w_{a,t} = L_t(a)\theta + v_t(a), \tag{D.261}$$

where L_t and v_t are finite-degree polynomials in a. Setting $a = 0$ in the recursion gives

$$L_t(0) = I_d, \qquad v_t(0) = 0. \tag{D.262}$$

Hence

$$\max_{-1 \le t \le T} \|w_{a,t} - \theta\|_2 \le K_{\mathrm{loc}} a(1 + \|\theta\|_2) \tag{D.263}$$

for

$$K_{\mathrm{loc}} := \max_{-1 \le t \le T} \max_{s \in [0,\bar{\alpha}]} \left(\|L_t'(s)\|_{\mathrm{op}} + \|v_t'(s)\|_2 \right). \tag{D.264}$$

Step 3: first derivative and remainder. At $a = 0$,

$$\bar{e}^D_{0,t} = \ell^\lambda_t, \qquad w_{0,t} = \theta, \qquad c_t(\theta, \theta) = 0. \tag{D.265}$$

Therefore

$$\left. \frac{\partial \Psi_a(\theta)}{\partial a} \right|_{a=0} = h_\lambda(\theta). \tag{D.266}$$

Set

$$K_{\mathrm{br}} := \frac{1}{2} \max_{s \in [0,\bar{\alpha}]} \left(\|L_T''(s)\|_{\mathrm{op}} + \|v_T''(s)\|_2 \right). \tag{D.267}$$

Taylor's formula gives

$$\mathcal{R}_a(\theta) = \int_0^a (a - s)[L_T''(s)\theta + v_T''(s)] \, \mathrm{d}s, \tag{D.268}$$

$$\|\mathcal{R}_a(\theta)\|_2 \le K_{\mathrm{br}} a^2 (1 + \|\theta\|_2). \tag{D.269}$$

On a fixed finite episode, the Dutch deformation is first order in the stepsize, while the episode-end map has the ordinary TD(λ) drift as its derivative at zero. The resulting discrepancy is therefore second order. □

Formal construction. During episode k, set

$$A_{k,t} := I_d - \alpha_k C_t, \qquad \bar{e}^D_{k,t} := \varphi_t + \beta A_{k,t} \bar{e}^D_{k,t-1}, \qquad 0 \le t < T, \tag{D.270}$$

and update

$$w_{k,t+1} = w_{k,t} + \alpha_k \, \mathbb{E}\Big[\delta_t(w_{k,t}) \bar{e}^D_{k,t} + c_t(w_{k,t}, w_{k,t-1})(\bar{e}^D_{k,t} - \varphi_t)\Big]. \tag{D.271}$$

Define the analytic mean TD comparator

$$\vartheta_{k+1} := \vartheta_k + \alpha_k h_\lambda(\vartheta_k), \qquad \vartheta_0 := \theta_0, \tag{D.272}$$

and the learned value after k completed episodes by

$$\widehat{V}_k(t, x) := \theta_k^\top \phi_t(x). \tag{D.273}$$

Episodewise stepsize setting. During episode k, the learner uses one constant stepsize α_k; the stepsize may change only between episodes.

Assumption D-R4 (Diminishing episodewise stepsizes). The episodewise stepsizes are positive, vanish asymptotically, and have infinite total mass. Because the repeated-episode update is an exact-expectation deterministic recursion, square summability is not required.

Mathematical form of D-R4.

$$\alpha_k > 0, \qquad \alpha_k \longrightarrow 0, \qquad \sum_{k=0}^{\infty} \alpha_k = \infty, \tag{D.274}$$

Episode reset convention. The critic parameter is carried from one episode to the next, while the Dutch trace and previous-parameter convention reset at the beginning of every episode:

$$w_{k,-1} = w_{k,0} = \theta_k, \qquad \bar{e}^D_{k,-1} = 0, \qquad \theta_{k+1} = w_{k,T}, \qquad \theta_0 = w_0. \tag{D.275}$$

Lemma 7 (Repeated-Episode Mean-Field Convergence)**.** *Under repeated-episode settings D-R1 and D-R2, Assumptions D-R3 and D-R4, and the episode reset convention, the repeated mean-field Dutch evaluator converges to the unique projected temporal-difference fixed point established above. Every within-episode critic parameter approaches the same limit.*

$$\begin{aligned} \theta_k &\longrightarrow \theta_\lambda^{\mathrm{TD}} = H_\lambda^{-1} b_\lambda, \\ \max_{-1 \le t \le T} \|w_{k,t} - \theta_\lambda^{\mathrm{TD}}\|_2 &\longrightarrow 0, \\ \|\theta_k - \vartheta_k\|_2 &\longrightarrow 0. \end{aligned} \tag{D.276}$$

Proof. The bridge lemma writes each episode endpoint as the stable linear TD step plus a second-order perturbation. Positive definiteness of the TD drift gives a strict first-order decrease when the stepsize is sufficiently small. The second-order perturbation is $o(\alpha_k)$ because $\alpha_k \to 0$, so it can be absorbed into the stable first-order drift. This yields boundedness and convergence of the episode-end sequence; the within-episode displacement vanishes for the same reason.

Assumption D-R4 gives $\alpha_k \to 0$, so the stepsize sequence $(\alpha_k)_{k \ge 0}$ is bounded. Set

$$\bar{\alpha} := \sup_{k \ge 0} \alpha_k < \infty, \qquad 0 < \alpha_k \le \bar{\alpha} \quad \forall k. \tag{D.277}$$

Hence Lemma 6 applies uniformly to every episode with this fixed $\bar{\alpha}$, giving

$$\theta_{k+1} = \theta_k + \alpha_k(b_\lambda - H\theta_k) + \mathcal{R}_{\alpha_k}(\theta_k). \tag{D.278}$$

Set

$$\begin{gathered} \theta_* := \theta_\lambda^{\mathrm{TD}}, \qquad H := H_\lambda, \qquad \mu := \mu_\lambda, \qquad L_H := \|H\|_{\mathrm{op}}, \\ a_* := \min\left\{ \frac{\mu}{L_H^2}, \frac{1}{\mu}, \frac{\mu}{4(1 + K_{\mathrm{br}})} \right\}, \\ c := \frac{\mu}{4}, \qquad C := K_{\mathrm{br}}(1 + \|\theta_*\|_2), \qquad d_k := \|\theta_k - \theta_*\|_2. \end{gathered} \tag{D.279}$$

By Assumption D-R4, $\alpha_k \to 0$. Choose k_* so that

$$\alpha_k \le a_*, \qquad \forall\, k \ge k_*. \tag{D.280}$$

Step 1: stable linear part. By Lemma 5,

$$\|(I_d - aH)z\|_2^2 \le (1 - 2\mu a + L_H^2 a^2)\|z\|_2^2. \tag{D.281}$$

By Eq. (D.279), for $k \ge k_*$,

$$\|I_d - \alpha_k H\|_{\mathrm{op}} \le 1 - \frac{\mu\alpha_k}{2}, \qquad K_{\mathrm{br}}\alpha_k \le \frac{\mu}{4}. \tag{D.282}$$

Step 2: perturbed error recurrence. By Eq. (D.255),

$$\|\mathcal{R}_{\alpha_k}(\theta_k)\|_2 \le K_{\mathrm{br}}\alpha_k^2(d_k + 1 + \|\theta_*\|_2). \tag{D.283}$$

Hence, by Eq. (D.279),

$$d_{k+1} \le (1 - c\alpha_k)d_k + C\alpha_k^2, \qquad k \ge k_*. \tag{D.284}$$

Consequently

$$d_k \le \max\left\{d_{k_*}, \frac{Ca_*}{c}\right\}, \qquad k \ge k_*. \tag{D.285}$$

Step 3: convergence. Fix $\varepsilon > 0$. Because $\alpha_k \to 0$, choose $K_\varepsilon \ge k_*$ such that

$$C\alpha_k \le \frac{c\varepsilon}{2}, \qquad k \ge K_\varepsilon. \tag{D.286}$$

Whenever $k \ge K_\varepsilon$ and $d_k \ge \varepsilon$, Eq. (D.284) gives

$$d_{k+1} \le d_k - c\alpha_k d_k + C\alpha_k^2 \le d_k - \frac{c\varepsilon}{2}\alpha_k. \tag{D.287}$$

If $d_k \ge \varepsilon$ held for every $k \ge K_\varepsilon$, summing (D.287) and using $\sum_k \alpha_k = \infty$ would force d_k eventually negative, a contradiction. Hence the sequence enters the interval $[0, \varepsilon]$. Once $d_k \le \varepsilon$ with $k \ge K_\varepsilon$,

$$d_{k+1} \le (1 - c\alpha_k)\varepsilon + C\alpha_k^2 \le \varepsilon, \tag{D.288}$$

because Eq. (D.286) is stronger than $C\alpha_k \le c\varepsilon$. Therefore $d_k \le \varepsilon$ eventually. Since $\varepsilon > 0$ is arbitrary,

$$d_k \longrightarrow 0, \qquad \theta_k \longrightarrow \theta_*. \tag{D.289}$$

Lemma 6 and Eq. (D.285) give

$$\max_{-1 \le t \le T} \|w_{k,t} - \theta_k\|_2 \le K_{\mathrm{loc}}\alpha_k(1 + \|\theta_k\|_2) \longrightarrow 0. \tag{D.290}$$

For the comparator,

$$\vartheta_{k+1} - \theta_* = (I_d - \alpha_k H)(\vartheta_k - \theta_*), \tag{D.291}$$

so the same product estimate yields

$$\vartheta_k \longrightarrow \theta_*. \tag{D.292}$$

Thus

$$\|\theta_k - \vartheta_k\|_2 \longrightarrow 0. \tag{D.293}$$

The repeated Dutch learner is a diminishing-stepsize perturbation of a stable linear TD recursion. Its second-order remainder is negligible relative to the first-order stable drift because $\alpha_k \to 0$, while $\sum_k \alpha_k = \infty$ prevents the stable drift from dying out before the error reaches zero. □

Formal construction. For a vector f on $\mathcal{Y}$, define

$$\|f\|_\nu^2 := f^\top D_\nu f, \tag{D.294}$$

and define

$$\overline{\mathrm{VE}}_\nu(\theta) := \|\Phi\theta - v^\pi\|_\nu^2, \qquad \varepsilon_{\mathrm{app}}^2 := \min_{\theta \in \mathbb{R}^d} \overline{\mathrm{VE}}_\nu(\theta). \tag{D.295}$$

Projected-TD value-error bound. The value-error estimate needed below follows from the projected fixed-point identity in Lemma 5; it is not an additional assumption. Let $v_\lambda := \Phi\theta_\lambda^{\mathrm{TD}}$. Since $v_\lambda = \Pi_\nu \mathcal{T}_\lambda v_\lambda$, $v^\pi = \mathcal{T}_\lambda v^\pi$, and Π_ν is the D_ν-orthogonal projection onto $\mathrm{range}(\Phi)$,

$$\|v_\lambda - v^\pi\|_\nu^2 = \|v_\lambda - \Pi_\nu v^\pi\|_\nu^2 + \|\Pi_\nu v^\pi - v^\pi\|_\nu^2 \tag{D.296}$$

$$\leq \kappa_\lambda^2 \|v_\lambda - v^\pi\|_\nu^2 + \|\Pi_\nu v^\pi - v^\pi\|_\nu^2. \tag{D.297}$$

Therefore

$$\|\Phi\theta_\lambda^{\mathrm{TD}} - v^\pi\|_\nu^2 \leq \frac{\|\Pi_\nu v^\pi - v^\pi\|_\nu^2}{1-\kappa_\lambda^2} \leq \frac{\|\Pi_\nu v^\pi - v^\pi\|_\nu^2}{1-\kappa_\lambda}. \tag{D.298}$$

Moreover, orthogonal projection gives $\|\Pi_\nu v^\pi - v^\pi\|_\nu^2 = \varepsilon_{\mathrm{app}}^2$.

Theorem 5 (Mean-Field Value Convergence and Error Bound)**.** *Under repeated-episode settings D-R1 and D-R2, Assumptions D-R3 and D-R4, and the episode reset convention, the repeated mean-field Dutch evaluator satisfies the projected temporal-difference asymptotic value-error bound, and under realizability its learned value converges to the exact fixed-policy mean-field value on the visited support.*

$$\begin{aligned} &\lim_{k\to\infty} \overline{\mathrm{VE}}_\nu(\theta_k) = \overline{\mathrm{VE}}_\nu(\theta_\lambda^{\mathrm{TD}}) \leq \frac{1-\gamma\lambda}{1-\gamma}\,\varepsilon_{\mathrm{app}}^2, \\ &\varepsilon_{\mathrm{app}} = 0 \implies \max_{(t,x)\in\mathcal{Y}} |\widehat{V}_k(t,x) - v_t^\pi(x)| \longrightarrow 0. \end{aligned} \tag{D.299}$$

Proof. The TD interface lemma identifies the same projected fixed point reached by the repeated Dutch learner. The orthogonal-projection estimate derived above bounds the distance between that fixed point and the true value, while convergence of the repeated learner transfers that bound to its asymptotic value error.

By Eq. (D.298) and Lemma 5,

$$\|\Phi\theta_\lambda^{\mathrm{TD}} - v^\pi\|_\nu^2 \leq \frac{1}{1-\kappa_\lambda}\varepsilon_{\mathrm{app}}^2 = \frac{1-\gamma\lambda}{1-\gamma}\varepsilon_{\mathrm{app}}^2. \tag{D.300}$$

Lemma 7 gives

$$\lim_{k\to\infty} \overline{\mathrm{VE}}_\nu(\theta_k) = \overline{\mathrm{VE}}_\nu(\theta_\lambda^{\mathrm{TD}}). \tag{D.301}$$

If $\varepsilon_{\mathrm{app}} = 0$, then

$$\Phi\theta_\lambda^{\mathrm{TD}} = v^\pi, \tag{D.302}$$

so

$$\max_{(t,x)\in\mathcal{Y}} |\widehat{V}_k(t,x) - v_t^\pi(x)| \leq B_\phi \|\theta_k - \theta_\lambda^{\mathrm{TD}}\|_2 \longrightarrow 0. \tag{D.303}$$

The asymptotic limit is the classical projected TD(λ) solution, so its approximation error is inherited directly; the Dutch-specific second-order perturbation vanishes under the episodewise stepsize schedule. □

Remark 11. *Lemma 7 and Theorem 5 concern the exact-expectation deterministic mean-field model with fixed episode horizon and fixed policy as the number of training episodes tends to infinity. They do not provide a finite-population training guarantee or justify exchanging the limits $N \to \infty$ and $k \to \infty$. A sampled representative learner would require a separate stochastic-approximation analysis. The value-error statement is restricted to the visited time-state support $\mathcal{Y}$. At $\lambda = 1$, the λ-return operator is the full finite-horizon return operator and the limiting projected solution is the corresponding weighted least-squares value approximation.*

E Extensions of DoubleSWEET to Eligibility-Trace Least-Squares Methods

E.1 Conventional-Trace Extension Setting

This section fixes the notation for the extension from Dutch traces to conventional accumulating eligibility traces and then to least-squares policy-evaluation estimators. The physical population process is unchanged. In the deterministic mean-field system, u_t is the population law, $K_t(x' \mid x) := P_t(x' \mid x, u_t)$ is the representative transition kernel, and

$$\Gamma_t(y, x) := u_{t-1}(y)K_{t-1}(x \mid y), \qquad 1 \le t \le T. \tag{E.1}$$

In the finite population, U_t^N and Γ_t^N denote the empirical state and transition-pair laws defined in Sec. D.1. By Lemma 3, on every fixed horizon,

$$\max_{0\le t\le T} \mathbb{E}\|U_t^N - u_t\|_1 = O_T(N^{-1/2}), \qquad \max_{1\le t\le T} \mathbb{E}\|\Gamma_t^N - \Gamma_t\|_1 = O_T(N^{-1/2}). \tag{E.2}$$

Write

$$0 \le \gamma \le 1, \qquad 0 \le \lambda \le 1, \qquad \beta := \gamma\lambda. \tag{E.3}$$

The representative Dutch trace from Appendix C obeys

$$\bar{e}_t^D = \phi_t(\bar{X}_t) + \beta A_t \bar{e}_{t-1}^D, \tag{E.4}$$

whereas the conventional accumulating eligibility trace introduced below obeys the same temporal recursion with the propagation operator specialized to I_d. Thus the extension uses the same state-wise population-credit transport mechanism but without the Dutch population operator in the trace recursion.

Table 3: Trace, credit, and least-squares objects used in Appendix E.

Object	Meaning
ℓ_t^λ	Representative mean-field conventional accumulating trace.
$z_t^\lambda(x)$	Mean-field conditional conventional trace at an occupied state.
$m_t^\lambda(x)$	Mean-field unnormalized conventional credit mass.
$\ell_t^{i,N,\lambda}$	Private conventional trace of finite-population agent i.
$m_t^{\mathrm{priv},N,\lambda}(x)$	State-aggregated private conventional credit mass.
$\widehat{m}_t^{N,\lambda}(x)$	Compressed finite-population conventional credit mass.
$\widehat{z}_t^{N,\lambda}(x)$	Compressed occupied-state conventional readout.
$H_\lambda^{N,\mathrm{priv}}$, $g_\lambda^{N,\mathrm{priv}}$	Private ridge-LSTD(λ) sufficient statistics.
$H_\lambda^{N,C}$, $g_\lambda^{N,C}$	DoubleSWEET-compressed ridge-LSTD(λ) sufficient statistics.
$w_{\lambda,\eta}^{N,\mathrm{priv}}$	Private ridge-LSTD(λ) estimator.
$w_{\lambda,\eta}^{N,C}$	DoubleSWEET ridge-LSTD(λ) estimator.

Throughout Appendix E, the superscript C denotes a compressed estimator, while the absence of C on $\widehat{m}_t^{N,\lambda}$ and $\widehat{z}_t^{N,\lambda}$ follows the state-wise credit notation already used for DoubleSWEET. The symbol $\eta > 0$ is reserved for the ridge coefficient introduced in the ridge-LSTD construction of Sec. E.3.

Overview of the LSTD(λ) Algorithm. We first briefly review LSTD(λ), which is the conventional eligibility-trace method extended later in this appendix. LSTD(λ) is naturally formulated through a projected λ-return fixed-point condition rather than by directly minimizing the squared one-step TD error Boyan (2002). For the linear value function $V_w(t, x) = w^\top \phi_t(x)$, the goal is to find a parameter w whose projected λ-return residual vanishes, i.e., $\Pi\mathcal{T}_\lambda V_w - V_w = 0$, where $\mathcal{T}_\lambda$ denotes the λ-return operator and Π denotes projection onto the linear function class.

Consider one transition (X_t, R_{t+1}, X_{t+1}), and write $\phi_t := \phi(t, X_t)$ and $\phi_{t+1} := \phi(t+1, X_{t+1})$. Under the episodic terminal convention used below, $\phi_T = 0$. For a fixed parameter w, the one-step TD residual is $\delta_t(w) = R_{t+1} + \gamma w^\top \phi_{t+1} - w^\top \phi_t = R_{t+1} - (\phi_t - \gamma\phi_{t+1})^\top w$.

LSTD(λ) propagates the TD residual to previously visited features through the conventional accumulating eligibility trace

$$\ell_{-1}^{\lambda} = 0, \qquad \ell_t^{\lambda} = \phi_t + \gamma\lambda\ell_{t-1}^{\lambda}. \tag{E.5}$$

The projected fixed-point condition leads to the eligibility-weighted TD orthogonality condition

$$\sum_{t=0}^{T-1} \ell_t^{\lambda}\delta_t(w) = 0. \tag{E.6}$$

Substituting $\delta_t(w)$ and rearranging gives

$$\left[\sum_{t=0}^{T-1} \ell_t^{\lambda}(\phi_t - \gamma\phi_{t+1})^{\top}\right] w = \sum_{t=0}^{T-1} \ell_t^{\lambda} R_{t+1}. \tag{E.7}$$

Therefore, define the LSTD(λ) sufficient statistics

$$H_{\lambda} := \sum_{t=0}^{T-1} \ell_t^{\lambda}(\phi_t - \gamma\phi_{t+1})^{\top}, \qquad g_{\lambda} := \sum_{t=0}^{T-1} \ell_t^{\lambda} R_{t+1}. \tag{E.8}$$

The ridge-regularized LSTD(λ) estimator used in this appendix is then obtained from

$$(H_{\lambda} + \eta I_d) w_{\lambda,\eta} = g_{\lambda}, \qquad \eta > 0. \tag{E.9}$$

Thus, unlike an online TD method that updates w after every transition, LSTD(λ) maintains the eligibility trace ℓ_t^{λ}, accumulates the sufficient statistics H_{λ} and g_{λ} throughout the episode, and solves one linear system at the end. In a finite population, a direct implementation maintains one such eligibility trace for every agent. The remainder of this appendix shows how these agent-wise traces can instead be represented by state-wise compressed credit, leading to DoubleSWEET-LSTD(λ).

The special case $\lambda = 1$ reduce to Ridge MC-LS under the episodic terminal convention.

E.2 Conventional Eligibility-Trace Compression Theorem 6

Explanation. A conventional accumulating eligibility trace stores discounted historical credit but does not contain the population Dutch operator. In the deterministic mean-field system, conditioning this trace on the current state yields an exact predecessor recursion. Multiplying the conditional trace by the current occupancy removes the predecessor Bayes denominator and gives a forward credit-mass recursion. Replacing the deterministic occupancy and pair flow by their empirical finite-population counterparts then yields a one-pass state-wise compressed recursion. The finite-population compressed and private credit masses can be compared through the same deterministic mean-field target, so the physical consistency result of Sec. D.2 transfers directly to conventional traces.

Formal construction. Let the representative conventional trace satisfy

$$\ell_{-1}^{\lambda} = 0, \qquad \ell_t^{\lambda} = \phi_t(\bar{X}_t) + \beta\ell_{t-1}^{\lambda}, \qquad 0 \le t \le T. \tag{E.10}$$

For an occupied mean-field state, define

$$z_t^{\lambda}(x) := \mathbb{E}[\ell_t^{\lambda} \mid \bar{X}_t = x], \qquad x \in \mathcal{S}_t := \{x : u_t(x) > 0\}, \tag{E.11}$$

and on the full state space define

$$m_t^{\lambda}(x) := \mathbb{E}[\ell_t^{\lambda}\mathbf{1}_{\{\bar{X}_t = x\}}]. \tag{E.12}$$

Mean-field conventional-trace setting. The representative process follows the deterministic population flow under the fixed policy:

$$\begin{aligned} \bar{X}_0 &\sim u_0, \\ \mathbb{P}(\bar{X}_{t+1} = x' \mid \bar{X}_t = x) &= K_t(x' \mid x), \qquad 0 \le t < T, \\ u_{t+1}(x') &= \sum_{x \in \mathcal{X}} u_t(x) K_t(x' \mid x), \qquad 0 \le t < T. \end{aligned} \tag{E.13}$$

Because the standing state space and horizon are finite, set

$$B_\phi := \max_{0\le t\le T}\max_{x\in\mathcal{X}} \|\phi_t(x)\|_2 < \infty, \qquad 0 \le \lambda \le 1, \qquad \beta = \gamma\lambda. \tag{E.14}$$

The trace expansion under this setting gives

$$\|\ell_t^\lambda\|_2 \le B_{\lambda,T} := B_\phi \sum_{j=0}^{T} \beta^j < \infty, \qquad 0 \le t \le T, \tag{E.15}$$

so the conditional expectations in (E.11) are well defined.

Exact conventional-trace state-wise closure. Under the mean-field conventional-trace setting above, the mean-field conditional conventional trace and its unnormalized credit mass satisfy

$$\begin{aligned}
z_0^\lambda(x) &= \phi_0(x), && x \in \mathcal{S}_0,\\
z_t^\lambda(x) &= \phi_t(x) + \beta \sum_{y\in\mathcal{S}_{t-1}} B_t(y \mid x) z_{t-1}^\lambda(y), && x \in \mathcal{S}_t,\ 1 \le t \le T,\\
m_0^\lambda(x) &= u_0(x)\phi_0(x), && x \in \mathcal{X},\\
m_t^\lambda(x) &= u_t(x)\phi_t(x) + \beta \sum_{y\in\mathcal{X}} K_{t-1}(x \mid y) m_{t-1}^\lambda(y), && x \in \mathcal{X},\ 1 \le t \le T,
\end{aligned} \tag{E.16}$$

where

$$B_t(y \mid x) := \frac{u_{t-1}(y)K_{t-1}(x \mid y)}{u_t(x)}, \qquad x \in \mathcal{S}_t. \tag{E.17}$$

Indeed, by the Markov property,

$$\mathbb{E}[\ell_{t-1}^\lambda \mathbf{1}_{\{\bar{X}_t = x\}}] = \mathbb{E}[\ell_{t-1}^\lambda K_{t-1}(x \mid \bar{X}_{t-1})] \tag{E.18}$$

$$= \sum_{y\in\mathcal{X}} K_{t-1}(x \mid y) m_{t-1}^\lambda(y), \tag{E.19}$$

$$m_t^\lambda(x) = u_t(x)\phi_t(x) + \beta \sum_{y\in\mathcal{X}} K_{t-1}(x \mid y) m_{t-1}^\lambda(y). \tag{E.20}$$

For $x \in \mathcal{S}_t$, dividing the last identity by $u_t(x)$ gives

$$z_t^\lambda(x) = \phi_t(x) + \beta \sum_{y\in\mathcal{S}_{t-1}} \frac{u_{t-1}(y)K_{t-1}(x \mid y)}{u_t(x)} z_{t-1}^\lambda(y), \tag{E.21}$$

and the $t = 0$ identities follow from $\ell_0^\lambda = \phi_0(\bar{X}_0)$.

Formal construction of the finite-population traces. For each agent, define

$$\ell_{-1}^{i,N,\lambda} = 0, \qquad \ell_t^{i,N,\lambda} = \phi_t(X_t^{i,N}) + \beta \ell_{t-1}^{i,N,\lambda}, \qquad 0 \le t \le T, \tag{E.22}$$

and aggregate the private credit by state:

$$m_t^{\mathrm{priv},N,\lambda}(x) := \frac{1}{N} \sum_{i=1}^{N} \ell_t^{i,N,\lambda} \mathbf{1}_{\{X_t^{i,N} = x\}}. \tag{E.23}$$

On an empirically occupied state, define the within-state private average

$$z_t^{\mathrm{priv},N,\lambda}(x) := \frac{m_t^{\mathrm{priv},N,\lambda}(x)}{U_t^N(x)}. \tag{E.24}$$

The compressed recursion is initialized by

$$\widehat{m}_0^{N,\lambda}(x) := U_0^N(x)\phi_0(x), \tag{E.25}$$

with occupied-state readout

$$\widehat{z}_t^{N,\lambda}(x) := \frac{\widehat{m}_t^{N,\lambda}(x)}{U_t^N(x)}, \qquad U_t^N(x) > 0, \tag{E.26}$$

and update

$$\widehat{m}_{t+1}^{N,\lambda}(x') = U_{t+1}^N(x')\phi_{t+1}(x') + \beta \sum_{y\in\mathcal{X}} \Gamma_{t+1}^N(y,x')\widehat{z}_t^{N,\lambda,0}(y), \tag{E.27}$$

where the zero extensions are

$$\widehat{z}_t^{N,\lambda,0}(x) := \begin{cases} \widehat{z}_t^{N,\lambda}(x), & U_t^N(x) > 0, \\ 0, & U_t^N(x) = 0, \end{cases} \qquad z_t^{\mathrm{priv},N,\lambda,0}(x) := \begin{cases} z_t^{\mathrm{priv},N,\lambda}(x), & U_t^N(x) > 0, \\ 0, & U_t^N(x) = 0, \end{cases} \tag{E.28}$$

and, for the deterministic mean-field field,

$$z_t^{\lambda,0}(x) := \begin{cases} z_t^{\lambda}(x), & u_t(x) > 0, \\ 0, & u_t(x) = 0. \end{cases} \tag{E.29}$$

These are algebraic zero extensions only; no conditional-expectation interpretation is assigned to an empty state.

Algorithm 3 One-pass state-wise compression of conventional accumulating traces

1: Compute U_0^N, cache $\phi_0(x)$ on the occupied support, and set $\widehat{m}_0^{N,\lambda}(x) = U_0^N(x)\phi_0(x)$.
2: **for** $t = 0, \dots, T-1$ **do**
3: Treat U_t^N, $\widehat{m}_t^{N,\lambda}$, and the current feature cache as read-only during the scan.
4: Initialize sparse destination counts $n[x] = 0$, credit accumulators $a[x] = 0 \in \mathbb{R}^d$, and an empty next-feature cache.
5: **for** each observed transition $y = X_t^{i,N} \to x = X_{t+1}^{i,N}$ **do**
6: $z \leftarrow \widehat{m}_t^{N,\lambda}(y)/U_t^N(y)$.
7: If x is new, evaluate and cache $\phi_{t+1}(x)$ once.
8: $n[x] \leftarrow n[x] + 1$ and $a[x] \leftarrow a[x] + z$; discard the processed record.
9: **end for**
10: Set $U_{t+1}^N(x) \leftarrow n[x]/N$ on occupied destination states.
11: Set $\widehat{m}_{t+1}^{N,\lambda}(x) \leftarrow U_{t+1}^N(x)\phi_{t+1}(x) + (\beta/N)a[x]$.
12: Commit the next occupancy, mass, and feature cache simultaneously.
13: **end for**

With state-wise feature caching, Algorithm 3 uses

$$\text{time per slice} = O(Nd + S_{t+1}^N c_\phi), \qquad \text{learner memory} = O((S_t^N + S_{t+1}^N)d). \tag{E.30}$$

The time bound includes the $O(Nd)$ transition scan and one feature evaluation per newly occupied destination state. The memory bound is learner-side temporal-credit storage and excludes physical simulator states and the feature evaluator's internal parameters. For later bounds, write

$$\|\widehat{z}_t^{N,\lambda}\|_{\infty,\mathrm{occ}} := \max_{x:U_t^N(x)>0} \|\widehat{z}_t^{N,\lambda}(x)\|_2. \tag{E.31}$$

Conventional-trace finite-window setting. The population is homogeneous and the state space and horizon are finite:

$$|\mathcal{X}| = S < \infty, \qquad T < \infty.$$

Assumption E-C1 (Consistent iid initialization). The finite population starts from independent draws of the same deterministic initial law used by the mean-field representative process.

Mathematical form of E-C1.

$$X_0^{1,N}, \dots, X_0^{N,N} \overset{\mathrm{iid}}{\sim} u_0. \tag{E.32}$$

Assumption E-C2 (Fresh idiosyncratic transitions without common noise). Conditioned on the current finite-population information, different agents draw their next states independently from the current state-population kernel, with no shared transition shock.

Mathematical form of E-C2.

$$\mathcal{L}\Big((X_{t+1}^{i,N})_{i=1}^{N} \mid \mathcal{F}_t^N\Big) = \bigotimes_{i=1}^{N} P_t(\cdot \mid X_t^{i,N}, U_t^N), \qquad 0 \le t < T. \tag{E.33}$$

Assumption E-C3 (Population regularity of the transition kernel). The closed-loop transition kernel is uniformly Lipschitz in the population distribution on the finite time window.

Mathematical form of E-C3.

$$\|P_t(\cdot \mid x,u) - P_t(\cdot \mid x,v)\|_{\mathrm{TV}} \le L_P \|u - v\|_1, \qquad {\scriptstyle 0\le t<T,\ x\in\mathcal{X},\ u,v\in\mathcal{P}(\mathcal{X})}. \tag{E.34}$$

Trace setting E-C4 (Shared time-state features and fixed trace parameter). The private, compressed, and mean-field conventional traces use the same time-state feature map and the same fixed trace parameter. The finite state-time domain gives

$$B_\phi := \max_{0\le t\le T} \max_{x\in\mathcal{X}} \|\phi_t(x)\|_2 < \infty.$$

Mathematical form of E-C4.

$$0 \le \gamma \le 1, \qquad 0 \le \lambda \le 1, \qquad \beta = \gamma\lambda. \tag{E.35}$$

Theorem 6 (Fixed-Window Conventional-Trace Compression Consistency)**.** *In the conventional-trace finite-window setting above, under Assumptions E-C1, E-C2, and E-C3, together with trace setting E-C4, both finite-population conventional credit constructions approach the same deterministic mean-field credit mass at the inverse-square-root population rate, and therefore their mutual compression gap has the same fixed-window rate.*

$$\begin{aligned}
\max_{0\le t\le T} \mathbb{E}\|\widehat{m}_t^{N,\lambda} - m_t^{\lambda}\|_{1,2} &\le \frac{C^{\lambda}_{\mathrm{comp},T}}{\sqrt{N}},\\
\max_{0\le t\le T} \mathbb{E}\|m_t^{\mathrm{priv},N,\lambda} - m_t^{\lambda}\|_{1,2} &\le \frac{C^{\lambda}_{\mathrm{priv},T}}{\sqrt{N}},\\
\max_{0\le t\le T} \mathbb{E}\|\widehat{m}_t^{N,\lambda} - m_t^{\mathrm{priv},N,\lambda}\|_{1,2} &\le \frac{C^{\lambda}_{\mathrm{bridge},T}}{\sqrt{N}},
\end{aligned} \tag{E.36}$$

Proof. The conventional trace has a deterministic finite-window envelope. The compressed-to-mean-field comparison uses a denominator-free transport inequality driven by the physical discrepancies controlled in Lemma 3. The private-to-mean-field comparison couples each private trace with the conventional trace of the corresponding nonlinear copy and then controls the iid copy empirical mass. The compressed/private bound follows through the common deterministic mean-field target.

The conventional-trace finite-window setting together with E-C1–E-C3 matches the initialization, fresh-noise, and population-regularity conditions used in Lemma 3; E-C4 uses the same shared state-only feature representation, whose bound is automatic on the finite state-time domain. Thus the physical-flow rates in Eq. (E.2) are available below.

Step 1: uniform bounds. By the finite-domain bound stated in E-C4,

$$\max_{i,N} \max_{0\le t\le T} \|\ell_t^{i,N,\lambda}\|_2 \le B_{\lambda,T}, \qquad \max_{0\le t\le T} \|z_t^{\lambda,0}\|_\infty \le B_{\lambda,T}. \tag{E.37}$$

The normalized empirical recursion obtained from (E.27) has convex predecessor weights, so

$$\max_{0\le t\le T} \|\widehat{z}_t^{N,\lambda}\|_{\infty,\mathrm{occ}} \le B_{\lambda,T} \tag{E.38}$$

pathwise.

Step 2: compressed credit. Define zero extensions and transports

$$\mathcal{T}_t^N(x) := \sum_y \Gamma_t^N(y,x)\widehat{z}_{t-1}^{N,\lambda,0}(y), \qquad \mathcal{T}_t(x) := \sum_y \Gamma_t(y,x) z_{t-1}^{\lambda,0}(y). \tag{E.39}$$

With

$$\Delta_t^N := \|\widehat{m}_t^{N,\lambda} - m_t^\lambda\|_{1,2}, \tag{E.40}$$

one has

$$\|\mathcal{T}_t^N - \mathcal{T}_t\|_{1,2} \le \Delta_{t-1}^N + B_{\lambda,T}\|U_{t-1}^N - u_{t-1}\|_1 + B_{\lambda,T}\|\Gamma_t^N - \Gamma_t\|_1. \tag{E.41}$$

Hence,

$$\begin{aligned}\Delta_t^N \le \beta\Delta_{t-1}^N + B_\phi\|U_t^N - u_t\|_1 \\ + \beta B_{\lambda,T}\big(\|U_{t-1}^N - u_{t-1}\|_1 + \|\Gamma_t^N - \Gamma_t\|_1\big).\end{aligned} \tag{E.42}$$

At $t = 0$,

$$\Delta_0^N \le B_\phi\|U_0^N - u_0\|_1. \tag{E.43}$$

By Lemma 3, fixed-window iteration gives

$$\max_{0\le t\le T} \mathbb{E}\Delta_t^N \le \frac{C^\lambda_{\mathrm{comp},T}}{\sqrt{N}}. \tag{E.44}$$

Step 3: private credit. Let $\bar{\ell}_t^{i,\lambda}$ be the conventional trace driven by the nonlinear copy $\bar{X}_t^i$ from Lemma 3 and define

$$\bar{\ell}_{-1}^{i,\lambda} = 0, \qquad \bar{\ell}_t^{i,\lambda} = \phi_t(\bar{X}_t^i) + \beta\bar{\ell}_{t-1}^{i,\lambda}. \tag{E.45}$$

Set

$$b_t^{N,\lambda} := \frac{1}{N}\sum_i \mathbb{E}\|\ell_t^{i,N,\lambda} - \bar{\ell}_t^{i,\lambda}\|_2. \tag{E.46}$$

Then

$$b_t^{N,\lambda} \le \beta b_{t-1}^{N,\lambda} + 2B_\phi a_t^N, \qquad b_0^{N,\lambda} = 0. \tag{E.47}$$

Lemma 3 implies

$$\max_{0\le t\le T} b_t^{N,\lambda} \le \frac{2B_\phi T C_{a,T}}{\sqrt{N}}. \tag{E.48}$$

Define

$$\bar{m}_t^{N,\lambda}(x) := \frac{1}{N}\sum_i \bar{\ell}_t^{i,\lambda}\mathbf{1}_{\{\bar{X}_t^i = x\}}. \tag{E.49}$$

For one agent,

$$\begin{aligned}\sum_{x\in\mathcal{X}} \Big\|\ell_t^{i,N,\lambda}\mathbf{1}_{\{X_t^{i,N}=x\}} - \bar{\ell}_t^{i,\lambda}\mathbf{1}_{\{\bar{X}_t^i=x\}}\Big\|_2 \\ \le \|\ell_t^{i,N,\lambda} - \bar{\ell}_t^{i,\lambda}\|_2 + 2B_{\lambda,T}I_t^{i,N}.\end{aligned} \tag{E.50}$$

Hence

$$\mathbb{E}\|m_t^{\mathrm{priv},N,\lambda} - \bar{m}_t^{N,\lambda}\|_{1,2} \le b_t^{N,\lambda} + 2B_{\lambda,T}a_t^N. \tag{E.51}$$

For each x, set

$$Y_{t,i}^x := \bar{\ell}_t^{i,\lambda}\mathbf{1}_{\{\bar{X}_t^i = x\}}. \tag{E.52}$$

Lemma 3 makes $(Y_{t,i}^x)_{i=1}^N$ iid with mean $m_t^\lambda(x)$ and

$$\mathbb{E}\|Y_{t,i}^x\|_2^2 \le B_{\lambda,T}^2 u_t(x). \tag{E.53}$$

Therefore

$$\begin{aligned}\mathbb{E}\|\bar{m}_t^{N,\lambda} - m_t^\lambda\|_{1,2} &\le \frac{B_{\lambda,T}}{\sqrt{N}}\sum_x \sqrt{u_t(x)} \\ &\le B_{\lambda,T}\sqrt{\frac{S}{N}}.\end{aligned} \tag{E.54}$$

Thus

$$\max_{0\le t\le T} \mathbb{E}\|m_t^{\mathrm{priv},N,\lambda} - m_t^\lambda\|_{1,2} \le \frac{C^\lambda_{\mathrm{priv},T}}{\sqrt{N}}. \tag{E.55}$$

Step 4: compressed/private bridge.

$$\|\widehat{m}_t^{N,\lambda} - m_t^{\mathrm{priv},N,\lambda}\|_{1,2} \le \|\widehat{m}_t^{N,\lambda} - m_t^{\lambda}\|_{1,2} + \|m_t^{\mathrm{priv},N,\lambda} - m_t^{\lambda}\|_{1,2}. \tag{E.56}$$

Taking expectations and maximizing over the fixed window gives the third bound in Eq. (E.36) with

$$C^{\lambda}_{\mathrm{bridge},T} := C^{\lambda}_{\mathrm{comp},T} + C^{\lambda}_{\mathrm{priv},T}. \tag{E.57}$$

Conventional-trace compression inherits the same physical $N^{-1/2}$ fluctuations as the Dutch construction but avoids the Dutch operator in the credit transport. Both finite-population credit masses are therefore controlled through the same deterministic mean-field conventional credit target. □

Direct finite-population bridge under weaker conditions. The two mean-field rates in Theorem 6 require E-C1 and E-C3. The compressed/private bridge itself does not. The next lemma isolates the weaker finite-population statement used by ridge-LSTD(λ) and ridge MC-LS.

Lemma 8 (Direct Conventional-Trace Credit Bridge)**.** *In the conventional-trace finite-window setting, under Assumption E-C2 and trace setting E-C4, the compressed and state-aggregated private conventional credit masses satisfy an inverse-square-root population gap over every fixed horizon without requiring iid initialization or population-Lipschitz transition regularity.*

$$\max_{0\le t\le T} \mathbb{E}\|\widehat{m}_t^{N,\lambda} - m_t^{\mathrm{priv},N,\lambda}\|_{1,2} \le \frac{\widetilde{C}^{\lambda}_{\mathrm{bridge},T}}{\sqrt{N}}, \qquad \widetilde{C}^{\lambda}_{\mathrm{bridge},T} < \infty. \tag{E.58}$$

Proof. Subtract the two finite-population conventional-trace recursions. The old-credit transport is nonexpansive in the unnormalized mass norm, while the remaining within-state residual is centered before the fresh independent destination draws. This gives a direct finite-window $N^{-1/2}$ recursion without any mean-field comparison.

Let

$$D_t^{N,\lambda} := \widehat{m}_t^{N,\lambda} - m_t^{\mathrm{priv},N,\lambda}. \tag{E.59}$$

Because the two constructions have the same empirical initialization,

$$D_0^{N,\lambda} = 0 \tag{E.60}$$

pathwise for an arbitrary realized initial population. For $0 \le t < T$, define

$$\rho_{t,i}^{\lambda} := z_t^{\mathrm{priv},N,\lambda,0}(X_t^{i,N}) - \ell_t^{i,N,\lambda}. \tag{E.61}$$

By definition of the within-state private average,

$$\sum_{i:X_t^{i,N}=y} \rho_{t,i}^{\lambda} = 0, \qquad y \in \mathcal{X}. \tag{E.62}$$

Subtracting the state-aggregated private recursion from Eq. (E.27) gives, for every destination x',

$$D_{t+1}^{N,\lambda}(x') = \beta\left[\sum_y \Gamma_{t+1}^N(y,x')\big(\widehat{z}_t^{N,\lambda,0}(y) - z_t^{\mathrm{priv},N,\lambda,0}(y)\big) + R_{t+1}^{N,\lambda}(x')\right], \tag{E.63}$$

where

$$R_{t+1}^{N,\lambda}(x') := \frac{1}{N}\sum_{i=1}^{N} \rho_{t,i}^{\lambda}\mathbf{1}_{\{X_{t+1}^{i,N}=x'\}}. \tag{E.64}$$

The predecessor marginal of Γ_{t+1}^N gives the pathwise transport bound

$$\sum_{x'}\left\|\sum_y \Gamma_{t+1}^N(y,x')\big(\widehat{z}_t^{N,\lambda,0}(y) - z_t^{\mathrm{priv},N,\lambda,0}(y)\big)\right\|_2 \le \|D_t^{N,\lambda}\|_{1,2}. \tag{E.65}$$

Moreover, the conventional-trace envelope implies

$$\|\rho_{t,i}^{\lambda}\|_2 \le 2B_{\lambda,T}. \tag{E.66}$$

Conditioned on $\mathcal{F}_t^N$, Eq. (E.62) and E-C2 imply

$$\mathbb{E}[R_{t+1}^{N,\lambda}(x') \mid \mathcal{F}_t^N] = 0. \tag{E.67}$$

$$p_{t,i}^N(x') := P_t\Big(x' \mid X_t^{i,N}, U_t^N\Big), \qquad \sum_{i=1}^N \rho_{t,i}^\lambda p_{t,i}^N(x') = 0. \tag{E.68}$$

$$R_{t+1}^{N,\lambda}(x') = \frac{1}{N}\sum_{i=1}^N \rho_{t,i}^\lambda \left(\mathbf{1}_{\{X_{t+1}^{i,N}=x'\}} - p_{t,i}^N(x')\right). \tag{E.69}$$

Using this zero total conditional mean together with conditional independence of the fresh destination draws,

$$\mathbb{E}\Big[\|R_{t+1}^{N,\lambda}(x')\|_2^2 \mid \mathcal{F}_t^N\Big] \le \frac{4B_{\lambda,T}^2}{N}. \tag{E.70}$$

Hence

$$\mathbb{E}\|R_{t+1}^{N,\lambda}\|_{1,2} \le \frac{2SB_{\lambda,T}}{\sqrt{N}}. \tag{E.71}$$

Combining the last three displays gives

$$\mathbb{E}\|D_{t+1}^{N,\lambda}\|_{1,2} \le \beta\mathbb{E}\|D_t^{N,\lambda}\|_{1,2} + \frac{2\beta SB_{\lambda,T}}{\sqrt{N}}. \tag{E.72}$$

Iteration from $D_0^{N,\lambda} = 0$ proves Eq. (E.58), for example with

$$\widetilde{C}_{\mathrm{bridge},T}^\lambda := 2SB_{\lambda,T}\sum_{j=1}^T \beta^j < \infty. \tag{E.73}$$

□

Remark 12. *For a supported state-population pair, the finite-population oracle field*

$$z_t^{\mathrm{or},N,\lambda}(x,u) := \mathbb{E}[\ell_t^{i,N,\lambda} \mid X_t^{i,N} = x, U_t^N = u] \tag{E.74}$$

has the same fixed-parameter conditional expected-update identity as an ordinary expected eligibility trace. This oracle object is not the empirical compressed readout $\widehat{z}_t^{N,\lambda}$; Theorem 6 compares the corresponding state-aggregated finite-population credit masses instead of asserting their pathwise equality.

E.3 DoubleSWEET Ridge-LSTD(λ) Algorithm

Explanation. Ridge-LSTD(λ) does not update a critic parameter after every transition. Instead, it accumulates a matrix and a right-hand vector through the episode and solves one regularized linear system at the end. The conventional-trace compression of Sec. E.2 can therefore be lifted by replacing each private trace in these sufficient statistics with the state-wise compressed readout. This subsection defines the resulting sufficient statistics and their one-pass implementation; Sec. E.4 establishes consistency and ridge stability.

Formal construction. For $0 \le t < T$, write

$$\phi_t^{i,N} := \phi_t(X_t^{i,N}), \qquad d_{t,i}^N := \phi_t^{i,N} - \gamma\phi_{t+1}^{i,N}. \tag{E.75}$$

Define the private sufficient statistics by

$$H_\lambda^{N,\mathrm{priv}} := \frac{1}{N}\sum_{t=0}^{T-1}\sum_{i=1}^N \ell_t^{i,N,\lambda}(d_{t,i}^N)^\top, \tag{E.76}$$

$$g_\lambda^{N,\mathrm{priv}} := \frac{1}{N}\sum_{t=0}^{T-1}\sum_{i=1}^N \ell_t^{i,N,\lambda} R_{t+1}^{i,N}, \tag{E.77}$$

The compressed statistics are

$$H_\lambda^{N,C} := \frac{1}{N}\sum_{t=0}^{T-1}\sum_{i=1}^{N}\widehat{z}_t^{N,\lambda}(X_t^{i,N})(d_{t,i}^N)^\top, \tag{E.78}$$

$$g_\lambda^{N,C} := \frac{1}{N}\sum_{t=0}^{T-1}\sum_{i=1}^{N}\widehat{z}_t^{N,\lambda}(X_t^{i,N})R_{t+1}^{i,N}, \tag{E.79}$$

The matrix H_λ is generally nonsymmetric. The construction keeps the LSTD system in this form; it does not symmetrize H_λ or replace the system by a residual least-squares normal equation.

Terminal convention E-L1 (Genuine episodic terminal time). For the least-squares extensions, the final time is a true terminal time and the terminal feature is zero.

Mathematical form of E-L1.

$$T \geq 1, \qquad \phi_T(x) = 0 \quad (x \in \mathcal{X}). \tag{E.80}$$

Least-squares reward setting. The private and compressed sufficient statistics use the same deterministic current-state/population reward from the standing model,

$$R_{t+1}^{i,N} = r_t(X_t^{i,N}, U_t^N). \tag{E.81}$$

Formal construction of the one-pass statistics. For the transition $t \to t+1$, define

$$\mathrm{Acc}_{t+1}^{N,\lambda}(x') := \sum_{i=1}^{N}\mathbf{1}_{\{X_{t+1}^{i,N}=x'\}}\widehat{z}_t^{N,\lambda}(X_t^{i,N}). \tag{E.82}$$

Then, pathwise,

$$\frac{1}{N}\sum_i \widehat{z}_t^{N,\lambda}(X_t^{i,N})(\phi_t^{i,N})^\top = \sum_x \widehat{m}_t^{N,\lambda}(x)\phi_t(x)^\top, \tag{E.83}$$

$$\frac{1}{N}\sum_i \widehat{z}_t^{N,\lambda}(X_t^{i,N})(\phi_{t+1}^{i,N})^\top = \frac{1}{N}\sum_{x'}\mathrm{Acc}_{t+1}^{N,\lambda}(x')\phi_{t+1}(x')^\top. \tag{E.84}$$

Therefore

$$\Delta H_{\lambda,t}^{N,C} = \sum_x \widehat{m}_t^{N,\lambda}(x)\phi_t(x)^\top - \frac{\gamma}{N}\sum_{x'}\mathrm{Acc}_{t+1}^{N,\lambda}(x')\phi_{t+1}(x')^\top, \tag{E.85}$$

$$\Delta g_{\lambda,t}^{N,C} = \sum_x \widehat{m}_t^{N,\lambda}(x)r_t(x, U_t^N), \tag{E.86}$$

$$\widehat{m}_{t+1}^{N,\lambda}(x') = U_{t+1}^N(x')\phi_{t+1}(x') + \frac{\beta}{N}\mathrm{Acc}_{t+1}^{N,\lambda}(x'). \tag{E.87}$$

Ridge setting E-L3 (Fixed positive ridge coefficient). The final least-squares estimator uses a fixed positive diagonal ridge shift. The ridge coefficient does not depend on the realized episode or the population size.

Mathematical form of E-L3.

$$\eta > 0. \tag{E.88}$$

Ridge systems and complete one-pass realization. Under ridge setting E-L3, the private and compressed ridge-LSTD outputs are specified by the linear systems

$$(H_\lambda^{N,\mathrm{priv}} + \eta I_d)w_{\lambda,\eta}^{N,\mathrm{priv}} = g_\lambda^{N,\mathrm{priv}}, \tag{E.89}$$

and

$$(H_\lambda^{N,C} + \eta I_d)w_{\lambda,\eta}^{N,C} = g_\lambda^{N,C}. \tag{E.90}$$

The diagonal-shifted LSTD system $(H_\lambda + \eta I_d)w = g_\lambda$ is used exactly as written. Section E.4 proves that, under terminal convention E-L1, ridge setting E-L3, and the standing trace parameters in Sec. E.1, both systems are uniquely solvable and therefore admit the usual inverse representations.

Algorithm 4 DoubleSWEET ridge-LSTD(λ) one-pass estimator

Compute U_0^N, cache $\phi_0(x)$ on the occupied support, set $\widehat{m}_0^{N,\lambda}(x) = U_0^N(x)\phi_0(x)$, and initialize $H = 0_{d\times d}$ and $g = 0_d$.
for $t = 0, \dots, T-1$ **do**
  $H \leftarrow H + \sum_y \widehat{m}_t^{N,\lambda}(y)\phi_t(y)^\top$.
  Initialize sparse destination counts $n[x] = 0$, credit accumulators $a[x] = 0 \in \mathbb{R}^d$, and an empty next-feature cache.
  for each record $(y, R, x) = (X_t^{i,N}, R_{t+1}^{i,N}, X_{t+1}^{i,N})$ **do**
    $z \leftarrow \widehat{m}_t^{N,\lambda}(y)/U_t^N(y)$.
    If x is new, evaluate and cache $\phi_{t+1}(x)$ once.
    $g \leftarrow g + (1/N)zR$, $n[x] \leftarrow n[x] + 1$, and $a[x] \leftarrow a[x] + z$; discard the record.
  end for
  $H \leftarrow H - (\gamma/N)\sum_x a[x]\phi_{t+1}(x)^\top$.
  Set $U_{t+1}^N(x) \leftarrow n[x]/N$ and $\widehat{m}_{t+1}^{N,\lambda}(x) \leftarrow U_{t+1}^N(x)\phi_{t+1}(x) + (\beta/N)a[x]$.
  Commit the next occupancy, mass, and feature cache simultaneously.
end for
Output $w_{\lambda,\eta}^{N,C} = \mathrm{solve}(H + \eta I_d, g)$.

Under sparse state maps and dense outer-product arithmetic, Algorithm 4 uses

$$\text{time per slice} = O\big(Nd + (S_t^N + S_{t+1}^N)d^2 + S_{t+1}^N c_\phi\big)\,, \tag{E.91}$$

$$\text{peak learner memory} = O\big((S_t^N + S_{t+1}^N)d + d^2\big)\,, \qquad \text{episode-end dense solve} = O(d^3). \tag{E.92}$$

The memory comparison concerns learner-side trace/statistic storage. An optimized private implementation may also group outer products by state, so no stronger asymptotic runtime separation is claimed solely from the compression argument.

E.4 DoubleSWEET Ridge-LSTD(λ) Consistency Theorem 7

Explanation. We now justify the construction in Sec. E.3. First, the compressed matrix and right-hand sufficient statistics approach their private counterparts at the fixed-window inverse-square-root population rate. Second, the complete-episode private and compressed LSTD matrices are pathwise accretive, so the positive ridge shift makes both systems uniquely solvable with uniform inverse bounds. These two facts yield parameter and learned-value consistency. An exact matrix-gap identity is then recorded as the structural bridge to the $\lambda = 1$ ridge MC-LS specialization.

Reward condition for the theory. Recall the deterministic least-squares reward setting (E.81) from Sec. E.3. The consistency argument additionally uses the following boundedness assumption.

Assumption E-L2 (Uniform reward boundedness). The deterministic reward function is uniformly bounded over the state–population domain.

Mathematical form of E-L2.

$$\sup_{\substack{0\le t<T,\ x\in\mathcal{X},\\ u\in\mathcal{P}(\mathcal{X})}} |r_t(x,u)| \le B_r < \infty. \tag{E.93}$$

Formal credit-gap quantities. For $0 \le t \le T$, define

$$D_t^{N,\lambda} := \widehat{m}_t^{N,\lambda} - m_t^{\mathrm{priv},N,\lambda}, \tag{E.94}$$

and the population-weighted readout gap

$$G_t^{N,\lambda} := \sum_{x\in\mathcal{X}} U_t^N(x)\|\widehat{z}_t^{N,\lambda,0}(x) - z_t^{\mathrm{priv},N,\lambda,0}(x)\|_2 = \|D_t^{N,\lambda}\|_{1,2}. \tag{E.95}$$

Lemma 9 (LSTD Sufficient-Statistic Compression Consistency)**.** *Using the least-squares reward setting above, under Assumptions E-C2 and E-L2, together with trace setting E-C4, the compressed and private LSTD(λ) matrix and right-hand sufficient statistics differ by at most the inverse-square-root population rate on the fixed episode without requiring iid initialization or population-Lipschitz transition regularity.*

$$\mathbb{E}\|H_\lambda^{N,C} - H_\lambda^{N,\mathrm{priv}}\|_{\mathrm{op}} \le \frac{C_{H,T}^\lambda}{\sqrt{N}}, \qquad \mathbb{E}\|g_\lambda^{N,C} - g_\lambda^{N,\mathrm{priv}}\|_2 \le \frac{C_{g,T}^\lambda}{\sqrt{N}}. \tag{E.96}$$

Proof. The reward statistic groups exactly by current state, so its gap is a bounded reward-weighted sum of the direct finite-population credit-mass gaps from Lemma 8. For the matrix statistic, the current-feature residual also cancels statewise. The next-feature term is split into the state-wise credit gap and the within-state private residual; the conditional mean of the latter cancels exactly within each current state, while the centered fresh-transition fluctuation contributes only $O(N^{-1/2})$ by Assumption E-C2.

Step 1: reward statistic. By E-L2,

$$g_\lambda^{N,C} - g_\lambda^{N,\mathrm{priv}} = \sum_{t=0}^{T-1} \sum_x D_t^{N,\lambda}(x) r_t(x, U_t^N), \tag{E.97}$$

$$\|g_\lambda^{N,C} - g_\lambda^{N,\mathrm{priv}}\|_2 \le B_r \sum_{t=0}^{T-1} \|D_t^{N,\lambda}\|_{1,2}. \tag{E.98}$$

Thus Lemma 8 gives the stated g bound.

Step 2: current-feature matrix term. For every $t < T$,

$$\frac{1}{N}\sum_i [\widehat{z}_t^{N,\lambda}(X_t^{i,N}) - \ell_t^{i,N,\lambda}](\phi_t^{i,N})^\top = \sum_x D_t^{N,\lambda}(x)\phi_t(x)^\top, \tag{E.99}$$

$$\left\| \sum_x D_t^{N,\lambda}(x)\phi_t(x)^\top \right\|_{\mathrm{op}} \le B_\phi \|D_t^{N,\lambda}\|_{1,2}. \tag{E.100}$$

Step 3: next-feature matrix term. Let

$$\rho_{t,i} := z_t^{\mathrm{priv},N,\lambda,0}(X_t^{i,N}) - \ell_t^{i,N,\lambda}, \tag{E.101}$$

and

$$\mu_t^N(x) := \sum_{x'} P_t(x' \mid x, U_t^N)\phi_{t+1}(x'), \qquad \xi_{t,i} := \phi_{t+1}(X_{t+1}^{i,N}) - \mu_t^N(X_t^{i,N}). \tag{E.102}$$

Then

$$\sum_{i: X_t^{i,N} = x} \rho_{t,i} = 0, \tag{E.103}$$

and, conditionally on $\mathcal{F}_t^N$, the $\xi_{t,i}$ are centered and independent. Therefore

$$\begin{aligned} &\frac{1}{N}\sum_i [\widehat{z}_t^{N,\lambda}(X_t^{i,N}) - \ell_t^{i,N,\lambda}](\phi_{t+1}^{i,N})^\top \\ &\quad = \frac{1}{N}\sum_i [\widehat{z}_t^{N,\lambda} - z_t^{\mathrm{priv},N,\lambda,0}](X_t^{i,N})(\phi_{t+1}^{i,N})^\top + \frac{1}{N}\sum_i \rho_{t,i}\xi_{t,i}^\top. \end{aligned} \tag{E.104}$$

The first term has operator norm at most

$$B_\phi \sum_x U_t^N(x) \|\widehat{z}_t^{N,\lambda,0}(x) - z_t^{\mathrm{priv},N,\lambda,0}(x)\|_2 = B_\phi \|D_t^{N,\lambda}\|_{1,2}. \tag{E.105}$$

Moreover,

$$\mathbb{E}\left[\left\| \frac{1}{N}\sum_i \rho_{t,i}\xi_{t,i}^\top \right\|_F^2 \middle| \mathcal{F}_t^N \right] \le \frac{16 B_{\lambda,T}^2 B_\phi^2}{N}. \tag{E.106}$$

Thus

$$\mathbb{E}\|H_\lambda^{N,C} - H_\lambda^{N,\mathrm{priv}}\|_{\mathrm{op}} \le \frac{C_{H,T}^\lambda}{\sqrt{N}}, \tag{E.107}$$

using Lemma 8 and summing over $t < T$.

The state-wise credit bridge controls every deterministic current-state contribution. Only the next-state feature introduces a residual agent-wise fluctuation, and conditional centering plus fresh idiosyncratic transition noise reduces that term to the standard inverse-square-root population scale. □

Lemma 10 (Pathwise Accretivity and Ridge Stability)**.** *Under terminal convention E-L1 and the standing trace parameters in Sec. E.1, both complete-episode LSTD matrices are pathwise accretive; under ridge setting E-L3, their ridge-shifted systems are uniformly invertible.*

$$\begin{aligned} v^\top H_\lambda^{N,\mathrm{priv}} v \ge 0, \qquad v^\top H_\lambda^{N,C} v \ge 0, \qquad \forall v \in \mathbb{R}^d, \\ \|(H_\lambda^{N,\mathrm{priv}} + \eta I_d)^{-1}\|_{\mathrm{op}} \le \eta^{-1}, \qquad \|(H_\lambda^{N,C} + \eta I_d)^{-1}\|_{\mathrm{op}} \le \eta^{-1}. \end{aligned} \tag{E.108}$$

Proof. For a private trajectory, projecting the feature and trace onto an arbitrary direction v converts the episodic LSTD quadratic form into an explicit sum of nonnegative squares and boundary terms. For the compressed matrix, the empirical occupancy and pair-flow marginals define a surrogate time-inhomogeneous Markov chain whose conditional conventional trace equals the compressed readout, so the private pathwise nonnegativity transfers by expectation. A positive ridge shift then lower-bounds the smallest singular value.

Step 1: private matrix. Fix $v \in \mathbb{R}^d$ and one private trajectory. Set

$$y_t := v^\top \phi_t(X_t^{i,N}), \qquad q_t := v^\top \ell_t^{i,N,\lambda}, \qquad q_{-1} := 0. \tag{E.109}$$

Then $q_t = y_t + \beta q_{t-1}$ and $y_T = 0$. Let $a := (\beta + \gamma)/2$. For $T \ge 2$,

$$\begin{aligned} \sum_{t=0}^{T-1} q_t(y_t - \gamma y_{t+1}) &= a \sum_{t=0}^{T-2} (q_t - q_{t+1})^2 + (1 + \beta\gamma - a) q_0^2 \\ &\quad + (1-\beta)(1-\gamma) \sum_{t=1}^{T-2} q_t^2 + (1-a) q_{T-1}^2 \ge 0. \end{aligned} \tag{E.110}$$

By (E.3), all coefficients in (E.110) are nonnegative. For $T = 1$, the contribution is $q_0^2 \ge 0$. Averaging over agents gives

$$v^\top H_\lambda^{N,\mathrm{priv}} v \ge 0. \tag{E.111}$$

Step 2: compressed matrix. On every occupied predecessor state define

$$\widehat{K}_t^N(x' \mid x) := \frac{\Gamma_{t+1}^N(x, x')}{U_t^N(x)}. \tag{E.112}$$

Extend $\widehat{K}_t^N(\cdot \mid x)$ arbitrarily to a probability vector on empty predecessor rows, which are never visited under the corresponding empirical marginal. Let $Y_0 \sim U_0^N$ and $Y_{t+1} \sim \widehat{K}_t^N(\cdot \mid Y_t)$, and define

$$\widetilde{\ell}_{-1} = 0, \qquad \widetilde{\ell}_t = \phi_t(Y_t) + \beta \widetilde{\ell}_{t-1}. \tag{E.113}$$

First, the empirical flow marginals give $\mathcal{L}(Y_t) = U_t^N$ for every t by induction. We next prove the conditional-trace identity. At $t = 0$,

$$\mathbb{E}_{\widehat{K}^N}[\widetilde{\ell}_0 \mid Y_0 = x] = \phi_0(x) = \widehat{z}_0^{N,\lambda}(x).$$

For $t \ge 1$ and every occupied x, using the reverse conditional law induced by the empirical pair flow and the induction hypothesis,

$$\begin{aligned} \mathbb{E}_{\widehat{K}^N}[\widetilde{\ell}_t \mid Y_t = x] &= \phi_t(x) + \beta \sum_y \frac{\Gamma_t^N(y,x)}{U_t^N(x)} \mathbb{E}_{\widehat{K}^N}[\widetilde{\ell}_{t-1} \mid Y_{t-1} = y] \\ &= \phi_t(x) + \beta \sum_y \frac{\Gamma_t^N(y,x)}{U_t^N(x)} \widehat{z}_{t-1}^{N,\lambda}(y) = \widehat{z}_t^{N,\lambda}(x). \end{aligned} \tag{E.114}$$

Thus the conditional trace of the surrogate chain equals the compressed readout on occupied states. Moreover, by the Markov construction, $\widetilde{\ell}_t \perp Y_{t+1} \mid Y_t$. Hence

$$H_\lambda^{N,C} = \mathbb{E}_{\widehat{K}^N}\left[\sum_{t=0}^{T-1} \widetilde{\ell}_t(\phi_t(Y_t) - \gamma\phi_{t+1}(Y_{t+1}))^\top\right], \tag{E.115}$$

and (E.110) yields

$$v^\top H_\lambda^{N,C} v \geq 0. \tag{E.116}$$

Step 3: ridge inverse. For either matrix H above,

$$v^\top(H + \eta I_d)v \geq \eta\|v\|_2^2. \tag{E.117}$$

Thus

$$\|(H + \eta I_d)v\|_2 \geq \eta\|v\|_2, \tag{E.118}$$

which implies the inverse bounds in (E.108).

Complete-episode terminal structure makes both LSTD matrices accretive even though they are generally nonsymmetric. The ridge shift therefore gives a direct singular-value bound, avoiding any symmetry assumption. □

Theorem 7 (DoubleSWEET Ridge-LSTD(λ) Parameter and Value Consistency)**.** *Using the least-squares reward setting above, under Assumptions E-C2 and E-L2, together with trace setting E-C4, terminal convention E-L1, and ridge setting E-L3, the DoubleSWEET ridge-LSTD(λ) parameter and its learned value readout approach their private counterparts at the inverse-square-root population rate for every fixed ridge coefficient.*

$$\begin{aligned} \mathbb{E}\|w_{\lambda,\eta}^{N,C} - w_{\lambda,\eta}^{N,\mathrm{priv}}\|_2 &\leq \frac{C_{w,T}^{\lambda,\eta}}{\sqrt{N}}, \\ \max_{0\leq t\leq T} \mathbb{E}\max_{x\in\mathcal{X}}\left|(w_{\lambda,\eta}^{N,C} - w_{\lambda,\eta}^{N,\mathrm{priv}})^\top\phi_t(x)\right| &\leq \frac{B_\phi C_{w,T}^{\lambda,\eta}}{\sqrt{N}}. \end{aligned} \tag{E.119}$$

Proof. Use the exact inverse perturbation identity to separate the right-hand statistic error from the matrix error. The ridge inverse bounds from Lemma 10 control both inverse factors, and Lemma 9 supplies the two sufficient-statistic rates.

Let $H_C := H_\lambda^{N,C}$, $H_P := H_\lambda^{N,\mathrm{priv}}$ and define g_C, g_P analogously. From (E.37) and E-L2,

$$\|g_P\|_2 \leq B_{g,T} := TB_rB_{\lambda,T}. \tag{E.120}$$

The inverse perturbation identity gives

$$\begin{aligned} w_C - w_P = (H_C + \eta I_d)^{-1}(g_C - g_P) \\ + (H_C + \eta I_d)^{-1}(H_P - H_C)(H_P + \eta I_d)^{-1}g_P. \end{aligned} \tag{E.121}$$

Hence, by Lemma 10,

$$\|w_C - w_P\|_2 \leq \frac{1}{\eta}\|g_C - g_P\|_2 + \frac{B_{g,T}}{\eta^2}\|H_C - H_P\|_{\mathrm{op}}. \tag{E.122}$$

Taking expectations and applying Lemma 9 proves the parameter bound in (E.119). The shared feature bound then gives

$$\max_{0\leq t\leq T} \mathbb{E}\max_{x\in\mathcal{X}}\left|(w_{\lambda,\eta}^{N,C} - w_{\lambda,\eta}^{N,\mathrm{priv}})^\top\phi_t(x)\right| \leq B_\phi\,\mathbb{E}\|w_{\lambda,\eta}^{N,C} - w_{\lambda,\eta}^{N,\mathrm{priv}}\|_2, \tag{E.123}$$

which proves the learned-value bound in (E.119). □

Exact matrix-gap quantity. For $0 \le t \le T$, define

$$F_t^{N,\lambda} := \sum_x D_t^{N,\lambda}(x)\phi_t(x)^\top. \tag{E.124}$$

Lemma 11 (Exact LSTD Matrix-Gap Identity)**.** *Under terminal convention E-L1 and trace setting E-C4, the compressed and private LSTD matrices satisfy an exact pathwise gap identity; in particular, at $\lambda = 1$ the two matrices coincide pathwise.*

$$\lambda(H_\lambda^{N,C} - H_\lambda^{N,\mathrm{priv}}) = (\lambda - 1)\sum_{t=1}^{T-1} F_t^{N,\lambda}. \tag{E.125}$$

Proof. The matrix gap can be written in terms of the current credit-mass gap and the next-feature credit gap. The conventional trace recursion converts the latter into the next credit-mass gap, causing an exact telescoping across time. The common initialization and genuine terminal feature remove the two boundary terms.

Subtracting the private and compressed conventional mass recursions yields

$$F_{t+1}^{N,\lambda} = \frac{\beta}{N}\sum_{i=1}^{N}[\widehat{z}_t^{N,\lambda}(X_t^{i,N}) - \ell_t^{i,N,\lambda}](\phi_{t+1}^{i,N})^\top. \tag{E.126}$$

The common initialization gives $F_0^{N,\lambda} = 0$, and terminal convention E-L1 gives $F_T^{N,\lambda} = 0$. Using $\beta = \gamma\lambda$ and the definitions of the two LSTD matrices,

$$\begin{aligned}\lambda(H_\lambda^{N,C} - H_\lambda^{N,\mathrm{priv}}) &= \sum_{t=0}^{T-1}(\lambda F_t^{N,\lambda} - F_{t+1}^{N,\lambda}) \\ &= (\lambda - 1)\sum_{t=1}^{T-1} F_t^{N,\lambda},\end{aligned} \tag{E.127}$$

which is (E.125). □

At $\lambda = 0$, the private and compressed conventional traces both equal the current feature on occupied states, so both sufficient statistics agree pathwise. At $\lambda = 1$, Lemma 11 gives

$$H_1^{N,C} = H_1^{N,\mathrm{priv}} \quad \text{pathwise.} \tag{E.128}$$

The latter identity does not imply pathwise equality of the two right-hand statistics. This exact endpoint identity is the structural bridge to the ridge MC-LS specialization in Sec. E.5.

E.5 DoubleSWEET Ridge MC-LS Consistency Theorem 8

Explanation. At $\lambda = 1$, the conventional trace contains all discounted past features. Under terminal convention E-L1, the private LSTD matrix telescopes to the ordinary feature Gram matrix, while the right-hand statistic becomes the feature-weighted full episodic return. Thus the private estimator is exactly ridge MC-LS. The compressed matrix is even more tightly aligned: Lemma 11 makes the compressed and private Gram matrices pathwise identical, so only the right-hand statistic carries a finite-population compression error.

Formal construction. For each agent, define the full episodic return

$$G_{t,i} := \sum_{s=t}^{T-1}\gamma^{s-t}R_{s+1}^{i,N}, \qquad 0 \le t < T. \tag{E.129}$$

At $\lambda = 1$,

$$\ell_t^{i,N,1} = \sum_{s=0}^{t}\gamma^{t-s}\phi_s^{i,N}. \tag{E.130}$$

Terminal convention E-L1 yields the exact private identities

$$H_1^{N,\mathrm{priv}} = \frac{1}{N}\sum_{i=1}^{N}\sum_{t=0}^{T-1}\phi_t^{i,N}(\phi_t^{i,N})^\top, \tag{E.131}$$

$$g_1^{N,\mathrm{priv}} = \frac{1}{N}\sum_{i=1}^{N}\sum_{t=0}^{T-1}\phi_t^{i,N}G_{t,i}. \tag{E.132}$$

Therefore $w_{1,\eta}^{N,\mathrm{priv}}$ is ridge MC-LS under the normalization used in Appendix E.

MC-LS specialization. Set

$$\lambda = 1, \qquad \beta = \gamma, \tag{E.133}$$

so the conventional eligibility trace contains the full discounted feature history.

Specialization of Algorithm 4. Set $\lambda = 1$ in the conventional credit transport. The matrix update no longer needs the LSTD next-feature outer-product term because (E.128) gives the exact Gram identity. Thus, at each time t, accumulate

$$H \leftarrow H + \sum_x U_t^N(x)\phi_t(x)\phi_t(x)^\top, \tag{E.134}$$

scan the transition batch once to update the compressed right-hand statistic and the next conventional credit mass, and solve $(H + \eta I_d)w = g$ once at the episode end. No second full pseudocode is required because every remaining scan and credit-transport step is identical to Algorithm 4 with $\beta = \gamma$.

The resulting per-slice complexity is

$$\text{time per slice} = O\big(Nd + S_t^N d^2 + S_{t+1}^N c_\phi\big), \tag{E.135}$$

$$\text{peak learner memory} = O\big((S_t^N + S_{t+1}^N)d + d^2\big), \qquad \text{episode-end solve} = O(d^3). \tag{E.136}$$

Theorem 8 (DoubleSWEET Ridge MC-LS Parameter and Value Consistency)**.** *At $\lambda = 1$, using the least-squares reward setting above and under Assumptions E-C2 and E-L2, together with trace setting E-C4, terminal convention E-L1, and ridge setting E-L3, the compressed and private ridge MC-LS Gram matrices are pathwise identical, while their right-hand statistics, final parameters, and learned value readouts agree up to the inverse-square-root population rate.*

$$\begin{aligned}
H_1^{N,C} &= H_1^{N,\mathrm{priv}} \quad \textit{pathwise},\\
\mathbb{E}\|g_1^{N,C} - g_1^{N,\mathrm{priv}}\|_2 &\le \frac{C_{g,T}^1}{\sqrt{N}},\\
\mathbb{E}\|w_{1,\eta}^{N,C} - w_{1,\eta}^{N,\mathrm{priv}}\|_2 &\le \frac{C_{g,T}^1}{\eta\sqrt{N}},\\
\max_{0\le t\le T}\mathbb{E}\max_{x\in\mathcal{X}}\Big|(w_{1,\eta}^{N,C} - w_{1,\eta}^{N,\mathrm{priv}})^\top\phi_t(x)\Big| &\le \frac{B_\phi C_{g,T}^1}{\eta\sqrt{N}}.
\end{aligned} \tag{E.137}$$

Proof. The exact Gram identity is the $\lambda = 1$ endpoint of Lemma 11. The right-hand statistic bound is the corresponding specialization of Lemma 9. Because the Gram matrices coincide, the ridge estimator difference contains only the right-hand statistic error and one inverse factor.

By Lemma 11 (equivalently, (E.125)) at $\lambda = 1$,

$$H_1^{N,C} = H_1^{N,\mathrm{priv}}. \tag{E.138}$$

By Lemma 9,

$$\mathbb{E}\|g_1^{N,C} - g_1^{N,\mathrm{priv}}\|_2 \le \frac{C_{g,T}^1}{\sqrt{N}}. \tag{E.139}$$

Hence, by Lemma 10,

$$\|w_{1,\eta}^{N,C} - w_{1,\eta}^{N,\mathrm{priv}}\|_2 = \|(H_1^{N,C} + \eta I_d)^{-1}(g_1^{N,C} - g_1^{N,\mathrm{priv}})\|_2 \\ \leq \eta^{-1}\|g_1^{N,C} - g_1^{N,\mathrm{priv}}\|_2. \tag{E.140}$$

Taking expectations gives the parameter bound in (E.137). The learned-value bound follows from

$$\max_{0\leq t\leq T} \mathbb{E} \max_{x\in\mathcal{X}} \left|(w_{1,\eta}^{N,C} - w_{1,\eta}^{N,\mathrm{priv}})^\top \phi_t(x)\right| \leq B_\phi \, \mathbb{E}\|w_{1,\eta}^{N,C} - w_{1,\eta}^{N,\mathrm{priv}}\|_2. \tag{E.141}$$

Thus all claims in (E.137) hold. □

Remark 13. *Appendix E compares compressed and private finite-population estimators on a fixed episode. It does not assert convergence of either ridge-LSTD or ridge MC-LS estimator to the deterministic mean-field critic or to the exact Bellman value. The $N^{-1/2}$ statements are expected-norm fixed-window bounds, and their estimator constants are not uniform as $\eta \downarrow 0$. The MC-LS telescoping identities require the genuine terminal convention $\phi_T \equiv 0$. At $\lambda = 1$, the Gram matrix is exactly preserved by compression, but the right-hand statistic is only consistent in expected norm. The stated storage bounds are learner-side and exclude physical simulator states; every population transition is still scanned once.*

F Experimental Details

F.1 Experimental Setup, Protocol, and Reproducibility Details

F.1.1 Overview of Experiments

We use experiments to answer three questions.

Question 1: Can our theoretical results be validated experimentally? Experiment 1 examines the finite-population behavior predicted by our theory. We increase the population size N and compare finite-population private constructions, their state-wise compressed counterparts, and the corresponding mean-field quantities. The numerical study covers the Dutch-trace construction as well as the conventional accumulating-trace and least-squares extensions developed in Secs. E.2–E.5. The purpose of this experiment is not to rank policy-evaluation algorithms, but to test whether the finite-population and compression discrepancies display the qualitative scaling predicted by the fixed-window theory.

Experiment 1 is conducted in two environments: a Beach-Bar-inspired ring population environment (Environment A) and a Garnet-inspired time-varying three-region environment (Environment B). Environment A provides a setting in which both the transition law and reward depend explicitly on the population distribution. Environment B provides a larger, time-varying fixed-policy setting in which the transition kernel is population-independent while the reward remains population-dependent. The two environments therefore probe complementary sources of finite-population error.

Question 2: How do different policy-evaluation approaches trade off estimation accuracy, computation, and memory? Experiment 2 studies this question at two levels. First, within three estimator families, we compare a per-agent implementation with its DoubleSWEET-compressed counterpart: Private Dutch versus DoubleSWEET, MC-LS versus DoubleSWEET-MC-LS, and LSTD(λ) versus DoubleSWEET-LSTD(λ). Second, we compare direct sample-based policy evaluation with baselines that first estimate an empirical reward/transition model and then apply a backward Bellman recursion. The model-based baselines estimate their models only from the observed forward samples; they do not receive the true transition kernel or the oracle value function.

We compare these approaches along

estimation accuracy, computational cost, memory usage.

Environment B is used for this study because S, N, T, and d can be varied one at a time while the remaining quantities are fixed. The formal grid contains one-factor-at-a-time sweeps over all four quantities.

Table 4: Roles of the three experimental environments.

Experiment	Main question	Env. A	Env. B	Env. C
Exp. 1	Do finite-population and compression discrepancies exhibit the theoretically predicted fixed-window behavior as N increases?	✓	✓	–
Exp. 2	What are the accuracy–computation–memory trade-offs of trace compression and of direct evaluation versus model-estimation-then-Bellman evaluation?	–	✓	–
Exp. 3	Can a critic learned from forward population evolution support policy improvement and finite-population MFG diagnostics?	–	–	✓

Question 3: Can a critic learned during forward population propagation support MFG solving? Experiment 3 places the critic inside an outer policy-improvement loop in a synthetic two-stage congestion-routing game (Environment C). Agents choose between two routes at the origin and again at an intermediate junction, while road progress probabilities depend on the current population congestion. The policy is frozen within each critic-training block and updated only after the block is completed. The final experiment package contains both a large-population implementation at $N = 20000$ and small-population settings $N \in \{20, 50, 100, 500\}$, together with Dutch-critic and MC-LS-critic variants and model-based mean-field controls. The goal is to test whether forward-trained critics can be embedded in a policy-learning loop and whether the resulting policies approach the numerical equilibrium references used by the experiment and reduce the evaluated unilateral-deviation gaps. Because Environment C uses a transition-based arrival reward, it is not used to validate the current-state-reward assumptions of Experiment 1.

Why three environments? The environments are complementary rather than redundant. Environment B is the main policy-evaluation benchmark because S, N, T, and d can be controlled separately, making it suitable for systematic scaling and resource measurements. Its transition kernel is independent of the population distribution, however, so population interaction enters only through the reward. Environment A supplements the theory validation with population-dependent dynamics: both its transition law and reward depend on the empirical population distribution.

Environment C addresses a different question. Its small, transparent strategy space makes policy improvement, deviation payoffs, and equilibrium diagnostics tractable, while congestion still creates a nontrivial population interaction. The final package evaluates the same physical routing game at both small and large population sizes, but the big- and small-population implementations use different critic initialization, policy-improvement tolerances, reference-construction procedures, and finite-population estimators. We therefore treat them as two implementations/configuration families of one environment rather than as interchangeable replicas of a single numerical protocol.

F.1.2 Environment A: Beach-Bar-Inspired Ring Population Environment

State space, policy, and initialization Environment A is an adapted Beach-Bar-inspired finite-population model used only for Experiment 1 and the associated estimator checks. The state space is

$$\mathcal{X} = \{0, 1, \ldots, S-1\}, \qquad S = 20,$$

where the states form a ring and the bar is located at state 0. State 10 is the antipodal state. There are no absorbing states and no early termination; in particular, an agent may leave the bar after reaching it.

The action space is

$$\mathcal{A} = \{-1, 0, +1\},$$

corresponding to moving left, staying in place, or moving right on the ring. All agents use the same fixed stationary stochastic policy. In the order (left, stay, right),

$$\pi(\cdot \mid x) = \begin{cases} (0.2, 0.6, 0.2), & x = 0, \\ (0.6, 0.2, 0.2), & x = 1, \dots, 9, \\ (0.4, 0.2, 0.4), & x = 10, \\ (0.2, 0.2, 0.6), & x = 11, \dots, 19. \end{cases}$$

The policy is independent of time, population size, and population distribution. The initial distribution is uniform,

$$u_0(x) = \frac{1}{S},$$

and finite-population agents are initialized i.i.d. from u_0.

Population-dependent transition dynamics At each time step, an agent first samples an action from the fixed policy and is then blocked with probability

$$\kappa U_t^N(X_t^{i,N}), \qquad \kappa = 0.8.$$

A blocked agent remains at its current state; otherwise it executes the sampled action. Let

$$K_0(y \mid x) = \sum_{a \in \{-1,0,+1\}} \pi(a \mid x) \mathbf{1}\{y = (x + a) \bmod S\}$$

denote the policy-induced kernel without blocking. The population-dependent closed-loop kernel is

$$P^\pi(y \mid x, u) = \big(1 - \kappa u(x)\big) K_0(y \mid x) + \kappa u(x) \mathbf{1}\{y = x\}.$$

Thus, the transition law depends explicitly on the current population distribution, although the functional form itself has no exogenous time dependence. Only the current state and its two ring neighbors are reachable in one step.

Population updates are synchronous. All agents use the same pre-transition empirical distribution U_t^N within a time step. Conditional on the current joint population state, the action and blocking draws are independent across agents. There is no additional common noise, shared capacity constraint, or collision mechanism.

Reward, horizon, and terminal convention Let

$$\mathrm{dist}(x, 0) = \min\{x, S - x\}$$

be the ring distance from the bar. The reward is

$$r(x, u) = -\left(\frac{\mathrm{dist}(x, 0)}{S/2}\right)^2 - u(x),$$

and the realized reward is

$$R_{t+1}^{i,N} = r(X_t^{i,N}, U_t^N).$$

Hence the reward is deterministic conditional on the current state and population distribution, lies in $[-2, 0]$, and does not depend on the next state. It is a negative distance-plus-congestion objective rather than a sparse positive reward.

The formal experiments use

$$T = 40, \qquad \gamma = 0.95.$$

An episode therefore contains $T + 1 = 41$ state slices, 40 transitions, and 40 rewards. There is no early termination. The final source-state reward at $t = T - 1$ is retained, while the terminal value and bootstrap are zero. The terminal feature table is physically zero. For trace-level diagnostics, however, a zero terminal feature does not erase the history already carried by the trace, and the Beach-Bar analysis retains the terminal transported-trace slice for those quantities whose definitions include $t = T$.

Critic features Let $e_x \in \mathbb{R}^S$ be the one-hot state vector. Define the four-dimensional time basis

$$\psi_A(t) = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 \\ \cos(2\pi t/T) \\ \sin(2\pi t/T) \\ \cos(4\pi t/T) \end{bmatrix}.$$

For $t < T$,

$$\phi(t, x) = e_x \otimes \psi_A(t),$$

and we set

$$\phi(T, x) = 0.$$

Thus $d = 4S = 80$. The feature map is deterministic and uses no transition-model, oracle-value, mean-field-occupancy, or future-sample information.

Experiment-1 configuration and mean-field reference Experiment 1 varies only the population size:

$$N \in \{50, 100, 200, 400, 800, 1600, 3200, 6400, 12800\}.$$

For each N, we simulate 100 complete finite-population trajectories. Every replicate resamples the initial population and reinitializes the critic parameters and traces. All paired estimators for a fixed $(N, \text{replicate})$ use the same realized population trajectory. Different values of N use separately derived random streams and are not coupled through a common random-number prefix.

The mean-field reference uses the same initial law, policy, transition rule, and reward parameters, replacing the empirical distribution by the deterministic population law. Specifically,

$$u_0(x) = \frac{1}{S}, \qquad u_{t+1}(y) = \sum_{x \in \mathcal{X}} u_t(x) P^\pi(y \mid x, u_t).$$

The associated mean-field trace and critic references are computed by deterministic finite-state summation rather than by simulating a very large population. The LSTD/MC-LS observers replay the same formal rollout identifiers and therefore do not create additional independent population replicates. In those observer checks, the general-λ and $\lambda = 1$ systems compare private and compressed regression statistics on the same data; the direct-MC tape is an additional numerical identity check, not a second mean-field value oracle.

F.1.3 Environment B: Garnet-Inspired Time-Varying Three-Region Environment

State space and three-region structure Environment B is a synthetic Garnet-inspired finite-state policy-evaluation environment with three state regions, a shared time-varying latent world, and population-dependent rewards. The simulator directly implements the fixed-behavior closed-loop kernel $P_t(y \mid x)$; there is no explicit action tensor or separately learned behavior policy.

The state space is

$$\mathcal{X} = \{0, \ldots, S-1\},$$

partitioned into an ordinary region O, an entrance region E, and a reward region G. The region sizes are

$$n_E = \max\{1, \lfloor 0.08S + 0.5 \rfloor\}, \qquad n_G = \max\{1, \lfloor 0.01S + 0.5 \rfloor\}, \qquad n_O = S - n_E - n_G.$$

For the formal S-grid,

S	64	128	256	384	512
n_O	58	117	233	349	466
n_E	5	10	20	31	41
n_G	1	1	3	4	5

and the default $S = 256$ therefore gives $(n_O, n_E, n_G) = (233, 20, 3)$. Agents are initialized i.i.d. uniformly over O,

$$u_0(x) = \begin{cases} 1/n_O, & x \in O, \\ 0, & x \notin O. \end{cases}$$

The region G is rewarding but is neither absorbing nor terminal: its outgoing transition returns the agent to O, so an agent may visit G repeatedly within the finite horizon.

Static heterogeneity and time-varying latent world Each state receives a static embedding

$$g_x \in [-1, 1]^5$$

with i.i.d. uniform coordinates. Let $W \in \mathbb{R}^{5\times 3}$ have i.i.d. standard-normal entries and define

$$h_x = \tanh\left(\frac{W^\top g_x}{\sqrt{5}}\right).$$

Independently, with Ξ_{xy} i.i.d. standard normal, define

$$B_{xy} = \frac{g_x^\top g_y}{5} + 0.5\Xi_{xy}.$$

These static quantities remain fixed within one environment world.

Time variation is generated by two AR(1) fields,

$$Z_t \in \mathbb{R}^{S\times S}, \qquad Z_t^q \in \mathbb{R}^{S\times 3},$$

with

$$Z_{t+1} = 0.9Z_t + \sqrt{1-0.9^2}\,\varepsilon_{t+1}, \qquad Z_{t+1}^q = 0.9Z_t^q + \sqrt{1-0.9^2}\,\varepsilon_{t+1}^q,$$

where the initial fields and innovations contain i.i.d. standard-normal entries under the frozen world seed. The world seed fixes the static embeddings, edge scores, and the complete latent time path. Resetting an episode under the same (S, seed) therefore replays the same sequence of transition kernels while resampling the finite population. Experiment 1 uses one world, $(S, \text{seed}) = (256, 1)$. In the formal Experiment 2 grid, the five values of S combined with the three formal seeds produce 15 distinct worlds; configurations that differ only in N, T, or d reuse the corresponding world.

Transition dynamics For $x \in O$, the probability of choosing target region E is

$$q_t(x, E) = \sigma(\text{logit}(0.02) + 0.3h_{x,E} + 0.15\tanh Z_t^q(x, E)),$$

with $q_t(x, O) = 1 - q_t(x, E)$ and $q_t(x, G) = 0$. For $x \in E$,

$$\big(q_t(x, O), q_t(x, E), q_t(x, G)\big) = \text{softmax}(\log(0.60, 0.15, 0.25) + 0.3h_x + 0.15\tanh Z_t^q(x, :)),$$

while for $x \in G$,

$$q_t(x, O) = 1, \qquad q_t(x, E) = q_t(x, G) = 0.$$

Within a target region b, the dense component is

$$D_t^{(b)}(x, y) = \frac{\exp(B_{xy} + 0.3\tanh Z_t(x, y))}{\sum_{z\in b}\exp(B_{xz} + 0.3\tanh Z_t(x, z))}, \qquad y \in b.$$

The local component $L_b(x, \cdot)$ keeps the agent at x if $x \in b$; otherwise it assigns equal mass to the at most three states in b whose embeddings are closest to g_x. The closed-loop transition is

$$P_t(y \mid x) = q_t(x, b(y))\left[0.7L_{b(y)}(x, y) + 0.3D_t^{(b(y))}(x, y)\right].$$

Rows are normalized after construction to remove floating-point roundoff, and the sampling CDF is forced to end at one. Structural zeros such as $O \to G$ and $G \to E/G$ are preserved.

The transition kernel does not depend on the population distribution. Population interaction in this environment enters only through the reward. The AR fields are shared exogenous latent variables, so the environment does contain common randomness at the world level; conditional on the fixed world and current population state, however, the agents' next-state draws are independent. The latent world is replayed across episodes rather than resampled at each reset.

Population-dependent deterministic reward Let

$$m_G(u) = \sum_{x\in G} u(x), \qquad \theta_G = \frac{n_G}{S}.$$

The pre-transition reward is

$$r(x, u) = \mathbf{1}\{x \in G\}\frac{R_0}{1 + \eta_r m_G(u)/\theta_G}, \qquad R_0 = 1, \quad \eta_r = 0.2,$$

so that

$$R_{t+1}^{i,N} = r(X_t^{i,N}, U_t^N).$$

Rewards are deterministic conditional on the current state and empirical population distribution and lie in $[0, 1]$. The population statistic includes the tagged agent itself. Reward sparsity means only that O and E have zero reward; G is not a terminal goal state. A transition into G receives no immediate arrival reward; if the agent first enters G at X_T, no post-terminal reward is added. The formal discount is $\gamma = 1$ and $V_T = 0$.

Critic features A full 64-dimensional state descriptor is constructed as

$$F^{\text{full}}(x) = \Big[1,\ \mathbf{1}_E(x),\ \mathbf{1}_G(x),\ g_{x,1}, \ldots, g_{x,5},\ \tanh\Big(g_x A/\sqrt{5} + b\Big)\Big],$$

where $A \in \mathbb{R}^{5\times 56}$ has i.i.d. standard-normal entries and $b \in \mathbb{R}^{56}$ has i.i.d. $\text{Uniform}[-1, 1]$ entries under the feature seed. The absolute-time basis is

$$\begin{aligned} c(t) = \big[1,\ & t/2000, \\ & \sin(2\pi t/2048),\ \cos(2\pi t/2048), \\ & \sin(2\pi t/512),\ \cos(2\pi t/512), \\ & \sin(2\pi t/128),\ \cos(2\pi t/128)\big]. \end{aligned}$$

For $d \in \{32, 64, 128, 256, 512\}$ let $p = d/8$ and define, for $t < T$,

$$\phi^{(d)}(t, x) = \frac{1}{\sqrt{512}} F^{\text{full}}(x)_{1:p} \otimes c(t).$$

The d-grid uses nested prefixes of the same descriptor table and a fixed scale $1/\sqrt{512}$. The features share the public state embedding and region labels with the environment but do not use the random edge scores, AR fields, true transition matrix, exact occupancy flow, oracle value, or future samples.

The terminal convention is implemented at the prediction/update layer: $V_w(T, x) = 0$ and learners skip terminal next-feature/transport terms. The low-level feature generator itself is not assumed to overwrite an entire terminal feature table with zeros. The Experiment 1 mean-field and observer constructions use their own explicit zero terminal feature convention.

Experiment-1 and Experiment-2 configurations The default Experiment 2 configuration is

$$S = 256, \qquad N = 20000, \qquad T = 500, \qquad d = 512.$$

The formal grids are

Study	Varied quantity	Values
Exp. 1	N	$500, 1000, 2000, 5000, 10000, 20000, 50000, 100000$
Exp. 2	N	$5000, 10000, 20000, 50000, 100000$
Exp. 2	S	$64, 128, 256, 384, 512$
Exp. 2	T	$100, 250, 500, 1000, 2000$
Exp. 2	d	$32, 64, 128, 256, 512$

with all unlisted quantities held at their default values. The four Experiment 2 sweeps share one default configuration, giving 17 unique configurations after deduplication. The saved main grid contains a single run for each method/configuration/seed triple; the common default point is reused across the four plotted sweeps rather than rerun four times.

Changing N leaves the world and feature map unchanged but changes the empirical occupancy, reward, and finite-N evaluation reference. Changing T truncates the same latent world path while changing the learning/evaluation horizon. Changing d changes only the nested representation. Changing S creates a new world and region partition.

In addition to the 612-run formal grid, the final package contains several completed supplementary configurations that are kept separate from the main statistical grid. A 50-episode default-configuration extension uses seed 1 and the nine pre-extension methods (the formal twelve-method set excluding DoubleSWEET-MC-LS and the two LSTD methods). A subsampling-sensitivity study evaluates the two subsampled methods at budgets $1000, 2000, 5000, 10000$ with seeds $1, 2, 3$ and

five episodes. Serial timing supplements evaluate the same nine pre-extension methods at the default configuration (seed 1, five episodes) and the Dutch pair at $N = 100000$ (seed 1, one episode). These runs are not pooled with the three-seed, five-episode main-grid estimates, and the larger proposed 50-episode-by-five-seed full grid was not executed.

For Experiment 1, $S = 256$, $T = 500$, $d = 512$, and the world seed is fixed to 1. Each N uses 100 formal population rollouts. Across N, the initial-state and transition random-number streams use common prefixes because N is not part of those stream keys; since the transition kernel is population-independent, the smaller-population state trajectories are prefixes of the larger ones, while rewards can still differ through the empirical population occupancy. Experiment 2 uses seeds $1, 2, 3$, and each method run contains five consecutive training episodes. Within one run, learner parameters and the accumulated least-squares/model statistics that are designed to persist are warm-started across episodes, while episode-specific traces and histories are reset.

Mean-field and finite-population references For a fixed world, the exact single-agent marginal occupancy satisfies

$$u_{t+1}(y) = \sum_x u_t(x) P_t(y \mid x).$$

Experiment 1 computes its mean-field trace, mass, and critic references by deterministic finite-state recursion.

Experiment 2 evaluates value error against a finite-population tagged-agent conditional expectation. Let

$$p_t = \sum_{x \in G} u_t(x).$$

Conditional on the tagged agent being in G, the number of the other $N - 1$ agents in G is

$$B_t \sim \mathrm{Binomial}(N - 1, p_t),$$

which yields

$$\bar{r}_t^N(x) = \mathbf{1}_G(x) \mathbb{E}\left[\frac{1}{1 + \eta_r(1 + B_t)/(N\theta_G)}\right].$$

The explicit $1 + B_t$ retains the tagged agent's self-effect. The reference value satisfies

$$V_T^N(x) = 0, \qquad V_t^N(x) = \bar{r}_t^N(x) + \sum_y P_t(y \mid x) V_{t+1}^N(y).$$

The numerical implementation evaluates the binomial expectation with a total tail truncation tolerance of 10^{-13}; oracle quantities are used only for evaluation and representation diagnostics, not for the sample-based learner updates.

F.1.4 Environment C: Two-Stage Congestion-Routing Environment

Network, population, and initialization Environment C is a custom synthetic finite-population two-stage congestion-routing game used for Experiment 3. The final package contains a large-population implementation and a small-population implementation of the same physical model. The physical state space contains $S = 15$ states,

$$O, A_1, A_2, A_3, B_1, B_2, B_3, J, C_1, C_2, C_3, D_1, D_2, D_3, G,$$

with directed topology

$$O \longrightarrow \begin{cases} A_1 \to A_2 \to A_3 \\ B_1 \to B_2 \to B_3 \end{cases} \longrightarrow J \longrightarrow \begin{cases} C_1 \to C_2 \to C_3 \\ D_1 \to D_2 \to D_3 \end{cases} \longrightarrow G.$$

Road states also admit self-loops representing waiting. There are no backward moves, cross-road jumps, or mid-road route changes, and G is absorbing.

The final package uses

$$N \in \{20, 50, 100, 500, 20000\}.$$

The small-population grids use $N \in \{20, 50, 100, 500\}$, while the large-population implementation fixes $N = 20000$. These settings are not treated as a single one-factor-at-a-time N-sweep because

the big and small implementations also differ in training budgets, critic initialization, and reference construction. In every case,

$$u_0 = \delta_O,$$

so all agents begin deterministically at the origin. Agents that reach G remain in the population; the normalization denominator remains N throughout the episode.

Route choice and congestion-dependent dynamics At O, an agent selects route A or B; at J, it selects route C or D. Waiting versus advancing on a road is an environment outcome rather than a control action. For road $e \in \{A, B, C, D\}$, let $\mathcal{R}_e$ denote its three physical states and define

$$\rho_e(u) = \sum_{x \in \mathcal{R}_e} u(x), \qquad q_e(u) = \mathrm{clip}(b_e - \kappa_e \rho_e(u), 0.1, 0.9)\,,$$

with

e	A	B	C	D
b_e	0.8	0.6	0.8	0.6
κ_e	0.5	0.1	0.5	0.1.

Let p_0 denote the probability of choosing A at O and let p_{t+1} denote the probability of choosing C at J at time t. The policy-induced kernel is

$$P_t^p(y \mid x, u) = \begin{cases} p_0, & x = O,\ y = A_1, \\ 1 - p_0, & x = O,\ y = B_1, \\ p_{t+1}, & x = J,\ y = C_1, \\ 1 - p_{t+1}, & x = J,\ y = D_1, \\ q_{e(x)}(u), & x \in \bigcup_e \mathcal{R}_e,\ y = s(x), \\ 1 - q_{e(x)}(u), & x \in \bigcup_e \mathcal{R}_e,\ y = x, \\ 1, & x = G,\ y = G, \\ 0, & \text{otherwise}, \end{cases}$$

where $s(x)$ is the forward successor on the current road. Population updates are synchronous, and all agents use the same pre-transition road loads. Conditional on the current joint state and policy, the individual simulator draws are independent. There is no common exogenous shock or random topology.

The two production implementations realize this population law differently. The big Private Dutch and the MC-LS families use individual-agent simulations. The big Dutch DoubleSWEET implementation instead draws exact binomial aggregate transition counts from each source state. This produces the same lumped finite-population transition law at the count level but is not the same realized per-agent trajectory as the private simulator. Small-population sample-based methods use individual-agent simulation. Hence equal seed labels do not imply pathwise pairing between all Environment C methods.

Arrival reward, horizon, and terminal convention Environment C uses a first-arrival transition reward,

$$R_{t+1}^{i,N} = \mathbf{1}\{X_t^{i,N} \neq G,\ X_{t+1}^{i,N} = G\}.$$

An agent receives reward one exactly when it first enters G and zero on all other transitions. There is no travel cost, waiting penalty, direct congestion penalty, or additional terminal bonus. The formal physical parameters use

$$T = 30, \qquad \gamma = 0.99.$$

If τ_G is the first state time at which the tagged agent reaches G, then

$$J^N = \mathbb{E}\left[\sum_{t=0}^{T-1} \gamma^t R_{t+1}\right] = \mathbb{E}\left[\gamma^{\tau_G - 1} \mathbf{1}\{\tau_G \leq T\}\right].$$

An arrival on the final transition still receives $R_T = 1$. The terminal value is $V_T = 0$, and G remains in the population after arrival.

Policy parameterization and outer improvement The behavioral policy depends only on time and the agent's own state. It does not observe the population distribution, first-stage route history, or other private history. The parameter vector has the interpretation

$$p_0 = \Pr(A \mid O), \qquad p_{t+1} = \Pr(C \mid J \text{ at time } t).$$

Effective decision probabilities are initialized to 0.8; irrelevant coordinates are initialized to 0.5. For $T = 30$, junction coordinates $t = 4, \ldots, 26$ can affect the return before the deadline. During training, effective probabilities are clipped to $[0.02, 0.98]$. The policy is frozen within each critic-training block.

At outer iteration k the common damping factor is

$$\eta_k = \frac{0.1}{1 + k/40}.$$

For a coordinate with estimated advantage d_j, the target is one if d_j is above the implementation-specific positive tolerance, zero if it is below the negative tolerance, and the current probability otherwise; the damped update is then clipped to $[0.02, 0.98]$.

The big and small implementations differ numerically. In the big implementation the action-value difference is multiplied by γ and the advantage tolerance is 10^{-6}; in the small implementation the raw entrance-value difference is used and the tolerance is 10^{-4}. Both require at least 20 visits to the two relevant entrance states in the current block before a sample-based policy coordinate is updated. The visit count is an occupancy count at the corresponding time-state pair, not merely the number of new road entries. Model-based mean-field controls are not subject to this sample-visit threshold. The training clip does not restrict the reference search or unilateral-deviation class, which allow action probabilities in $[0, 1]$.

Critic representation The critic is a time-state one-hot value function. Conceptually,

$$V_w(t, x) = \phi(t, x)^\top w, \qquad \phi(t, x) = e_{(t,x)}$$

for nonterminal non-G coordinates, with zero value at $t = T$ and at G. The nominal space contains

$$d_{\text{nominal}} = 14T = 420$$

coordinates, while only 339 nonterminal non-G time-state pairs are structurally reachable from the deterministic origin under the routing topology. The big implementation uses a reachability mask within the nominal space; the small implementation's explicit feature tensor indexes only the 339 reachable coordinates. Both production kernels store a zero-padded value table of shape $(T+1, 15)$. Thus 420 versus 339 is an implementation/representation distinction, not a d-sweep.

The Dutch critic initialization also differs: the big implementation initializes reachable nonterminal values to 0.9, while the small implementation initializes them to zero. MC-LS and DoubleSWEET-MC-LS start each policy block from zero least-squares statistics and solve a new ridge system at the end of the block rather than warm-starting the previous block's regression solution.

Completed Experiment-3 configurations The final rerun contains several completed training configurations. Because the experiment package itself does not encode a final paper-level "main versus supplementary" selector, we describe their execution role without treating every completed diagnostic as an independent primary experiment.

The big main, follow-up, and MC configurations use training seeds $101, 202, 303$ for their sample-based methods; the big extended run adds $404, 505$. The small reproduced and small MC configurations use five training seeds, $11, 22, 33, 44, 55$, and the small extended configuration uses ten seeds, $11, 22, \ldots, 99, 110$. Runs that share a training prefix under the same seed but use different budgets are not pooled as additional independent replicates. Diagnostic smoke/quick configurations are kept separate from the reported scientific settings.

Mean-field controls and finite-population references For a fixed policy p, the deterministic mean-field recursion is

$$u_0 = \delta_O, \qquad u_{t+1}(y) = \sum_x u_t(x) P_t^p(y \mid x, u_t),$$

Table 5: Completed non-smoke Environment C configurations in the final rerun. Sample-based methods are listed; model-based controls are described in the text.

Impl.	Configuration	Sample-based methods	N	Updates × eps./update
big	main	Private Dutch, DoubleSWEET	20000	80×40
big	follow-up	DoubleSWEET	20000	400×80
big	MC	MC-LS, DoubleSWEET-MC-LS	20000	100×1
big	MC long-budget	MC-LS, DoubleSWEET-MC-LS	20000	1000×1
big	extended	Private Dutch, DoubleSWEET	20000	200×80
small	reproduced	Private Dutch, DoubleSWEET	$20, 50, 100, 500$	100×100
small	MC	MC-LS, DoubleSWEET-MC-LS	$20, 50, 100, 500$	100×100
small	extended	Private Dutch, DoubleSWEET	$20, 50, 100, 500$	200×200

with value recursion

$$V_T^p(x) = 0, \qquad V_t^p(x) = \sum_y P_t^p(y \mid x, u_t) \left[\mathbf{1}\{x \neq G, y = G\} + \gamma V_{t+1}^p(y) \right].$$

These are finite-state model calculations. The model-oracle/MF-DP controls may use these values directly for policy improvement; they therefore have strictly richer information than the sample-based critics.

The finite-population reference preserves the tagged agent's $1/N$ contribution to road congestion and evaluates deviations within the same time–own-state policy class as the training policy. The big and small reference constructors are different. The big implementation first obtains a numerical mean-field candidate and then performs finite-population correction/refinement using tagged simulations and a mean-field control variate. The small implementation constructs a separate numerical reference for each N using tagged-agent payoff coefficients, damped policy iteration, and fixed restarts; it does not initialize from the big mean-field solution and does not use the big control variate. Both references are numerical approximate candidates rather than proven exact or unique Nash equilibria.

Some completed batches reuse compatible references generated earlier in the same final rerun rather than rerunning the reference search. Such reuse is not counted as an additional independent reference search or holdout. The principal reference/validation budgets are as follows. The big main reference (reused by the follow-up and both big MC configurations) uses 8 refinement rounds with 32768 tagged trials per round, a final-candidate validation budget of 131072, and a reference holdout budget of 262144. The big extended reference uses 12 rounds with 131072 trials per round, followed by budgets 524288 and 1048576, respectively. For each small-population N, the reproduced reference (reused by the small MC configuration) uses 100 rounds with 4096 replications and two restarts, followed by 65536 final-candidate and 65536 per-restart reference-holdout replications. The small extended reference uses 200 rounds with 16384 replications, two restarts, and 262144 replications for each of the final-candidate and per-restart holdout evaluations. In the big implementation the trial budget is counted per forced first-stage route; in the small implementation one replication jointly produces four correlated counterfactual branches, so those branches are not counted as four independent samples. Final finite-population validation is applied to the final policy p_K, not to auxiliary tail-averaged training diagnostics.

F.1.5 Compared Methods and Experimental Protocol

The algorithms and their theoretical properties are defined in the main text and preceding appendices. This subsection records only the experimental roles of the executed methods, their frozen hyperparameters, and the information shared across comparisons.

Objects compared in Experiment 1 Experiment 1 is a theory-validation study rather than an algorithm-ranking benchmark. For the original Dutch construction, the main objects are the finite-population private trace, the finite-population compressed trace, and the corresponding mean-field quantities. The recorded diagnostics include population and transition-pair masses, Dutch credit masses, normalized conditional traces, one-step update maps evaluated at a common parameter pair, the adaptive critic parameters generated by each construction, and value readouts.

The Garnet-based validation additionally tracks conventional accumulating-λ traces and a prefix/MC trace construction. The LSTD/MC-LS observer replays the same formal population rollouts and

Table 6: Methods in the formal Experiment 2 grid.

Category	Methods	Experimental role
Compression pairs	Private Dutch / DoubleSWEET; MC-LS / DoubleSWEET-MC-LS; LSTD(λ) / DoubleSWEET-LSTD(λ)	Compare one estimator family before and after state-wise compression.
Other sample-based baselines	TD(0); MC-SGD	Standard direct value-learning references without the proposed state-wise trace compression.
Subsampled baselines	Subsampled Private Dutch; Subsampled MC-LS	Reduce learner sample usage rather than compressing the full population information.
Model-estimation-then-Bellman	Model-Based Tabular; Model-Based Projected	Estimate empirical reward/transition models from samples and then evaluate backward.

compares private and compressed LSTD/MC-LS sufficient statistics (H, g), ridge solutions w, full-query value readouts, and trace-bridge quantities for general λ and for the episodic $\lambda = 1$ case. The direct-MC tape is used to check the algebraic $\lambda = 1$ identity on the same data; it is not a second independent population replicate or a separate mean-field oracle. Consequently, not every LSTD/MC-LS observer quantity has a private–compressed–mean-field three-way version.

Methods compared in Experiment 2 The formal Experiment 2 grid contains twelve methods:

The formal grid contains 17 unique configurations, three seeds, and twelve methods, hence

$$17 \times 3 \times 12 = 612$$

completed method–configuration–seed runs, each with five consecutive training episodes. Within a run, episode-specific traces/histories are reset, while parameters and estimator statistics designed to accumulate across episodes are retained.

For the model-based baselines, let $n_t(x)$ be the accumulated visit count, $C_t(x, y)$ the accumulated transition count, and $s_t(x)$ the accumulated reward sum. With transition prior $\rho = 0.1$,

$$\widehat{r}_t(x) = \begin{cases} s_t(x)/n_t(x), & n_t(x) > 0, \\ 0, & n_t(x) = 0, \end{cases} \qquad \widehat{P}_t(y \mid x) = \frac{C_t(x, y) + \rho/S}{n_t(x) + \rho}.$$

The tabular baseline applies a backward Bellman recursion to this empirical model. The projected baseline applies the same empirical Bellman step and then performs a ridge projection onto the common feature representation. These methods never call the true transition matrix or the evaluation oracle in their learner updates.

The main subsampling budget is

$$K = \min(N, 2000).$$

At the beginning of each episode, the two subsampled methods draw the same K agent IDs without replacement and sort them. The environment and rewards are still generated by the complete N-agent population; subsampling occurs only at the learner-observation stage. The subset is redrawn across episodes rather than fixed for an entire multi-episode run.

Methods compared in Experiment 3 Environment C uses four sample-based critic families: Private Dutch, DoubleSWEET, MC-LS, and DoubleSWEET-MC-LS. It also includes true-model mean-field controls (the big "model oracle" and the small "MF-DP" control) and an untrained initial-policy control in the small-population evaluations. No empirical Model-Based Tabular/Projected baseline, subsampled baseline, TD(0), MC-SGD, or general-λ LSTD method is part of the final Environment C package.

The production implementations are not identical to their Experiment 2 engineering counterparts. Environment C's private Dutch implementation stores the episode's visited time-state coordinates as a sparse/history representation, while the big Dutch DoubleSWEET simulator samples aggregate transition counts. Environment C MC-LS stores an episode state history and performs the

Table 7: Principal learner and policy-learning hyperparameters.

Setting	Value	Scope
Beach-Bar Dutch step size	$\alpha = 0.05$	Environment A
Beach-Bar trace parameter	$\lambda = 0.8$	Environment A Dutch/general-λ observer
Beach-Bar averaged ridge	10^{-8}	LSTD/MC-LS observer systems
TD(0) step size	10	Environment B Experiment 2
MC-SGD step size	1	Environment B Experiment 2
Dutch step size	3	Environment B Dutch families
Trace parameter	$\lambda = 0.99$	Environment B trace/LSTD families
Averaged-system ridge	10^{-8}	Environment B least-squares/projected estimators
Model transition prior	0.1	Environment B empirical model baselines
Main subsample budget	2000	Environment B subsampled baselines
Routing Dutch λ	0.95	Environment C big and small
Routing MC ridge	10^{-8}	Environment C MC-LS families
Routing policy rate	$0.1/(1 + k/40)$	Environment C outer update

return accumulation after the episode, whereas DoubleSWEET-MC-LS maintains state-wise incoming credits and compressed sufficient statistics. These implementation choices belong to the routing experiment and should not be used to redefine the Experiment 2 memory model.

Because Environment C contains policy learning, there is no global pathwise-pairing statement across methods. The big Private Dutch and DoubleSWEET implementations already use different population samplers. The big MC pair and the small sample-based implementations share random streams while their policies agree, but once the learned policies diverge the realized trajectories and rewards need not remain identical. The model-based controls have true-model access and are interpreted as information-richer references rather than sample-matched competitors.

Hyperparameters and parameter selection The principal frozen hyperparameters are summarized below.

For Environment B, the final rerun repeated the previous learning-rate calibration on

$$S = 128, \qquad N = 5000, \qquad T = 250, \qquad d = 512,$$

using seeds $9002, 9003$, three episodes, and candidate step sizes $\{0.1, 1, 3, 10\}$. The selection criterion was the seed-mean final-episode relative RMSE; the Dutch step size was selected jointly from the Private Dutch and DoubleSWEET calibration scores. The repeated calibration agreed with the frozen values $(10, 1, 3)$ for TD(0), MC-SGD, and the Dutch pair. Because these values pre-existed in the frozen configuration, we treat the final rerun as a confirmation rather than as evidence that the original choices were made blindly before all historical experiments.

The routing Dutch step-size schedules differ between the two implementations:

$$\alpha_m^{\text{big}} = \frac{1}{1 + m/100}, \qquad \alpha_m^{\text{small}} = \frac{0.8}{1 + m/100},$$

where m is the episode index within a fixed-policy training block. The big Dutch critic initializes reachable values to 0.9; the small Dutch critic initializes them to zero. The two implementations also use the distinct policy-improvement tolerances described in Section F.1.4. The least-squares ridge 10^{-8} is applied to an averaged population-step system. Because different experiments use different averaging denominators, the corresponding unaveraged-system ridge is not a universal constant across Experiment 2, the LSTD/MC-LS observer, and Environment C.

Fair-comparison and information-access protocol In Experiment 2, all twelve methods at a fixed configuration, seed, and episode receive the same generated population data. The runner stores per-episode hashes of the (x, y, r) stream, and the report stage verifies the expected cross-method pairing. The two subsampled methods additionally store and verify their sampled-ID hashes. All feature-based methods use numerically identical features for a fixed (S, d, seed) configuration. DoubleSWEET methods still process the complete observed population; their resource savings come from compressed retained state, not from discarding agents.

Environment A uses within-replicate pairing: private, compressed, and observer quantities for one $(N, \mathrm{replicate})$ are computed from the same population rollout. Different values of N use distinct random streams. Environment B Experiment 1 instead uses common random-number prefixes across N for the population state trajectories, while its bootstrap analysis resamples each N separately and therefore does not construct a joint paired bootstrap across the entire N-grid.

Environment C shares the same physical model, policy parameterization, and outer-update structure within each implementation family, but does not provide a global realized-trajectory pairing across methods. Its sample-based critic updates do not query the true transition model or the numerical reference policy. The mean-field controls and finite-population reference builders do use the true model and are kept separate in interpretation.

F.1.6 Evaluation Metrics and Statistical Protocol

The three experiments answer different questions and therefore use different primary metrics. We state the aggregation order explicitly because several Experiment 1 objects have different valid time ranges, while Experiment 3 uses different large- and small-population estimators.

Experiment 1: finite-population discrepancies and empirical scaling For a time-indexed theoretical quantity Q, let $e_{Q,r,t}(N)$ denote the prescribed nonnegative discrepancy for replicate r. The main time-series statistic is

$$E_Q(N) = \max_{t \in \mathcal{T}_Q} \frac{1}{R} \sum_{r=1}^{R} e_{Q,r,t}(N), \qquad R = 100,$$

where the valid time set $\mathcal{T}_Q$ depends on the object. The operation order is fixed: the per-replicate norm is computed first, then averaged across population trajectories, and only then maximized over the valid time range. For value-readout quantities, the maximum over queried states is taken before the replicate average. Final regression quantities such as the LSTD/MC-LS observer (H, g, w, V) discrepancies are instead computed once per trajectory at the end of the rollout and then averaged across replicates; they are not converted into artificial time maxima.

The principal norms are:

- empirical population and transition-pair masses: L^1;
- trace/credit-mass tables: $\|M\|_{1,2} = \sum_x \|M(x,:)\|_2$;
- conditional traces: empirical-occupancy-weighted statewise ℓ_2;
- feature-Gram/Dutch-operator and LSTD/MC-LS observer H discrepancies: operator norm;
- one-step updates, regression vectors, and parameter differences: Euclidean norm;
- value readouts: maximum absolute prediction difference over the specified query set.

The Beach-Bar state/trace diagnostics generally include the terminal transported slice $t = T$, while its one-step update diagnostics use $t < T$. The original Garnet time-series diagnostics use $t = 0, \ldots, T-1$. The LSTD/MC-LS trace bridge retains the terminal transported slice in Environment A but not in Environment B. These conventions reflect the stored objects and are not post-hoc truncations.

For empirical scaling, we fit

$$\log E_Q(N) = a_Q + b_Q \log N$$

by ordinary least squares. Environment A uses all nine formal N values for the full fit and the five points with $N \geq 800$ for the prespecified tail fit. Environment B uses all eight formal N values and the five points with $N \geq 5000$. The Environment A analysis uses natural logarithms and the original Environment B analysis uses base-10 logarithms; the fitted slope is invariant to this common change of base. The $N^{-1/2}$ reference line is anchored to the first tail point rather than fitted as an additional free curve.

The bootstrap unit is one complete finite-population trajectory. We use 2000 trajectory-level resamples and report pointwise 95% percentile intervals for the scalar $E_Q(N)$ statistic. Environment A

and the LSTD/MC-LS observer analyses additionally refit bootstrap slopes where specified by their analysis code. The original Environment B slope script records point slopes but does not provide a separate original-slope confidence interval. Although Environment B shares random-number prefixes across N, its bootstrap draws are performed separately for each N; the resulting intervals should therefore not be interpreted as a joint paired confidence band across the N-grid.

The numerical-audit status is stored separately from run completion. Complete observer checkpoints are not silently filtered from the scientific summaries merely because a numerical audit flag failed; conversely, a missing replicate makes the corresponding quantity/configuration unavailable rather than being replaced by zero. Detailed audit outcomes are reported with the Experiment 1 results rather than treated as part of the experimental definition.

Experiment 2: value-estimation accuracy Experiment 2 evaluates each trained estimator against the finite-N reference $V_t^N(x)$ from Environment B. Let $\widehat{V}_t(x)$ be the final estimator after episode M and let u_t be the exact world occupancy used only for evaluation. Define

$$\mathrm{RMSE}_{\mathrm{occ}} = \left[\frac{1}{T}\sum_{t=0}^{T-1}\sum_{x} u_t(x)\big(\widehat{V}_t(x) - V_t^N(x)\big)^2\right]^{1/2},$$

$$V_{\mathrm{rms}} = \left[\frac{1}{T}\sum_{t=0}^{T-1}\sum_{x} u_t(x)\big(V_t^N(x)\big)^2\right]^{1/2}, \qquad \mathrm{RelRMSE} = \frac{\mathrm{RMSE}_{\mathrm{occ}}}{V_{\mathrm{rms}} + 10^{-12}}.$$

The terminal slice is excluded. The main tables and scaling figures use the final episode rather than averaging accuracy across the training episodes. For linear estimators, $\widehat{V}$ is generated from the final parameter; Model-Based Tabular instead uses its final tabular value table. We therefore describe the metric as evaluating the final estimator, rather than assuming every method has a single linear parameter vector.

For paired private/compressed estimators, we also report an occupancy-weighted prediction discrepancy obtained by replacing the oracle error with

$$\widehat{V}_t^{\mathrm{compressed}}(x) - \widehat{V}_t^{\mathrm{private}}(x).$$

Its relative version is normalized by V_{rms} without the additional 10^{-12} term used in the oracle RelRMSE denominator. Oracle feature projection is a separate representation diagnostic: it projects the finite-N reference using true occupancy weights and a numerical pseudoinverse without ridge, and is not the same object as the empirical Model-Based Projected baseline.

Experiment 2: learner-side memory The primary memory metric is the peak byte count of persistent learner-owned numerical buffers, reported as

$$\mathrm{MiB} = \frac{\mathrm{bytes}}{2^{20}}.$$

The implementation counts each unique underlying ndarray buffer once and includes the public feature and region arrays retained by the learner. For CSR count layers, the `data`, `indices`, and `indptr` buffers are counted explicitly. The main metric excludes simulator/oracle arrays, Python/Numba/BLAS runtime state, and temporary workspaces that are not persistently retained by the learner.

This accounting is based on the actual executed representations. For example, the Dutch implementations apply their state/time Kronecker structure without materializing a dense $d \times d$ Dutch operator; by contrast, MC-LS and LSTD-type methods retain genuine $d \times d$ regression statistics, which are included. Model-Based Projected constructs its projection matrices as episode-end temporaries, so these are not counted as persistent arrays. Model-transition counts begin sparse and are converted to dense storage when the actual sparse-buffer cost reaches the implemented dense-storage threshold.

A secondary workspace-inclusive estimate adds the major explicitly known temporary arrays, but it is not a complete malloc/BLAS/Python peak. Whole-process RSS is sampled separately at approximately 25 ms and includes simulator, oracle, runtime-library, allocator, and other process-level memory. Neither auxiliary quantity replaces the learner-owned-array metric in the principal memory comparisons.

Experiment 2: processing time Learner timing is separated from environment simulation. For a run with M episodes and horizon T, let t_{begin}, t_{online}, and t_{end} denote the total timed learner cost in episode initialization, all `observe` calls, and all episode-end operations. The principal metric is

$$t_{\text{amortized}} = 1000\,\frac{t_{\text{begin}} + t_{\text{online}} + t_{\text{end}}}{MT} \quad \text{ms/population step.}$$

The denominator is MT, not MNT and not the number of workers. Episode-end time contains the complete method-specific operation—for example a Monte Carlo pass, least-squares solve, or empirical-model Bellman pass—and is therefore not a pure linear-solver microbenchmark. Environment `step`, initial-state generation, oracle construction/evaluation, trajectory hashing, artifact serialization, plotting, and the pre-run JIT warm-up are outside the principal learner timers. Feature construction performed inside a learner update and the learner's memory bookkeeping are inside the timed path.

The final main grid records these metrics under a common server execution protocol, but under concurrent rather than isolated load. We therefore interpret them as measured processing costs under the recorded shared-node protocol and reserve the separate serial supplements for latency diagnostics. The serial supplements use different workload/concurrency conditions and are not pooled with the main-grid timing statistics.

Experiment 3: critic and MFG-solving metrics Environment C evaluates both critic quality and the policy-learning process. For a fixed policy p, let u^p and V^p denote the deterministic mean-field flow and Bellman value. The occupancy-weighted critic diagnostic is

$$\mathrm{RMSE}_{\mathrm{MF}}(w,p) = \left[\frac{\sum_{t=0}^{T-1}\sum_{x\neq G} u_t^p(x)\big(w_t(x) - V_t^p(x)\big)^2}{\sum_{t=0}^{T-1}\sum_{x\neq G} u_t^p(x)}\right]^{1/2}.$$

This is neither the Experiment 2 relative RMSE nor an error against an exact finite-N value table. The logging time differs between the implementations: in the big implementation the critic trained under p_k is evaluated against p_k before the subsequent policy update, whereas the small implementation logs w_k against the current p_k even though, for $k > 0$, w_k was produced by the previous policy block. We therefore do not merge the two critic-error trajectories under a single "current-policy fitted critic" interpretation.

Let p^* be the numerical finite-population reference and let μ_t denote the tagged-agent junction-arrival mass under the reference. Define

$$D_O = |p_0 - p_0^*|, \qquad D_J = \frac{\sum_{t\in\mathcal{T}} \mu_t |p_{t+1} - p_{t+1}^*|}{\sum_{t\in\mathcal{T}} \mu_t}, \qquad D = \frac{1}{2}(D_O + D_J).$$

The big implementation normalizes over all $t = 0, \ldots, 29$, while the small implementation sets the irrelevant $t < 4$ and $t > 26$ masses to zero before normalization. This distinction is kept explicit in the reported policy-distance metric.

The principal finite-population equilibrium target is the unilateral-deviation gap within the time–own-state policy class Π_{TS}:

$$\mathrm{Gap}_N(p) = \sup_{\pi'\in\Pi_{\mathrm{TS}}} J^N(\pi'; p^{-i}) - J^N(p; p^{-i}).$$

The deviation class may use pure action probabilities in $[0,1]$ and is not restricted by the training exploration clip. Tagged-agent simulation uses $N-1$ background agents and retains the tagged agent's own contribution to congestion. Both implementations estimate route/time-specific payoff coefficients and then form the best deviation within Π_{TS}; the big implementation additionally uses a mean-field control variate, whereas the small implementation uses direct finite-population coefficient estimates. Their 95% confidence constructions are therefore implementation-specific and are not pooled into a single universal interval formula. In the big implementation, the reported primary gap upper bound is the class-level bound associated with the control-variate records and is distinct from the separately stored coefficient-box diagnostic; in the small implementation, the reported interval is clipped to $[0,1]$. These confidence calculations apply to one independently selected candidate and the evaluated restricted deviation class; they are not simultaneous confidence guarantees across all methods, population sizes, seeds, and figures. The resulting gap is a restricted-class finite-population diagnostic, not the exact Nash gap of the unrestricted N-player game. A deterministic mean-field best-response gap is recorded separately and should not be conflated with the finite-population gap.

Independent replicates and reporting conventions For Experiment 1, one independent replicate is one complete population rollout at fixed N; each formal N uses 100 such rollouts. The LSTD/MC-LS observer replays the same rollouts and does not double the number of independent replicates.

For the Experiment 2 main grid, one independent replicate is one complete five-episode learner run under one of the three formal seeds $1, 2, 3$. The five episodes are a continuing training sequence, not five independent replicates. Main-grid curves report the arithmetic mean and sample standard deviation ($\mathrm{ddof} = 1$) across the three seeds. Single-seed long-budget or serial supplements are not assigned a three-seed error bar.

For Experiment 3, the independent training unit is one complete policy-learning run under one training seed. The number of seeds depends on the configuration: three for the big main, follow-up, and MC comparisons, five for the big extended and the small reproduced/MC settings, and ten for the small extended setting. Episodes and outer iterations within one policy-learning run are not independent replicates. Runs with different budgets that reuse a common training prefix under the same seed are not pooled as additional independent seeds. Finite-population holdout samples form a separate validation layer and are not counted as training replicates.

F.1.7 Implementation, Timing Protocol, and Computational Resources

The final rerun uses two distinct execution environments: Environments A–B and Experiment 2 were rerun on a single HPC compute node, whereas Environment C was rerun locally through a two-pipeline coordinator. We report these settings separately and do not present Environment C wall-clock times as if they were measured by the Experiment 2 learner-only timing protocol.

Software and numerical implementation The Environment A–B rerun uses Python 3.12.14, NumPy 2.3.5, SciPy 1.17.0, Numba 0.65.1, and OpenBLAS 0.3.30. Numerical-library thread pools are restricted to one thread for the Experiment 2 measurements. The Environment C rerun uses Python 3.12.10, NumPy 2.4.2, SciPy 1.17.1, and Numba 0.66.0 under Windows 11; its launcher also sets the major BLAS/OpenMP/Numba thread-limit environment variables to one, although an independent runtime thread-pool inspection was not recorded.

The high-frequency kernels use Numba JIT compilation without `parallel=True` or `prange`. The Experiment 2 kernels run without fast-math. Experiment 2 performs an explicit pre-run warm-up that exercises all twelve methods on tiny $d = 32$ and $d = 512$ configurations before formal learner timing starts. The saved JIT signatures show no new signature during the 760 recorded Experiment 2 learner runs. Experiment 1 uses persistent pinned workers with its own smaller warm-up paths, while Environment C warms representative production paths at the configuration-process level; these mechanisms are not identical and are therefore not described as one universal warm-up benchmark.

The least-squares solvers also differ by estimator. Experiment 2 MC-LS and Model-Based Projected solve symmetric averaged ridge systems by Cholesky factorization, with a least-squares fallback if factorization fails. Experiment 2 LSTD(λ) solves the generally nonsymmetric averaged system and falls back to least squares when the direct solve fails its residual check. The LSTD/MC-LS observer instead uses a general solve and marks an unavailable system on solver failure rather than applying the Experiment 2 fallback. Environment C's production MC-LS systems are diagonal in the time-state representation and are solved coordinatewise. These implementation choices are kept separate from the common mathematical ridge notation.

Experiment 2 timing boundaries For Experiment 2, episode-begin and episode-end operations are timed with `time.perf_counter()`, while each `observe` call is timed with `time.perf_counter_ns()`. The stored online-mean statistic averages all $M \times T$ observe calls in a run, and the online 95th percentile is computed over the same call-level array. The recorded episode-end time includes all method-specific work inside `end_episode()`, not just the underlying linear solver. The amortized metric is the one defined in Section F.1.6.

Environment simulation is tracked separately. Static environment/feature construction, oracle generation/loading, evaluation, data hashing, finite-value guards outside the learner call, artifact serialization, plotting, and JIT warm-up are not included in the principal learner timers. The runner stores

both an all-episode online mean and a final-episode online mean; whenever an online-mean value is reported, the corresponding field is named explicitly.

HPC execution protocol for Environments A–B The final Environment A–B rerun was executed through LSF on one compute node. The selected node contains two AMD EPYC 7763 processors with 64 physical cores each; the operating system exposed 128 logical CPUs with one thread per core, two NUMA nodes, and approximately 540 GB of visible RAM. The job requested 100 slots and `span[hosts=1]`. The project selected 100 distinct physical CPUs and pinned one Experiment 2 learner subprocess to each selected CPU. Numerical-library thread counts were restricted to one. The job did not request exclusive node ownership, and NUMA memory binding was not enforced; external system load is therefore not known to be identical across all measurements.

The Experiment 2 main grid uses a fresh pinned subprocess for each method, configuration, and seed combination. Recorded run intervals reach a project concurrency of 100 on the node, but this does not imply that exactly 100 `observe` kernels execute simultaneously at every instant. Jobs are admitted subject to the project's memory controller and progress asynchronously; method ordering is deterministic rather than randomized into perfectly balanced waves. Accordingly, the main-grid timing values are measurements under one common concurrent server protocol, not isolated single-process latency measurements.

The serial-latency and large-N timing supplements run under deliberately different concurrency conditions and are reported separately. Experiment 1 measures whole rollout/diagnostic tasks rather than the Experiment 2 learner-only begin/observe/end boundary and is not pooled with the Experiment 2 runtime comparison.

Experiment 2 memory implementation The principal memory curves use the actual learner-owned buffer accounting described in Section F.1.6. The private Dutch and DoubleSWEET Dutch implementations exploit the feature Kronecker structure to apply the Dutch operator without retaining a dense $d \times d$ Dutch matrix. This does not imply that every compressed estimator avoids d^2 storage: DoubleSWEET-MC-LS and DoubleSWEET-LSTD retain the corresponding $d \times d$ regression statistics, and the private LSTD implementation also retains a full regression matrix in addition to its per-agent trace representation.

For the empirical-model baselines, the transition-count layers are stored sparsely until the implemented sparse-buffer cost reaches the dense $S \times S$ storage threshold, after which the layer is converted to a dense `uint32` array. The tabular baseline persistently retains its $(T+1) \times S$ value table; the projected baseline constructs projection matrices at episode end as temporaries, so those matrices enter the workspace estimate rather than the persistent owned-buffer metric. RSS is monitored for the formal Experiment 2 runner but is treated only as an auxiliary whole-process diagnostic.

Environment C does not contain a corresponding production learner-owned-memory benchmark or RSS-monitoring pipeline. Its array shapes describe implementation state but are not presented as measured peak-memory results.

Environment C runtime and local execution Environment C was rerun under Windows 11 on a machine recorded as an Intel Core Ultra 7 155H with 16 physical cores, 22 logical CPUs, and approximately 31.5 GiB of visible RAM. A local coordinator launched one big-population and one small-population pipeline concurrently; within each package, configurations were executed sequentially in fresh configuration processes. No scheduler-level exclusive allocation, CPU pinning, NUMA binding, or per-learner hardware timeline was recorded.

The routing experiment uses `time.perf_counter()` block timers rather than the Experiment 2 per-observe instrumentation. Its reported training-block time includes the population simulation and the outer policy-improvement work. In particular, the big Dutch Private and DoubleSWEET implementations use different population simulators (individual-agent versus exact aggregate count sampling), so their Environment C wall-clock measurements cannot be interpreted as a pure trace-update microbenchmark. There is no Environment C analogue of the Experiment 2 amortized learner milliseconds per population step, and no isolated routing runtime rerun is part of the final package.

Reproducibility and run provenance The Environment A–B server rerun stores run-time source hashes, configuration digests, the actual LSF launcher, compute-node package metadata, per-run

records, timing arrays, estimator artifacts, trajectory hashes for Experiment 2, and reference inventories. The Experiment 2 main grid contains 612 completed formal learner runs. Across all Experiment 2 suites, 760 learner runs were saved; calibration, long-budget, subsampling-sensitivity, serial-timing, and regression diagnostics remain separate from the 612-run main statistical grid. Experiment 1 contains 1700 formal original population rollouts plus 170 pilot rollouts; the LSTD/MC-LS observers replay the same 1870 rollout identities and are not additional independent population replicates.

Figures and reports may be regenerated from the preserved raw outputs without generating new population rollouts. Such post-processing is treated separately from the experimental measurements, and its local software environment is not reported as the software environment of the server runs.

Environment C stores frozen configs, source/config hashes, run metadata, completion/audit files, training arrays, numerical references, and validation artifacts for all completed big and small configurations. Source/config hashes certify those files, but the routing package does not store full training population-trajectory hashes and we do not claim pathwise identity where such evidence is absent.

Run completion and numerical-audit status are treated as distinct notions throughout the appendix. A completed run is not relabeled as missing simply because a numerical tolerance check fails, and the detailed numerical audit outcomes are reported with the corresponding experimental results rather than silently discarded.

F.2 Trace and Parameter Convergence of LSTD(λ) and MC-LS

All figures below use the population size N on the horizontal axis. The vertical axis reports either the compressed–private trace/credit-mass discrepancy or the compressed–private parameter discrepancy. Dotted reference lines scale as $N^{-1/2}$.

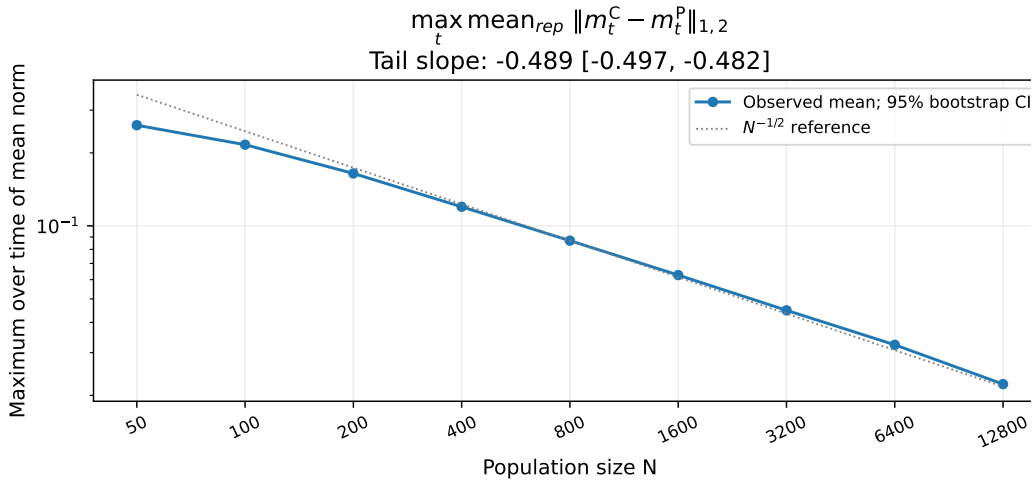

Figure 3: Beach-Bar LSTD(λ) trace convergence, validating Theorem 6. The horizontal axis is the population size N; the vertical axis is the maximum-over-time mean compressed–private credit-mass discrepancy. The dotted line is an $N^{-1/2}$ reference.

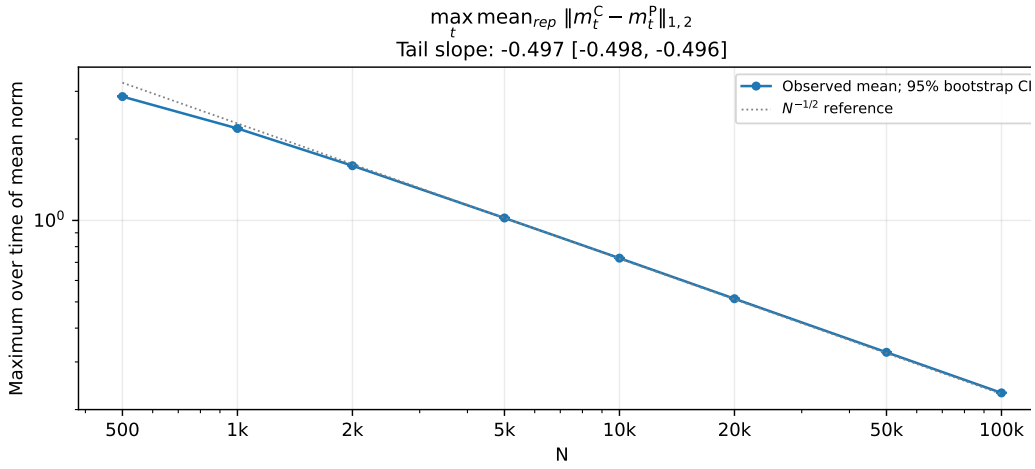

Figure 4: Garnet LSTD(λ) trace convergence, validating Theorem 6. The horizontal axis is the population size N; the vertical axis is the maximum-over-time mean compressed–private credit-mass discrepancy. The dotted line is an $N^{-1/2}$ reference.

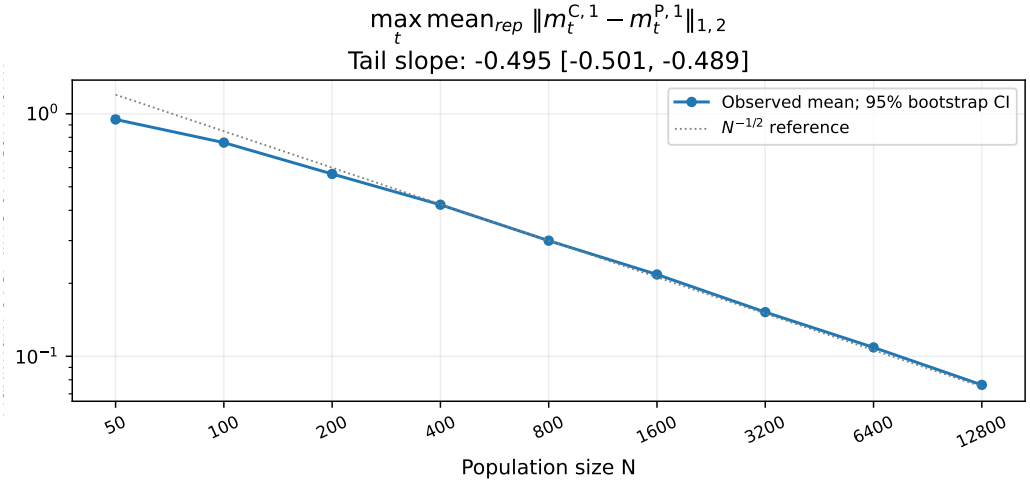

Figure 5: Beach-Bar MC-LS trace convergence at $\lambda = 1$, validating the $\lambda = 1$ specialization of Theorem 6. The horizontal axis is the population size N; the vertical axis is the maximum-over-time mean compressed–private credit-mass discrepancy. The dotted line is an $N^{-1/2}$ reference.

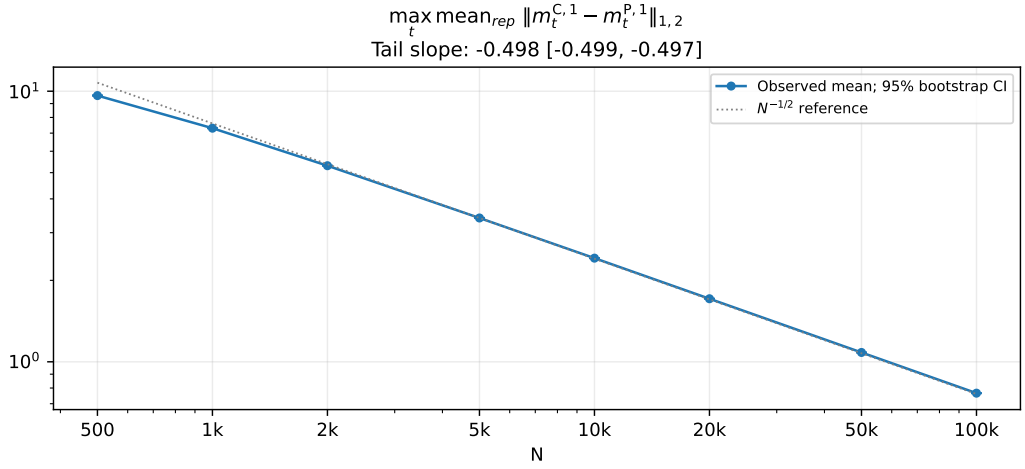

Figure 6: Garnet MC-LS trace convergence at $\lambda = 1$, validating the $\lambda = 1$ specialization of Theorem 6. The horizontal axis is the population size N; the vertical axis is the maximum-over-time mean compressed–private credit-mass discrepancy. The dotted line is an $N^{-1/2}$ reference.

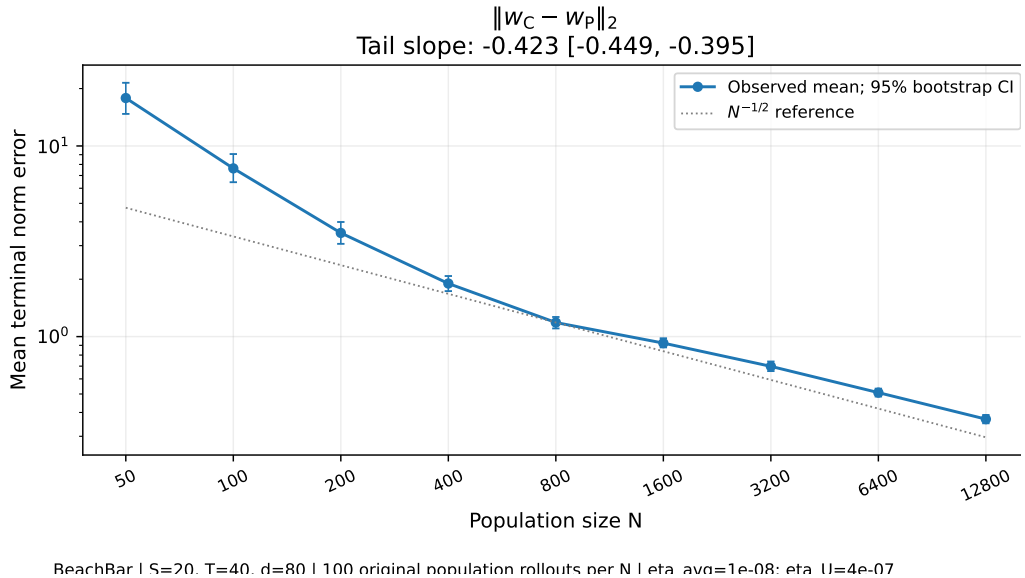

Figure 7: Beach-Bar ridge-LSTD(λ) parameter convergence, illustrating the finite-population consistency in Theorem 7.

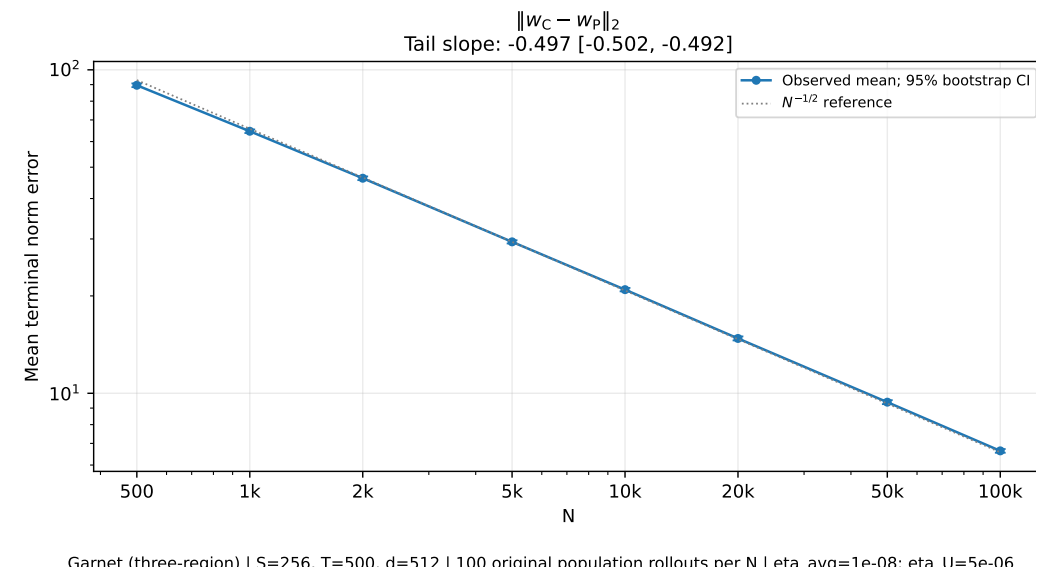

Figure 8: Garnet ridge-LSTD(λ) parameter convergence, validating Theorem 7. The horizontal axis is the population size N; the vertical axis is the mean compressed–private parameter discrepancy $\|w_C - w_P\|_2$. The dotted line is an $N^{-1/2}$ reference.

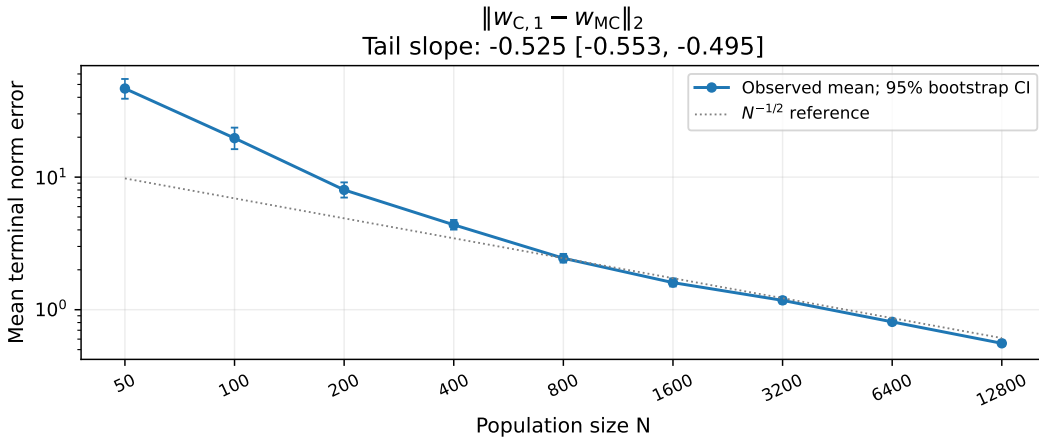

Figure 9: Beach-Bar ridge MC-LS parameter convergence, validating Theorem 8. The horizontal axis is the population size N; the vertical axis is the mean compressed–private parameter discrepancy $\|w_{C,1} - w_{MC}\|_2$. The dotted line is an $N^{-1/2}$ reference.

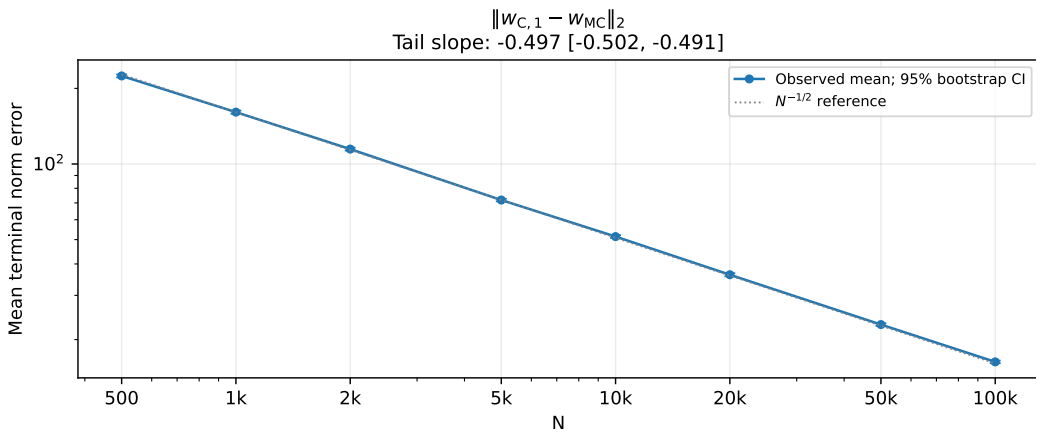

Figure 10: Garnet ridge MC-LS parameter convergence, validating Theorem 8. The horizontal axis is the population size N; the vertical axis is the mean compressed–private parameter discrepancy $\|w_{C,1} - w_{MC}\|_2$. The dotted line is an $N^{-1/2}$ reference.

F.3 Additional Scaling Results

Tables 8–10 report the endpoint results of the S-, T-, and N-scaling experiments. Error is the final relative RMSE. Memory is the learner-owned-array memory in MiB, and time is the amortized processing time per population step in milliseconds. For memory and time, each table reports both the absolute values at the first and last sweep points and their last-to-first ratios.

Table 8: State-space scaling from $S = 64$ to $S = 512$.

Method	Error (%)		Memory (MiB)			Time (ms/step)		
	$S = 64$	$S = 512$	$S = 64$	$S = 512$	Ratio	$S = 64$	$S = 512$	Ratio
TD(0)	27.645	30.187	0.035	0.254	7.224	0.391	2.553	6.528
MC-SGD	26.511	25.601	38.339	38.558	1.006	0.462	1.167	2.526
MC-LS	0.570	0.677	80.164	80.383	1.003	31.464	24.179	0.768
Subsampled MC-LS	1.289	1.463	9.867	10.086	1.022	5.801	8.646	1.490
Model-Based (Tabular)	0.300	0.609	8.581	117.604	13.706	0.310	1.571	5.075
Model-Based (Projected)	0.430	0.539	8.336	115.647	13.873	1.224	3.531	2.885
Private Dutch	6.841	7.554	78.164	78.383	1.003	92.495	47.425	0.513
Subsampled Private Dutch	7.373	8.118	7.867	8.086	1.028	7.106	6.712	0.944
DoubleSWEET	6.844	7.548	0.289	2.258	7.811	4.026	16.573	4.117
DoubleSWEET-MC-LS	0.429	0.534	2.289	4.258	1.860	9.446	14.765	1.563
LSTD(λ)	0.457	0.562	80.164	80.383	1.003	41.206	34.252	0.831
DoubleSWEET-LSTD(λ)	0.429	0.534	2.289	4.258	1.860	7.983	23.829	2.985

Table 9: Horizon scaling from $T = 100$ to $T = 2000$.

Method	Error (%)		Memory (MiB)			Time (ms/step)		
	$T = 100$	$T = 2000$	$T = 100$	$T = 2000$	Ratio	$T = 100$	$T = 2000$	Ratio
TD(0)	32.275	62.764	0.129	0.129	1.000	1.317	1.269	0.963
MC-SGD	28.315	52.148	7.912	152.885	19.323	0.876	0.773	0.883
MC-LS	2.749	0.229	80.258	80.258	1.000	30.825	24.979	0.810
Subsampled MC-LS	5.855	0.500	9.961	9.961	1.000	9.873	5.920	0.600
Model-Based (Tabular)	1.902	0.124	18.169	360.959	19.866	1.653	1.529	0.925
Model-Based (Projected)	2.307	0.195	17.972	357.050	19.867	4.006	3.098	0.773
Private Dutch	28.154	36.158	78.258	78.258	1.000	79.085	40.650	0.514
Subsampled Private Dutch	29.010	35.887	7.961	7.961	1.000	9.136	11.414	1.249
DoubleSWEET	28.147	36.162	1.133	1.133	1.000	10.400	10.131	0.974
DoubleSWEET-MC-LS	2.311	0.193	3.133	3.133	1.000	8.397	9.404	1.120
LSTD(λ)	2.564	0.196	80.258	80.258	1.000	31.858	28.850	0.906
DoubleSWEET-LSTD(λ)	2.311	0.195	3.133	3.133	1.000	15.058	12.955	0.860

Table 10: Population-size scaling from $N = 5{,}000$ to $N = 100{,}000$.

Method	Error (%)		Memory (MiB)			Time (ms/step)		
	$N = 5{,}000$	$N = 100{,}000$	$N = 5{,}000$	$N = 100{,}000$	Ratio	$N = 5{,}000$	$N = 100{,}000$	Ratio
TD(0)	26.976	26.844	0.129	0.129	1.000	0.710	2.117	2.980
MC-SGD	25.982	26.127	9.708	191.631	19.740	0.575	1.775	3.086
MC-LS	0.956	0.508	21.664	392.758	18.129	10.315	78.093	7.571
Subsampled MC-LS	1.438	1.537	9.961	9.961	1.000	5.191	6.562	1.264
Model-Based (Tabular)	0.818	0.181	31.959	128.061	4.007	0.812	2.892	3.560
Model-Based (Projected)	0.504	0.472	30.980	127.082	4.102	3.095	6.466	2.089
Private Dutch	6.875	6.779	19.664	390.758	19.871	16.525	148.005	8.957
Subsampled Private Dutch	7.063	6.814	7.961	7.961	1.000	8.710	5.663	0.650
DoubleSWEET	6.872	6.779	1.133	1.133	1.000	5.062	21.087	4.165
DoubleSWEET-MC-LS	0.490	0.472	3.133	3.133	1.000	5.836	26.451	4.533
LSTD(λ)	0.597	0.480	21.664	392.758	18.129	9.374	102.493	10.933
DoubleSWEET-LSTD(λ)	0.492	0.474	3.133	3.133	1.000	8.418	31.458	3.737

F.4 MFG-Solving Results with Forward-Learned Critics

Environment C tests whether critics learned during forward population evolution can be used inside an outer policy-improvement loop. We report the large-population extended configuration with $N = 20{,}000$, 200 outer policy updates, and 80 critic-training episodes per update. Private Dutch and DoubleSWEET are each evaluated over five independent training seeds. The numerical reference used below is an independently validated finite-population candidate and is not claimed to be an exact or unique Nash equilibrium. The policy-distance metric and the finite-population unilateral-deviation gap are defined in Sec. F.1.

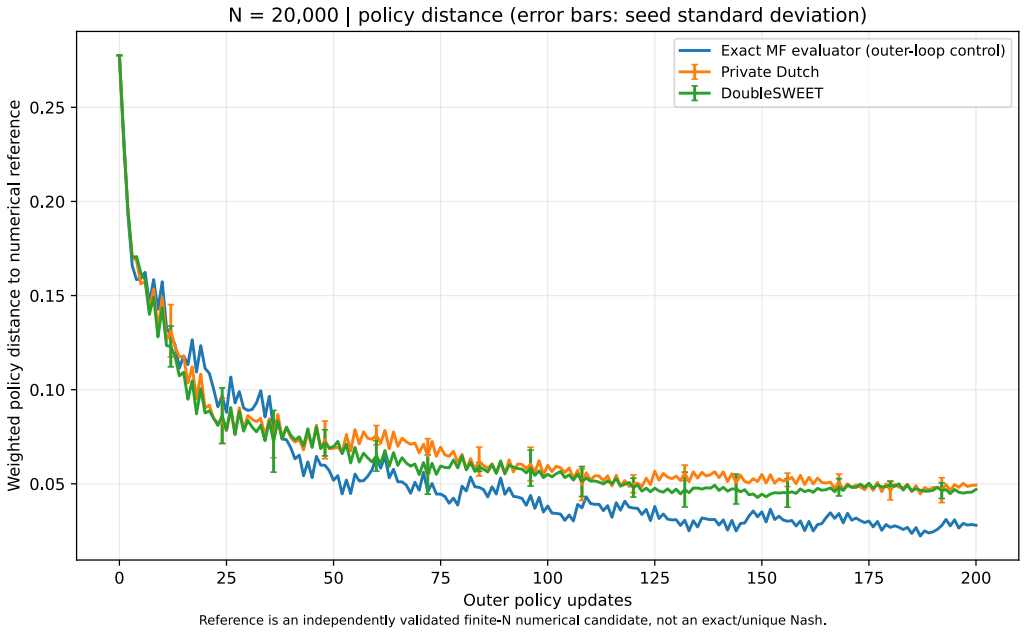

Figure 11: Policy learning with forward-learned critics at $N = 20{,}000$. The horizontal axis is the number of outer policy updates and the vertical axis is the weighted policy distance D to the numerical finite-population reference. Error bars for Private Dutch and DoubleSWEET are seed standard deviations. The model-based exact mean-field evaluator is included as an outer-loop control.

Figure 11 shows that both trace-based learners substantially reduce the policy distance during the early outer-loop updates and then stabilize near the numerical reference. At the final update, the mean policy distance is 0.04935 ± 0.00592 for Private Dutch and 0.04701 ± 0.00329 for DoubleSWEET, while the exact mean-field evaluator reaches 0.02795. Thus, the forward-learned critics provide a sufficiently informative policy-evaluation signal for repeated policy improvement, and the state-wise compression in DoubleSWEET does not visibly degrade the final policy-learning outcome relative to Private Dutch. The residual distance should be interpreted only relative to the chosen numerical reference, rather than as a certificate of the exact Nash gap.

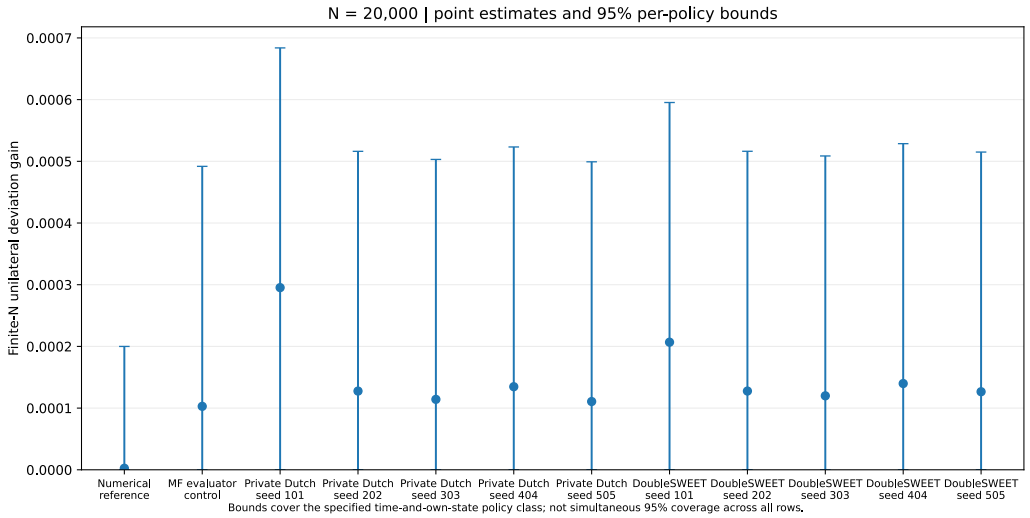

Figure 12: Finite-population unilateral-deviation gaps of the final policies at $N = 20{,}000$. Points are gap estimates in the time–own-state deviation class Π_{TS}, and vertical bars are the corresponding 95% per-policy bounds. These bounds apply to each evaluated policy separately and are not simultaneous confidence bounds across all rows.

Figure 12 directly evaluates whether a tagged agent can profit from a unilateral deviation within the restricted class Π_{TS}. Averaged across the five learning seeds, Private Dutch has a finite-N gap estimate of 1.56×10^{-4} with a mean 95% upper bound of 5.45×10^{-4}, while DoubleSWEET has a gap estimate of 1.44×10^{-4} with a mean upper bound of 5.33×10^{-4}. The exact mean-field evaluator control has a gap estimate of 1.03×10^{-4} and an upper bound of 4.92×10^{-4}. The small evaluated gaps indicate that the policies learned from forward-trained critics admit little profitable unilateral deviation within the tested policy class, and DoubleSWEET remains comparable to its uncompressed private-trace counterpart after being embedded in the complete MFG policy-improvement loop.

Table 11: Final MFG-solving results for the large-population routing experiment at $N = 20{,}000$. For Private Dutch and DoubleSWEET, policy distance is reported as mean $\pm$ seed standard deviation over five independent runs; the gap columns report the across-seed means of the per-policy finite-N estimates and upper bounds.

Method	Final policy distance	Finite-N gap estimate	95% upper bound
Exact MF evaluator	0.02795	1.03×10^{-4}	4.92×10^{-4}
Private Dutch	0.04935 ± 0.00592	1.56×10^{-4}	5.45×10^{-4}
DoubleSWEET	0.04701 ± 0.00329	1.44×10^{-4}	5.33×10^{-4}