LAPLACIAN FLOWS FOR POLICY LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern AI systems improve task performance through local policy updates, yet gains on the current training objective can come at the expense of previously acquired behaviours, a risk that conservative single-step updates alone do not eliminate. To guide adaptation with a reference to accumulated experience, we draw on geometric measure theory and model historical policies as samples from an experience-induced, locally low-dimensional manifold in Wasserstein space. We introduce Policy Laplacian Trace (PLT), an evolving trajectory map that uses geometric relations among relevant historical policies to regularise subsequent updates. Unlike Wasserstein constraints imposed only between successive iterates, PLT relates each candidate update to multiple historical policies. A history-induced graph Dirichlet energy translates these geometric relations into a Laplacian-type correction to the task-driven update, bringing trajectory-level geometric context into local optimisation. Learning shapes the map, while its evolving graph geometry guides subsequent policy updates. We instantiate PLT in reinforcement learning and language-model training, using a scalable semantic-transport proxy for the latter. Experiments with PPO and MAPPO on Atari, MuJoCo, and StarCraft II show improvements in final performance and sample efficiency. Controlled language-model stress tests show gains in robustness to contradictory evidence, long-range factual recall, and few-shot generalisation. On the recall task, Qwen3-4B improves weighted token accuracy by 4.83%, with a 3.56% increase in median training-step time under the tested configuration. This feedback between policy evolution and its geometric map suggests a potential role for experience-induced geometry in recursive learning.

1 INTRODUCTION

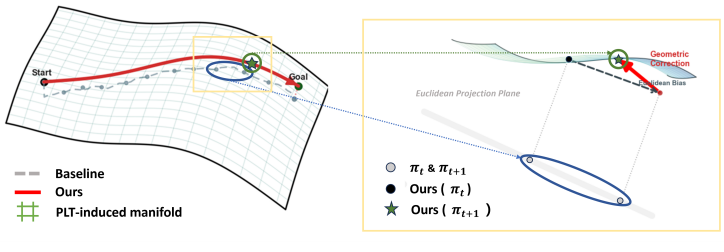

Figure 1: An illustration on how policies change over the course of training in a Wasserstein manifold. Policies lie on an (idealized) low-dimensional Wasserstein policy manifold, with the gray dot denoting the current policy π_t. A standard update computed in a Euclidean proxy space (parameter/KL–Fisher geometry) follows the dashed direction, introducing an *Euclidean bias* that can drift off the manifold and yield misaligned transport-scale changes. PLT adds an experience-induced geometric correction (red) to realign the step with the manifold, producing π_{t+1} (green star). Dashed vertical projections emphasize that small Euclidean/KL-constrained steps may correspond to large W_2 displacements. **Left:** top-down view of update directions. **Right:** side view of off-manifold drift and correction.

Much of modern machine learning improves predictions and decisions through iterative parameter updates. In reinforcement-learning agents and language models (Zhao et al., 2023; Wang et al., 2020), these updates also reshape conditional distributions over actions or tokens. We represent

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.

such behaviour by a policy $\pi_\theta(\cdot \mid x) \in \mathcal{P}(\mathcal{A})$, where x denotes a state, context, or prompt and $\mathcal{A}$ the action space (Doob, 1941; Ash & Doléans-Dade, 2000). For fixed x, training therefore traces a behavioural trajectory $\{\pi_{\theta_t}(\cdot \mid x)\}_t$, not merely a sequence of parameter vectors. However, constraining individual updates does not, by itself, prevent cumulative behavioural drift. This motivates a historical reference for learning: can past policies be organised into an *evolving geometric map* that guides subsequent updates? To quantify relations between these behavioural landmarks, we equip $\mathcal{A}$ with a task-semantic ground metric and compare policy distributions with finite second moments using the induced 2-Wasserstein distance W_2 on $\mathcal{P}_2(\mathcal{A})$ (Villani et al., 2008).

Constructing such a map requires understanding the geometric structure of policy history. Inspired by geometric measure theory and the interplay between directional organisation and concentration in Kakeya-type problems (Wang & Zahl, 2025), we propose an *experience-induced policy manifold hypothesis*: locally coherent policy histories can be modelled as samples from a locally low-dimensional manifold in Wasserstein space.[1] Appendix F.6 provides empirical motivation: in the reported setting, policy snapshots have an estimated effective dimension of approximately 46 versus roughly 10^6 model parameters, yielding a parameter-count-to-effective-dimension ratio exceeding 2×10^4.

This policy-centred view differs from the data manifold hypothesis (Bengio et al., 2013; Fefferman et al., 2016; Narayanan & Niyogi, 2009): its geometric samples are the learner's own policies, rather than observations. Wasserstein gradient flows and Laplacian learning motivate regularisation in this distributional geometry (Ambrosio et al., 2005; Chow & Gangbo, 2019; Oliver et al., 2025). Our approach complements policy-gradient, actor–critic and value-based methods (Sutton et al., 1999; Konda & Tsitsiklis, 1999; Gu et al., 2024), decision transformers (Chen et al., 2021), and proximal or constrained updates (Schulman et al., 2017; Achiam et al., 2017), by making historical geometry an explicit reference for ongoing learning.

To turn this geometric reference into an executable update, we introduce **Policy Laplacian Trace (PLT)** (Fig. 1). PLT retrieves historical conditional policies from comparable states or contexts to construct local experience graphs. A variational step couples a Wasserstein graph Dirichlet energy with a KL-proximal anchor to the task-driven update, yielding Laplacian-type corrections informed by multiple historical iterates. Learning shapes the map, while its evolving graph topology guides subsequent policy updates. We instantiate PLT in reinforcement learning and language modelling, using a scalable semantic-transport proxy for the latter. Experiments with PPO and MAPPO show gains in sample efficiency and final performance, while controlled language-model stress tests show improvements in robustness to contradictory evidence, long-range factual recall and few-shot adaptation. Our contributions are:

- **A trajectory-map view of policy learning.** We formulate an experience-induced policy manifold hypothesis, treating historical policies as geometric landmarks for subsequent learning.
- **Policy Laplacian Trace.** We develop a history-conditioned operator combining Wasserstein graph regularisation with KL-proximal anchoring.
- **Theory and empirical evidence.** We analyse graph Dirichlet energies, their p-Dirichlet extension, and Laplacian-type and proximal interpretations. Experiments show smoother trajectories, improved RL sample efficiency, and gains in robustness, long-range recall and few-shot adaptation on controlled language-model tasks.

2 PRELIMINARIES

This section collects the optimal-transport preliminaries used throughout. We treat a policy as a conditional measure $\pi_\theta(\cdot \mid x) \in \mathcal{P}_2(A)$ and compare policies via the 2-Wasserstein geometry. Appendices A and C provide related work and notation, respectively.

Hypothesis 1 (Experience-induced Wasserstein policy geometry is learnable and behavior-relevant)**.** *Fix x and equip $(\mathcal{A}, d_{\mathcal{A}})$ with a task-semantic metric; let $\Lambda_x := \{\pi_\theta(\cdot \mid x) : \theta \in \Theta\} \subset \mathcal{P}_2(\mathcal{A})$ and denote by $\widetilde{\Lambda}_x \subset \Lambda_x$ the set of policies visited during training. Assume $\widetilde{\Lambda}_x$ lies on a locally d_x-dimensional ($d_x \ll \dim \Theta$) bi-Lipschitz chart under $(\|\cdot\|_2, W_2)$ and is sampled densely enough*

[1] This conceptually echoes geometric reduction (Crooks & Mayrand, 2022), but infers geometry from experience rather than explicit symmetries.

along $\{\pi_{\theta_t}(\cdot \mid x)\}_t$ to consistently estimate W_2-kNN neighborhoods and graph-Laplacians from historical policies. Moreover, rewards are Lipschitz in action, $|r(x,a) - r(x,a')| \le L_r d_{\mathcal{A}}(a,a')$ (optionally $W_1(P(\cdot \mid x,a), P(\cdot \mid x,a')) \le L_P d_{\mathcal{A}}(a,a')$), so for any π, π',

$$\left|\mathbb{E}_{a\sim\pi(\cdot|x)}[r(x,a)] - \mathbb{E}_{a\sim\pi'(\cdot|x)}[r(x,a)]\right| \le L_r\, W_2\big(\pi(\cdot \mid x), \pi'(\cdot \mid x)\big). \tag{1}$$

Appendix F.6 reports diagnostics of low intrinsic dimension/spectral decay in policy snapshots and shows that +PLT reduces W_2 drift at comparable KL, supporting the geometry-mismatch motivation and Hypothesis 1.

Let (A, d_A) be a Polish metric space, where $A \subset \mathbb{R}^m$ or a finite action set equipped with a cost matrix. We write $\mathcal{P}_2(A)$ for the set of probability measures on A with finite second moment, and $\mathcal{B}(A)$ for the Borel σ-algebra. Given a measurable map $T : A \to A$ and a measure $\mu \in \mathcal{P}_2(A)$, the pushforward measure $T_{\#}\mu$ is defined by

$$\int_A \varphi(a)\, \mathrm{d}(T_{\#}\mu)(a) = \int_A \varphi(T(a))\, \mathrm{d}\mu(a), \quad \forall \varphi \in L^\infty(A). \tag{2}$$

For two measures $\mu, \nu \in \mathcal{P}_2(A)$, let $\Pi(\mu,\nu)$ denote the set of couplings on $A \times A$ with marginals (μ,ν). The 2-Wasserstein distance is

$$W_2^2(\mu,\nu) := \inf_{\pi\in\Pi(\mu,\nu)} \int_{A\times A} d_A(a,a')^2\, \mathrm{d}\pi(a,a'). \tag{3}$$

In computation, we use an entropic-regularized variant:

$$W_{2,\varepsilon}^2(\mu,\nu) := \min_{\pi\in\Pi(\mu,\nu)} \int d_A(a,a')^2\, \mathrm{d}\pi(a,a') + \varepsilon\, \mathrm{KL}(\pi \,\|\, \mu\otimes\nu), \tag{4}$$

which admits efficient Sinkhorn-type solvers in discrete settings and stable sample-based approximations for $\varepsilon > 0$.

Finally, any measures $\{\mu_i\}_{i=1}^k \subset \mathcal{P}_2(A)$ and weights $w \in \Delta^{k-1}$, an (entropic) Wasserstein barycenter is defined by

$$\mathrm{Bar}_\varepsilon(\{\mu_i, w_i\}) \in \arg\min_{\mu\in\mathcal{P}_2(A)} \sum_{i=1}^k w_i\, W_{2,\varepsilon}^2(\mu,\mu_i). \tag{5}$$

3 Policy Learning on Experience-Induced Wasserstein Manifolds

Regardless of the learning paradigm (e.g., reinforcement learning or language modeling), we view policy learning as the evolution of conditional distributions,

$$x \;\mapsto\; \pi_\theta(\cdot \mid x) \;\in\; \mathcal{P}_2(\mathcal{A}). \tag{6}$$

At iteration t, standard policy optimization performs a myopic one-step update

$$\pi_{\theta_{t+1}} = \mathcal{U}_t(\pi_{\theta_t}; \mathcal{T}), \tag{7}$$

where $\mathcal{T}$ denotes a task-dependent operator, such as Bellman evaluation or a policy-gradient objective. Although parameter updates $\theta_{t+1} = \theta_t - \eta\nabla_\theta L_t$ treat each policy π_{θ_t} in isolation, the induced sequence $\{\pi_{\theta_t}\}_t$ traces a low-dimensional realizable manifold embedded in Wasserstein space. Ignoring this latent geometric structure, purely task-driven updates can induce distributionally jagged policy variations across neighboring states, thereby degrading stability and generalization. We propose to regularize policy learning by its own evolution: each update couples the task objective with Laplacian smoothing in distribution space, enforcing consistency with geometrically nearby historical policies. This leads to the *Policy Laplacian Trace* (PLT) framework, whose components and learning procedure are detailed in the following sections.

3.1 POLICY LAPLACIAN TRACE

To incorporate potentially valuable geometric information embedded in the policy manifold into conventional updates, we construct *Policy Laplacian Trace* (PLT).

Definition 3.1 (Policy Laplacian Trace)**.** At optimisation iteration t, we log an evolution sample

$$\mathcal{E}_t = (s_t, \pi_t, \xi_t), \tag{8}$$

where s_t is a visited state (or context), $\pi_t := \pi_{\theta_t}(\cdot \mid s_t) \in \mathcal{P}_2(\mathcal{A})$ is the *conditional policy distribution* at that input, and ξ_t optionally encodes auxiliary local training signals (e.g., a low-dimensional summary of the advantage or gradient direction). The Policy Laplacian Trace (PLT) at step t is the collection

$$\mathcal{H}_t = \{\mathcal{E}_i\}_{i \le t}, \tag{9}$$

maintained with a fixed capacity (e.g., FIFO or similarity-based prioritization). Crucially, PLT regularizes in the *output* Wasserstein geometry, not in state space: the input s is used only to retrieve comparable experience (define neighborhoods), while geometry is measured between the retrieved *policy distributions* via W_2. Thus $\mathcal{H}_t$ is more than a replay buffer, it provides geometric landmarks in $(\mathcal{P}_2(\mathcal{A}), W_2)$ for policies conditioned on nearby inputs, inducing a local, time-varying geometry for the map $s \mapsto \pi(\cdot \mid s)$ and enabling Laplacian-style regularization in distribution space.

A standard policy optimisation (Eq.7) with PLT is modified through a fusion operator $\mathcal{M}_t$:

$$\pi_{\theta_{t+1}} = \mathcal{M}_t\{\mathcal{U}_t(\pi_{\theta_t}; \mathcal{T});\ \mathcal{H}_t\}. \tag{10}$$

3.2 POLICY LAPLACIAN LEARNING FROM EXPERIENCE

At each iteration, the PLT serves solely as a geometric operator: it is queried to construct a local policy graph, which in turn induces a Wasserstein–Dirichlet energy used to regularize the next update in distribution space. The specific three steps will be presented in following section.

3.2.1 POLICY LAPLACIAN CONSTRUCTION

Given the current input/state s_t, an s_t-centered local experience graph could be constructed. We first retrieve similar past states s_i from PLT $\mathcal{H}_t$ and select the top-k neighbors to form the set $\mathcal{N}_k(s_t)$. The edge weights w in the graph are then obtained by locally normalizing the affinities of these neighbors:

$$w_{tj} = \frac{\kappa(s_t, s_j)}{\sum_{i \in \mathcal{N}_k(s_t)} \kappa(s_t, s_i)}, \qquad j \in \mathcal{N}_k(s_t). \tag{11}$$

where κ is a similarity kernel $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$, which could be instantiated with any task-specific similarity measure, for example:

$$\kappa(s, s') = \exp\Big(-\frac{\|\phi(s) - \phi(s')\|^2}{\sigma^2}\Big) \text{ (RBF)} \qquad \text{or} \qquad \kappa(s, s') = \exp\Big(\frac{\phi(s)^\top \psi(s')}{\sqrt{d}}\Big) \text{ (Attention)} \tag{12}$$

The s_t-centered graph is sparse, with a topology induced by experience similarity and edge weights that quantify neighbor relevance. By construction, the weights are nonnegative and sum to one (defining a convex combination), so that higher-similarity neighbors receive larger weights. Symmetry depends on the kernel: distance-based kernels typically produce symmetric graphs, whereas attention-based affinities are generally asymmetric, allowing for context-dependent matching.

3.2.2 WASSERSTEIN-DIRICHLET FUSION ENERGY

Building on the construction in the previous Section 3.2.1 and Eq.4, to bridge the geometric view with an executable update, we define a local memory potential at iteration t by

$$\Phi_t(\pi_\theta) \ := \ \frac{1}{2} \sum_{j \in \mathcal{N}(s_t)} w_{tj}\, W_{2,\varepsilon}^2\big(\pi_\theta(\cdot \mid s_t), \pi_\theta(\cdot \mid s_j)\big), \tag{13}$$

Geometrically, Φ_t acts as a local Wasserstein-Dirichlet energy on the neighborhood graph: its (Wasserstein) first variation induces a graph Laplacian force in $\mathcal{P}_2(\mathcal{A})$ that pulls the candidate policy

toward a weighted barycentric configuration of nearby policies, thereby implementing Laplacian type smoothing directly in distribution space. Formally, the corresponding Wasserstein subgradient $\nabla_{W_2}\Phi_t(\pi)$ takes the form of a weighted aggregation of logarithmic transport directions from π_θ to its neighbors (Lemma H.5; Remark H.6) and is defined as:

$$\nabla_{W_2}\Phi_t(\pi_\theta) \;\propto\; -\sum_{j\in\mathcal{N}(s_t)} w_{tj}\,\mathrm{Log}_{\pi_\theta}(\pi_\theta(\cdot\mid s_j)), \tag{14}$$

where $\mathrm{Log}_{\pi_\theta}(\pi_\theta(\cdot\mid s_j)$ denotes the (entropic) Wasserstein log direction from $\pi_\theta(\cdot\mid s_t)$ to $\pi_\theta(\cdot\mid s_j)$. We then instantiate the abstract fusion operator $\mathcal{M}_t$ in Eq.10 by solving the proximal objective

$$\pi_{\theta_{t+1}} \;=\; \mathcal{M}_t\{\arg\min_{\pi\in\mathcal{P}_2(\mathcal{A})} \langle \hat{g}_t, \pi_{\theta_t}\rangle \;,\; \lambda\,\Phi_t(\pi_\theta)\}, \tag{15}$$

where λ weights the geometry regularizer and $\hat{g}_t$ denotes a task-induced linear functional on $\mathcal{P}_2(\mathcal{A})$, which can be equivalently viewed as the myopic update $\mathcal{U}_t(\pi_\theta)$ in Eq. 7 (e.g., in RL, an empirical estimate of $-A_t(\cdot\mid s_t)$, so that $\langle \hat{g}_t, \pi_{\theta_t}\rangle$ recovers the standard policy-gradient surrogate).

The fusion process in Eq. 15 can be viewed as an implicit (proximal) time-discretization of a composite geometric energy, yielding existence/uniqueness and an energy descent guarantee under mild convexity assumptions (Lemma H.8). See Fig. 1 for a visual intuition of the geometry mismatch and PLT's geometric correction. The composite objective couples two geometrically distinct terms: Φ_t provides Laplacian smoothing in W_2, while KL bounds displacement from the task-leading policy: W_2 governs direction, KL governs step size. Neither alone suffices: without Φ_t one recovers standard KL-proximal updates and the mismatch of Section 1; without KL the barycentric step can overshoot. Lemma H.8 establishes well-posedness and composite energy descent under a KL trust region, necessary conditions for the empirical stability improvements in Section 5. Appendix D provides additional details on efficient implementation.

4 PLT Implementation

Building on the previous section, we instantiate the Policy Laplacian Trace (PLT) as an experience-driven operator within concrete algorithms. We consider two canonical realizations of conditional policy learning: Proximal Policy Optimization (PPO) in reinforcement learning, and token-level policy learning in large language models on simple tasks. These settings are selected for clarity rather than generality. PLT is a framework-level construct that naturally generalizes to broader learning paradigms, enabling principled and systematic reuse of experience across domains. Pseudocode implementations are provided in Appendix E.

4.1 RL/MDP Instantiation and Implementation Details

We implement PLT Eq. 15 in an *amortized* manner: for each state s sampled from the rollout buffer, we compute a conservative reference policy $\pi_{\mathrm{align}}(\cdot\mid s)$ as a tractable approximation to the local proximal solution, and use it as an anchor in the PPO surrogate objective (Sutton et al., 1998; Schulman et al., 2017). Given a MDP with PPO algorithm, an on-policy rollout buffer $\mathcal{B}$ of horizon T under the behavior policy $\pi_{\theta_{\mathrm{old}}}$ is defined as

$$\mathcal{B} = \Big\{(s_t, a_t, r_t, s_{t+1}, \log\pi_{\theta_{\mathrm{old}}}(a_t\mid s_t), \widehat{A}_t)\Big\}_{t=1}^{T}, \tag{16}$$

where s is state, a is action, r is reward and $\widehat{A}_t$ is advantage. The actor objective satisfies

$$\mathcal{L}_{\mathrm{actor}}(\theta) = \mathbb{E}_{(s,a,\hat{A})\sim\mathcal{B}}\Big[\min\Big(\tfrac{\pi_\theta(a_i|s_i)}{\pi_{\theta_{\mathrm{old}}}(a_i|s_i)}\,\hat{A}, \mathrm{clip}\big(\tfrac{\pi_\theta(a_i|s_i)}{\pi_{\theta_{\mathrm{old}}}(a_i|s_i)}, 1-\epsilon, 1+\epsilon\big)\,\hat{A}\Big)\Big] \tag{17}$$

In parallel, we maintain a fixed capacity Policy Laplacian Trace $\mathcal{H}$ that logs state indexed local action distributions at each time step t,

$$\mathcal{H} \leftarrow \mathcal{H}\cup\{(z_t,\varphi_t)\},\; z_t = f_{\mathrm{enc}}(s_t),\; \varphi_t \coloneqq \pi_{\theta_{\mathrm{old}}}(\cdot\mid s_t), \tag{18}$$

where z_t is a retrieval key (state embedding) and φ_t is a lightweight representation of the policy distribution at s_t. For continuous actions, we store diagonal Gaussian parameters $\varphi_t = (\mu_t, \log\sigma_t)$; for discrete actions, we store categorical probabilities or logits $\varphi_t = p_t$.

Given a query state s (e.g., from PPO minibatches), we compute its key $z = f_{\text{enc}}(s)$ using the same encoder as the policy (e.g., the penultimate feature layer), and retrieve the top-k trace indices $I(s) \subset \{1, \dots, |\mathcal{H}|\}$ under Euclidean or cosine distance in key space. Concretely, the Eq.11 is realized by a temperature-controlled normalized kernel for computing the retrieval weights w,

$$w_i(s) = \frac{\exp\big(-\|z - z_i\|^2/\tau\big)}{\sum_{j \in I(s)} \exp(-\|z - z_j\|^2/\tau)}, \sum_{i \in I(s)} w_i(s) = 1, \tag{19}$$

where τ is a temperature. Given neighbors $I(s)$ and weights $\{w_i(s)\}$, we form a state-conditioned historical reference $\pi_{\text{hist}}(\cdot \mid s)$ with $\varphi \in H$: for discrete actions,

$$\pi_{\text{hist}}(\cdot \mid s) \leftarrow \sum_{i \in I(s)} w_i(s)\, p_i, \tag{20}$$

while for continuous actions we use a weighted Gaussian mixture

$$\pi_{\text{hist}}(\cdot \mid s) \leftarrow \sum_{i \in I(s)} w_i(s) \mathcal{N}(\mu_i, \sigma_i^2). \tag{21}$$

We then compute a conservative Wasserstein alignment as a computable KL-regularized proximal application of the local memory potential Φ_t Eq.13 by the objective

$$\pi_{\text{align}}(\cdot \mid s) \in \arg\min_{\pi \in \mathcal{P}_2(\mathcal{A})} \Big\{ W_{2,\varepsilon}^2\big(\pi, \pi_{\text{hist}}(\cdot \mid s)\big) + \rho\, \text{KL}\big(\pi \| \pi_\theta(\cdot \mid s)\big) \Big\} \tag{22}$$

About Eq 22 design, we have make an ablation study for why we choose this deisgn in the Appendix F.4. approximated via sample-based entropic OT: draw $A = \{a_i\}_{i=1}^m \sim \pi_\theta(\cdot \mid s)$ and $B = \{b_j\}_{j=1}^m \sim \pi_{\text{hist}}(\cdot \mid s)$, set $C_{ij} = \|a_i - b_j\|_2^2$, compute the Sinkhorn plan [2],

$$P^\star \in \arg\min_{P \in \mathcal{U}(u,v)} \langle C, P \rangle + \varepsilon H(P), \qquad u = v = \tfrac{1}{m}\mathbf{1}, \tag{23}$$

and take barycentric projections $\tilde{a}_i = \sum_j \big(P_{ij}^\star / \sum_{j'} P_{ij'}^\star\big) b_j$ followed by a conservative step

$$a_i^{\text{align}} = (1 - \alpha) a_i + \alpha \tilde{a}_i, \quad \alpha = \alpha(\rho) \ \ (\text{e.g. } \alpha = \tfrac{1}{1+\rho}), \tag{24}$$

after which $\pi_{\text{align}}(\cdot \mid s)$ is obtained by refitting the policy family to $\{a_i^{\text{align}}\}_{i=1}^m$ (moment matching for Gaussians; direct entropic OT on categorical support for discrete actions).

We incorporate π_{align} into PPO as a conservative, geometry-aware anchor by optimizing the unified actor objective

$$\mathcal{L}_{\text{actor}}(\theta) = \mathbb{E}_{(s,a,\hat{A}) \sim \mathcal{B}} \Big[\min\Big(\tfrac{\pi_{align}(a_i|s_i)}{\pi_{\theta_{\text{old}}}(a_i|s_i)}\, \hat{A}, \text{clip}\big(\tfrac{\pi_{align}(a_i|s_i)}{\pi_{\theta_{\text{old}}}(a_i|s_i)}, 1-\epsilon, 1+\epsilon\big)\, \hat{A} \Big) \Big] \tag{25}$$

In all cases, π_{align} is computed from π_θ in Eq.17 and the PLT $\mathcal{H}$ and is held fixed within each PPO update epoch, consistent with PPO's fixed-behavior-policy protocol.

To amortize OT cost, we precompute and cache $\pi_{\text{align}}(\cdot \mid s)$ for all states in the rollout buffer $\mathcal{B}$ once per PPO iteration (or only for a subset of high-importance states, e.g., the top-quantile of $|\widehat{A}|$). When the trace is not yet sufficiently populated (early training) or retrieval is unreliable, we fall back to the identity anchor $\pi_{\text{align}} = \pi_{\theta_{\text{old}}}$. Per queried state, the dominant overhead is Sinkhorn on an $m \times m$ cost matrix, i.e., $\mathcal{O}(m^2 I)$ for I Sinkhorn iterations, with small m (e.g., 8–32) sufficient in practice. Trace storage scales as $\mathcal{O}\big(M(d_z + d_\varphi)\big)$, where d_z is the key dimension and d_φ is the distribution-parameter dimension (Gaussian: $d_\varphi = 2d_a$; categorical: $d_\varphi = |\mathcal{A}|$).

4.2 LLM Instantiation and Implementation Details

We view autoregressive LLM training as a policy-learning problem on a discrete action space given by the vocabulary $\mathcal{V}$.

[2]The cost is modest: m is small (8–32), the Sinkhorn kernel is precomputed once, and only 1–3 iterations are needed per fusion step. See Appendix G for a full overhead analysis.

At training iteration n, we do not freeze the model parameters. For a prefix $x_{\le t} \in \mathcal{V}^t$ at position t, with vocabulary $\mathcal{V} = \{1, \dots, V\}$, the model parameterized by θ_n outputs a next-token distribution on the probability simplex $\Delta^{V-1} \subset \mathbb{R}^V$:

$$p_{n,t}^{\text{out}} \coloneqq p_{\theta_n}(\cdot \mid x_{\le t}) \in \Delta^{V-1}, \quad h_{n,t} \coloneqq f_{\theta_n}(x_{\le t}) \in \mathbb{R}^{d_h}, \quad \Delta^{V-1} \coloneqq \Big\{ p \in \mathbb{R}^V_{\ge 0} : \sum_{v=1}^{V} p(v) = 1 \Big\} \tag{26}$$

where $h_{n,t}$ denotes the contextual representation of the prefix (e.g., the Transformer hidden state).

To update θ_n, the model is supervised with the ground-truth next token y_t by minimizing the cross-entropy loss $\mathcal{L}_{\text{CE}}$:

$$\mathcal{L}_{\text{CE}}^{(n,t)} = -\log p_{n,t}^{\text{out}}(y_t). \tag{27}$$

4.2.1 Policy Laplacian Trace in LLM

To implement Policy Laplacian Track (PLT), we construct position-conditioned memories $\mathcal{M}_\ell$:

$$\mathcal{M}_\ell \coloneqq \{\mathcal{E}_{n,\ell}\}_{n=1}^{N_{\text{train}}}, \qquad \ell = 1, \dots, L_{\max}, \tag{28}$$

where ℓ indexes the token position and coincides with the current step t when retrieving memories and the PLT $\mathcal{E}_{n,\ell=t}$ is defined as:

$$\mathcal{E}_{n,t} \coloneqq (h_{n,t},\, p_{n,t}^{\text{out}},\, s_{n,t}), \tag{29}$$

where $s_{n,t}$ is an uncertainty/difficulty signal such as surprisal $s_{n,t} \coloneqq -\log p_{n,t}^{\text{out}}(y_t)$.

To fuse PLT and update model given a query at position $\ell = t$ with $(h_{n,t}, p_{n,t}^{\text{out}})$, we retrieve a top-$k$ neighborhood $N_{n,t} \subseteq \mathcal{M}_\ell$ using a similarity score $\kappa(h_{n,t}, h_i)$ (e.g., dot-product or cosine similarity), with softmax weights:

$$w_i(h_{n,t}) = \frac{\exp(\kappa(h_{n,t}, h_i)/\tau_{\text{ret}})}{\sum_{j \in N_{n,t}} \exp(\kappa(h_{n,t}, h_j)/\tau_{\text{ret}})},\ i \in N_{n,t}, \tag{30}$$

where $\tau_{\text{ret}} > 0$ is a retrieval temperature. The retrieved neighborhood induces a memory distribution:

$$p_{n,t}^{\text{mem}} \coloneqq \sum_{i \in N_{n,t}} w_i(h_{n,t})\, q_i, \tag{31}$$

where each $q_i \in \Delta^{V-1}$ is the next-token distribution stored in memory.

Inner view: Fusion with proximal target and gated one-step operator. Computing the Wasserstein distance W_2 over a large vocabulary is computationally expensive due to the $V \times V$ cost matrix. To address this, we adopt a scalable semantic-transport proxy. Let $E \in \mathbb{R}^{V \times d_e}$ be the token embedding matrix with rows e_a, $a \in \mathcal{V}$. For any distribution $p \in \Delta^{V-1}$, we define the embedding mean

$$\mu(p) \coloneqq p^\top E \in \mathbb{R}^{d_e}, \tag{32}$$

and measure the semantic distance between two distributions p and q as

$$d_{\text{sem}}(p, q) \coloneqq \|\mu(p) - \mu(q)\|_2^2. \tag{33}$$

This reduces the $V \times V$ transport problem to d_e dimensions while preserving the intuition of moving probability mass between semantic clusters.

Therefore PLT defines a proximal target for position t as

$$p_{n,t}^\star \in \arg\min_{q \in \Delta^{V-1}} \big\{ \text{KL}(q \| p_{n,t}^{\text{out}}) + \lambda\, R(q; p_{n,t}^{\text{mem}}) \big\}, \quad R(q; p_{n,t}^{\text{mem}}) = \begin{cases} W_2^2(q, p_{n,t}^{\text{mem}}), & \text{(ideal)} \\ \|\mu(q) - \mu(p_{n,t}^{\text{mem}})\|_2^2, & \text{(scalable)} \end{cases} \tag{34}$$

Here, the PLT target $p_{n,t}^\star$ can be interpreted as a proximal application of a local memory potential analogous to Φ_t Eq.13, where $R(q; p_{n,t}^{\text{mem}})$ measures the deviation from the memory distribution, using either the exact Wasserstein distance or its scalable semantic approximation.

Building on this, the fusion is implemented by a single gated step:

$$p_{n,t}^{\mathrm{PLT}} = (1 - g_{n,t})\, p_{n,t}^{\mathrm{out}} + g_{n,t}\, p_{n,t}^{\mathrm{mem}}, \quad g_{n,t} = \sigma\Big(\mathrm{MLP}(h_{n,t}) - \alpha\, \|\mu(p_{n,t}^{\mathrm{out}}) - \mu(p_{n,t}^{\mathrm{mem}})\|_2^2\Big) \tag{35}$$

where $\mathrm{MLP}(h_{n,t})$ scores contextual reliability and the semantic displacement term suppresses fusion under large distributional shifts. The fusion is a tractable one-step approximation to $p_{n,t}^{\star}$, with $g_{n,t}$ controlling interpolation between the model output and memory-fused distribution.

Outer view: operator-induced targets with hardness gating. At position t, PLT produces the inner-fused distribution $p_{n,t}^{\mathrm{PLT}}$ (e.g., via Eq. 35) and applies it selectively according to token difficulty. With surprisal $s_{n,t} := -\log p_{n,t}^{\mathrm{out}}(y_t)$, define a hardness gate

$$\gamma_{n,t} := \sigma\big(\beta(s_{n,t} - \tau_{\mathrm{hard}})\big) \in [0, 1], \tag{36}$$

where τ_{hard} sets the hardness threshold and β controls its sharpness, so easy tokens ($\gamma_{n,t} \approx 0$) remain unchanged, while hard tokens ($\gamma_{n,t} \approx 1$) trigger memory and geometry informed fusion. The final operator-induced output is

$$p_{n,t}^{\mathrm{final}} := (1 - \gamma_{n,t})\, p_{n,t}^{\mathrm{out}} + \gamma_{n,t}\, p_{n,t}^{\mathrm{PLT}} \in \Delta^{V-1}. \tag{37}$$

Optionally, $\gamma_{n,t}$ (or $s_{n,t}$) can be detached to prevent the model from exploiting the gating during training. The parameters are then trained by minimizing the reformed cross-entropy loss:

$$\mathcal{L}_{\mathrm{CE}}^{(n,t),\mathrm{PLT}} = -\log p_{n,t}^{\mathrm{final}}(y_t), \tag{38}$$

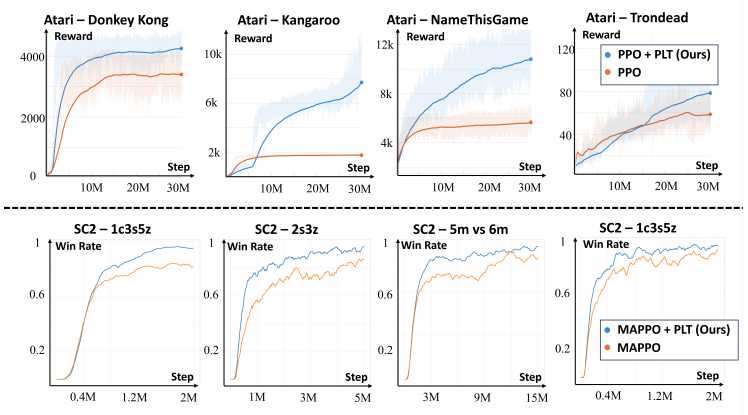

Figure 2: Learning Curves with PLT: PPO Cumulative Returns on Atari (top) and MAPPO Win Rates on StarCraft II (bottom)

Table 1: **PLT improves robustness across three settings.** We report Base and +PLT for each metric. Green indicates improvement over Base: positive relative gains for accuracies, and negative relative drops for losses. Full breakdowns are in Tables 9, 10, and 11.

Model	Param.	(I) Multi-Category Accuracy				(II) Long-Range Factual Recall				(III) Few-Shot New Categories			
		General Acc ↑		Retrieval Acc ↑		Loss (Late) ↓		Token Acc (W) ↑		New Acc ↑		Overall ↑	
		Base	+PLT	Base	+PLT	Base	+PLT	Base	+PLT	Base	+PLT	Base	+PLT
GPT-Nano	2.5M	8.38%	**9.09%**(+8.5%)	8.38%	**10.62%**(+26.7%)	1.58	**1.01**(-36.1%)	56.9%	**59.6%**(+4.7%)	0.00%	0.00%	70.00%	70.00%
GPT-Micro	5M	6.11%	**10.39%**(+70.0%)	7.02%	**11.20%**(+59.5%)	2.03	**1.20**(-40.9%)	48.8%	**51.8%**(+6.1%)	10.78%	**39.22%**(+263.8%)	78.00%	**85.00%**(+9.0%)
GPT-Mini	10M	8.35%	**10.15%**(+21.6%)	9.71%	**11.49%**(+18.3%)	1.35	**1.05**(-22.2%)	58.7%	**59.3%**(+1.0%)	13.86%	**16.83%**(+21.4%)	82.00%	**83.00%**(+1.2%)
Gopher	44M	6.05%	**11.00%**(+81.8%)	7.10%	**12.50%**(+76.1%)	1.84	**1.16**(-37.0%)	54.5%	**58.0%**(+6.4%)	0.97%	**13.59%**(+1301.0%)	85.00%	**90.00%**(+5.9%)
OpenAI-GPT	117M	9.23%	**10.22%**(+10.7%)	10.53%	**11.78%**(+11.9%)	1.80	**1.68**(-6.7%)	58.7%	**60.0%**(+2.2%)	0.00%	**0.99%**(+∞%)	88.00%	**93.00%**(+5.7%)
GPT-2	124M	10.83%	**12.20%**(+12.7%)	12.32%	**13.77%**(+11.8%)	1.71	**1.01**(-40.9%)	56.3%	**59.7%**(+6.0%)	2.13%	**14.89%**(+599.1%)	89.00%	**90.00%**(+1.1%)

5 Experiments and Evaluation

We evaluate Policy Laplacian Trace (PLT) in two representative policy-learning settings: reinforcement learning and large language models. Appendices F.3–F.5 report ablation studies on neighborhood size k, the alignment kernel, and the contribution of PLT's Wasserstein–Laplacian structure. We further extend the Long-Range Factual Recall evaluation to Qwen3-4B, with performance results and computational overhead reported in Appendices F.7 and G.3, respectively. Key hyperparameters and the experimental setup are provided in Appendices F.1 and J, respectively.

5.1 PLT IN RL

The evaluation is conducted on Atari (Mnih et al., 2013), StarCraft II (Samvelyan et al., 2019)and Mujoco (Todorov et al., 2012), by following the Xuance protocol (Liu et al., 2023) with averaged over five random seeds (1, 22, 123, 321, 666). Results are shown in Figure 2, Figure 3 and Table 8 in Appendix F).

Better Sample Efficiency. In the Atari game tasks shown in Figure 2, PPO with PLT achieves higher returns at earlier stages of training and reaches strong performance with fewer environment interaction steps, with the effect being most pronounced on Donkey Kong and Name This Game.

Higher Final Performance. Figure 2 also shows that RL with PLT consistently converges to higher performance across different environments. In particular, on StarCraft II, MAPPO with PLT surpasses the baseline convergence bottleneck in all evaluated scenarios. On Atari, while PPO+PLT can be less sample-efficient early in a few games (e.g., Kangaroo and TronDead), it avoids premature convergence to suboptimal plateaus and later delivers clear gains. Furthermore, as shown in Table 8, PPO+PLT also substantially improves performance on MuJoCo tasks.

5.2 PLT IN LLM

We model LLMs as policies as conditional distribution learners and evaluate whether PLT's experience-induced Wasserstein regularization improves **stability sample efficiency** beyond scaling via three stress tests: **Counterfactual Trust** (fact–rumor/credibility shifts; *General/Retrieval/Conflict/Official* Acc), **Long-Range Factual Recall** (controlled overwrite at known query–evidence distances; targets-only *Loss (Late)*, *Move/Pass Acc*, *Token Acc (W)*, *NLL (Std)*), and **Few-Shot Learning** (novel-class acquisition with base retention; *Base/New/Overall* Acc). Full dataset construction and protocols are in Appendix I.2. We match budgets/protocols between baselines and +PLT and report multi-seed statistics. Because Table 9 uses strict, conflict-heavy multi-way settings with non-trivial chance levels, we emphasize consistent *lift* and cross-scale trends. Under these controls, PLT consistently improves retrieval-conditioned decisions and robustness, indicating more stable long-range evidence integration beyond scaling (Table 1, Appendix I).

Counterfactual Trust Task. Across scales, +PLT improves retrieval-conditioned decisions under counterfactual and credibility-shifted evidence, with the largest gains typically in *Retrieval* accuracy and frequent improvements in *Conflict/Official* accuracy (Table 9). This benchmark provides a controlled proxy for robustness to evidence shifts and contradictory retrieval.

Long-Range Factual Recall. In a controlled overwrite setting with explicit query–evidence distances and distractors, +PLT consistently reduces *Loss (Late)* and improves far-distance recall while maintaining comparable (or better) calibration (*NLL (Std)*; Table 10). These results isolate distance-driven degradation and are consistent with more stable long-range evidence integration.

Few-Shot Learning. Under base-class retention plus novel-class acquisition, +PLT increases *New* accuracy while keeping *Base* accuracy essentially unchanged (Table 11). This indicates improved sample efficiency for few-shot acquisition with reduced forgetting, complementing the retrieval- and long-context stress tests.

6 CONCLUSION

We present Policy Laplacian Trace (PLT), an evolving trajectory map that uses policy history to guide subsequent learning. Inspired by geometric measure theory, we model locally coherent histories as samples from an experience-induced, locally low-dimensional manifold in Wasserstein space. PLT organises relevant historical policies into local graphs, coupling a Wasserstein–Dirichlet potential with KL-proximal anchoring to produce Laplacian-type corrections to task-driven updates. This complements single-step trust regions with trajectory-level geometric regularisation. Experiments show gains in RL sample efficiency and final performance, alongside improvements in robustness, long-range recall and few-shot adaptation in controlled language-model stress tests. Effective regularisation relies on reliable historical neighbourhoods and task-relevant transport metrics. PLT thus connects policy evolution and geometric guidance: learning constructs the map, while the map guides subsequent updates.

AI USE STATEMENT

In this work, we used generative AI tools to assist with proving mathematical claims. The overall theoretical framework, proof strategies, and key arguments were developed by the authors, while the role of generative AI in the proofs was limited to assisting with intermediate derivations and supporting steps. This assistance covered intermediate derivations and proof steps in Appendix H. These proofs support the theoretical discussion in Section 3.2.

We did not use generative AI tools for any other applicable tasks requiring disclosure. Qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools for language editing and polishing.

We have reviewed all AI-assisted work. The authors checked the derivations and proof steps against the stated assumptions and relevant mathematical results. The authors also reviewed the language edits to preserve the intended technical meaning.

We take responsibility for the final content of this work. This includes text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Sections 3 and 4 describe the proposed method and its implementations. Appendix D describes the OT-Fuse solver. Appendix E provides pseudocode for the applied algorithms. Appendix F.1 lists key hyperparameters. Appendix I describes the experimental tasks and evaluation protocols. Appendix H presents the theoretical assumptions and proof derivations.

REFERENCES

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International conference on machine learning*, pp. 22–31. PMLR, 2017.

Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

Shun-ichi Amari. *Information geometry and its applications*. Springer, 2016.

Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer, 2005.

Robert B Ash and Catherine A Doléans-Dade. *Probability and measure theory*. Academic press, 2000.

Nihat Ay, Jürgen Jost, Hông Vân Lê, and Lorenz Schwachhöfer. *Information geometry*, volume 64. Springer, 2017.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 3119–3137, 2024.

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research*, 7 (11), 2006.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

Sergey G. Bobkov and Friedrich Götze. Exponential integrability and transportation cost related to logarithmic sobolev inequalities. *Journal of Functional Analysis*, 163:1–28, 1999. URL `https://api.semanticscholar.org/CorpusID:7642937`.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

Yat Tin Chow and Wilfrid Gangbo. A partial laplacian as an infinitesimal generator on the wasserstein space. *Journal of Differential Equations*, 267(10):6065–6117, 2019.

Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006.

Peter Crooks and Maxence Mayrand. Symplectic reduction along a submanifold. *Compositio Mathematica*, 158(9):1878–1934, 2022.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

Robert Dadashi, Léonard Hussenot, Matthieu Geist, and Olivier Pietquin. Primal wasserstein imitation learning. *arXiv preprint arXiv:2006.04678*, 2020.

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in neural information processing systems*, 34:17695–17709, 2021.

Piero Deidda, Martin Burger, Mario Putti, and Francesco Tudisco. The graph-laplacian eigenvalue problem. *SIAM Journal on Mathematical Analysis*, 58(1):1–34, 2026.

Vincent Divol, Jonathan Niles-Weed, and Aram-Alexandre Pooladian. Tight stability bounds for entropic brenier maps. *International Mathematics Research Notices*, 2025(7):rnaf078, 2025.

Joseph L Doob. Probability as measure. *The Annals of Mathematical Statistics*, 12(2):206–214, 1941.

Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, 2016.

Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3 (4):128–135, 1999.

Josef Greilhuber and Willi Kepplinger. On arnold's transversality conjecture for the laplace-beltrami operator. *arXiv preprint arXiv:2312.16939*, 2023.

Xingrui Gu, Guanren Qiao, Chuyi Jiang, Tianqing Xia, and Hangyu Mao. Mimicking human intuition: Cognitive belief-driven q-learning. *arXiv preprint arXiv:2410.01739*, 2024.

Shuncheng He, Yuhang Jiang, Hongchang Zhang, Jianzhun Shao, and Xiangyang Ji. Wasserstein unsupervised reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 6884–6892, 2022.

Sham M Kakade. A natural policy gradient. *Advances in neural information processing systems*, 14, 2001.

Vijay Konda and John Tsitsiklis. Actor-critic algorithms. *Advances in neural information processing systems*, 12, 1999.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 3214–3252, 2022.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024.

Wenzhang Liu, Wenzhe Cai, Kun Jiang, Guangran Cheng, Yuanda Wang, Jiawei Wang, Jingyu Cao, Lele Xu, Chaoxu Mu, and Changyin Sun. Xuance: A comprehensive and unified deep reinforcement learning library. *arXiv preprint arXiv:2312.16248*, 2023.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Hariharan Narayanan and Partha Niyogi. On the sample complexity of learning smooth cuts on a manifold. In *COLT*, 2009.

Mary Chriselda Antony Oliver, Michael Roberts, Carola-Bibiane Schönlieb, and Matthew Thorpe. Laplace learning in wasserstein space. *arXiv preprint arXiv:2511.13229*, 2025.

Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

Adil Salim, Anna Korba, and Giulia Luise. The wasserstein proximal gradient algorithm. *Advances in Neural Information Processing Systems*, 33:12356–12366, 2020.

Mikayel Samvelyan, Tabish Rashid, Christian Schroeder De Witt, Gregory Farquhar, Nantas Nardelli, Tim GJ Rudner, Chia-Man Hung, Philip HS Torr, Jakob Foerster, and Shimon Whiteson. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043*, 2019.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015.

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

Dejan Slepcev and Matthew Thorpe. Analysis of p-laplacian regularization in semisupervised learning. *SIAM Journal on Mathematical Analysis*, 51(3):2085–2120, 2019.

Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

Antonio Terpin, Nicolas Lanzetti, Batuhan Yardim, Florian Dorfler, and Giorgia Ramponi. Trust region policy optimization with optimal transport discrepancies: Duality and algorithm for continuous actions. *Advances in Neural Information Processing Systems*, 35:19786–19797, 2022.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*, 2018.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2008.

Max-K von Renesse and Karl-Theodor Sturm. Transport inequalities, gradient estimates, entropy and ricci curvature. *Communications on pure and applied mathematics*, 58(7):923–940, 2005.

Haoran Wang, Thaleia Zariphopoulou, and Xun Yu Zhou. Reinforcement learning in continuous time and space: A stochastic control approach. *Journal of Machine Learning Research*, 21(198):1–34, 2020.

Hong Wang and Joshua Zahl. The assouad dimension of kakeya sets in r 3. *Inventiones mathematicae*, 241(1):153–206, 2025.

Zhenda Xie, Yixuan Wei, Huanqi Cao, Chenggang Zhao, Chengqi Deng, Jiashi Li, Damai Dai, Huazuo Gao, Jiang Chang, Liang Zhao, et al. mhc: Manifold-constrained hyper-connections. *arXiv preprint arXiv:2512.24880*, 2025.

Ruiyi Zhang, Changyou Chen, Chunyuan Li, and Lawrence Carin. Policy optimization as wasserstein gradient flows. In *International Conference on machine learning*, pp. 5737–5746. PMLR, 2018.

Zhengxin Zhang, Ziv Goldfeld, Kristjan Greenewald, Youssef Mroueh, and Bharath K Sriperumbudur. Gradient flows and riemannian structure in the gromov-wasserstein geometry: Z. zhang et al. *Foundations of Computational Mathematics*, pp. 1–93, 2025.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2), 2023.

Xiaojin Zhu, Zoubin Ghahramani, and John D Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International conference on Machine learning (ICML-03)*, pp. 912–919, 2003.

A Related Work

Laplacian and manifold regularization. Laplacian and manifold-regularization methods, manifold regularization, label propagation, diffusion maps, smooth functions on a data graph by minimising a graph Dirichlet energy, yielding harmonic extensions that converge to the Laplace-Beltrami operator under dense sampling (Zhu et al., 2003; Belkin et al., 2006; Greilhuber & Kepplinger, 2023; Coifman & Lafon, 2006; Deidda et al., 2026). Nonlinear variants based on p-Dirichlet energies improve boundary sensitivity in low-label regimes (Slepcev & Thorpe, 2019). These ideas have recently been lifted to *manifolds of probability measures*: endowing the action space with a ground metric induces 2-Wasserstein geometry on $\mathcal{P}_2(\mathcal{A})$, which admits a Riemannian-like structure via *Otto calculus*, tangent vectors identify with gradient fields, and Wasserstein gradient flows correspond to continuity-type PDEs (Ambrosio et al., 2005; Villani et al., 2008; Zhang et al., 2025). *Laplace learning in Wasserstein space* (Oliver et al., 2025; Chow & Gangbo, 2019) builds on this by treating measures as graph nodes, constructing graphs via W_2, and establishing Γ-convergence of discrete p-Dirichlet energies to continuous energies on Wasserstein submanifolds. All of this work, however, is offline: it regularises static graphs for label or embedding smoothing, whereas we treat the policy trajectory as a set of W_2 samples and regularise iteratively during task-driven learning.

Local optimization and Wasserstein geometry in ML. Most policy learning reduces to first-order updates in Euclidean parameter space, cross-entropy in supervised and sequence models, policy gradient and actor–critic in RL—with geometry introduced only locally via natural gradients, mirror descent, or KL-based trust regions (Amari, 1998; Kakade, 2001; Peyré et al., 2019; Beck & Teboulle, 2003; Schulman et al., 2017; 2015). This assumes small parameter or KL steps reflect the true scale of policy change, but it ignores Wasserstein proximity: KL-controlled updates can still produce large W_2 mass transport, notably for multimodal, moving-support, or continuous-action policies (Bobkov & Götze, 1999; von Renesse & Sturm, 2005; Villani et al., 2008; Xie et al., 2025), yielding oscillations, mode collapse, and out-of-distribution behaviour. OT and Wasserstein tools have been most influential in generative and diffusion modelling (Cuturi, 2013; De Bortoli et al., 2021); in RL and imitation learning they typically appear as a distribution-matching loss, reward signal, or occasional KL substitute for trust regions (Dadashi et al., 2020; Terpin et al., 2022; Zhang et al., 2018; Salim et al., 2020; He et al., 2022). These uses penalise marginal discrepancies. Our method is different: it treats the evolving policy sequence as geometric samples and imposes a Dirichlet/Laplacian trace constraint via a historical W_2 graph during training.

B Discussion

PLT frames policy learning as geometric reduction in behavioral distribution space. Although policies are parameterized in high dimensions, their behavioral evolution often follows a small number of experience-coherent directions. Historical policies can therefore act as Wasserstein landmarks for recovering a locally low-dimensional policy geometry, rather than merely as replay samples. This echoes the intuition of geometric reduction: redundant high-dimensional degrees of freedom are quotiented away, while the effective dynamics are described in a reduced space (Crooks & Mayrand, 2022). Unlike classical reduction, however, PLT derives this geometry from the training trajectory itself rather than from explicit symmetries or conservation laws.

This compression is not automatic. Kakeya-type phenomena remind us that if directions are sufficiently transverse across scales, low-dimensional concentration may fail (Wang & Zahl, 2025). Thus, PLT is most appropriate when policy evolution is locally coherent and historical policies form reliable Wasserstein neighborhoods. When trajectories are highly dispersed or the action metric does not reflect task semantics, historical smoothing may be harmful. Future work should therefore study diagnostics for trajectory compressibility, adaptive criteria for accepting or rejecting history-based smoothing, and learned transport metrics. Overall, PLT suggests a transport-geometric view of behavioral compression: learning not only policies from experience, but also the geometry of policy space induced by experience.

C NOTATION

This appendix summarizes the main mathematical symbols used throughout the paper. Unless otherwise stated, policies are interpreted as conditional probability measures over the action space.

Table 2: **General notation and Wasserstein preliminaries.**

Symbol	Meaning	Context
$\mathcal{S}$	State/context space	RL or general policy learning
x	Input, state, context, or prompt	General conditional policy input
s, s_t	State/context at time or iteration t	RL / PLT trace
$\mathcal{A}$	Action space	Policy output space
a, a'	Actions in $\mathcal{A}$	Action-level transport
$d_{\mathcal{A}}(a, a')$	Task-semantic ground metric on actions	Defines transport cost
$\theta \in \Theta \subset \mathbb{R}^d$	Policy parameters	Parametric policy family
$\pi_\theta(\cdot \mid x)$	Conditional policy distribution	Element of $\mathcal{P}_2(\mathcal{A})$
μ, ν	Generic probability measures	Optimal transport notation
$\mathcal{P}(\mathcal{A})$	Probability measures on $\mathcal{A}$	Distribution space
$\mathcal{P}_2(\mathcal{A})$	Measures on $\mathcal{A}$ with finite second moment	Wasserstein space
$\Pi(\mu, \nu)$	Couplings with marginals μ and ν	OT definition
$W_2(\mu, \nu)$	2-Wasserstein distance	Policy transport metric
$W_{2,\varepsilon}(\mu, \nu)$	Entropic-regularized Wasserstein distance	Sinkhorn approximation
ε	Entropic OT regularization strength	Sinkhorn / OT fusion
$D_{\mathrm{KL}}(\mu\|\nu)$	Kullback–Leibler divergence	Trust-region / proximal anchor
$T_{\#}\mu$	Pushforward of μ under map T	OT / measure transport
$\mathrm{Bar}_\varepsilon(\{\mu_i, w_i\})$	Entropic Wasserstein barycenter	Local policy aggregation
$\mathbb{E}[\cdot]$	Expectation	Reward / loss definitions

Table 3: **Notation for experience-induced Wasserstein policy geometry and PLT.**

Symbol	Meaning	Context
Λ_x	Realizable policy set $\{\pi_\theta(\cdot \mid x) : \theta \in \Theta\}$	Policy manifold hypothesis
$\widetilde{\Lambda}_x$	Subset of policies visited during training	Experience-induced geometry
d_x	Local intrinsic dimension of $\widetilde{\Lambda}_x$	Low-dimensional policy geometry
L_r	Lipschitz constant of reward in action metric	Behavior relevance of W_2
L_P	Optional Lipschitz constant of transition kernel	MDP regularity
$\mathcal{E}_t$	Evolution sample (s_t, π_t, ξ_t)	PLT memory item
ξ_t	Optional auxiliary signal, e.g., advantage or gradient summary	PLT trace
$\mathcal{H}_t$	Policy Laplacian Trace $\{\mathcal{E}_i\}_{i \le t}$	Historical policy trace
$\mathcal{U}_t$	Base myopic policy update operator	Standard learner
$\mathcal{M}_t$	PLT-induced fusion operator	History-aware update
$\mathcal{T}$	Task operator, e.g., Bellman or policy-gradient objective	Learning objective
$\mathcal{N}_k(s_t)$	Top-k retrieved neighborhood of s_t	Local policy graph
k	Number of retrieved neighbors	PLT graph construction
$\kappa(s, s')$	Similarity kernel between states/contexts	Retrieval weighting
$\phi(s), \psi(s)$	State/context embeddings for similarity	RBF or attention kernel
w_{tj}	Normalized edge weight from s_t to neighbor s_j	Local graph weights
$\Phi_t(\pi)$	Local Wasserstein–Dirichlet memory potential	PLT geometric regularizer
$\nabla_{W_2}\Phi_t$	Wasserstein first variation / gradient of Φ_t	Laplacian force
$\mathrm{Log}_\pi(\nu)$	Wasserstein logarithm direction from π to ν	OT tangent direction
$\hat{g}_t$	Linearized task loss functional	Task-gradient drift
λ	Weight of the geometry regularizer	PLT objective
η	Proximal step size	Variational update
ρ	KL anchoring weight in alignment objective	PPO implementation

Table 4: **Notation for the RL/PPO instantiation of PLT.**

Symbol	Meaning	Context
$\mathcal{B}$	PPO rollout buffer	RL implementation
T	Rollout horizon	PPO data collection
(s_t, a_t, r_t, s_{t+1})	Transition tuple	MDP interaction
r_t	Reward at time t	RL objective
$\widehat{A}_t$	Estimated advantage	PPO surrogate
$\pi_{\theta_{\text{old}}}$	Behavior policy used for rollout collection	PPO clipping
$\mathcal{L}_{\text{actor}}$	PPO actor objective	Policy optimization
ϵ	PPO clipping parameter	PPO surrogate
$f_{\text{enc}}(s)$	State encoder used for retrieval key	PLT memory retrieval
z_t	Retrieval key $f_{\text{enc}}(s_t)$	PLT trace storage
φ_t	Stored lightweight policy representation	Memory item
d_φ	Dimension of stored policy representation	Complexity analysis
p_i	Stored categorical policy vector	Discrete action case
(μ_i, σ_i)	Stored Gaussian policy parameters	Continuous action case
$I(s)$	Retrieved neighbor index set for query state s	PPO+PLT retrieval
τ	Retrieval temperature	Soft neighbor weighting
$\pi_{\text{hist}}(\cdot \mid s)$	Historical reference policy from retrieved neighbors	Local barycentric prior
$\pi_{\text{align}}(\cdot \mid s)$	Conservative Wasserstein-aligned reference policy	PPO+PLT anchor
$A = \{a_i\}_{i=1}^m$	Samples from current policy	Sample-based OT
$B = \{b_j\}_{j=1}^n$	Samples from historical policy	Sample-based OT
C_{ij}	OT cost matrix entry $\|a_i - b_j\|_2^2$	Sinkhorn fusion
$P^\star$	Entropic OT / Sinkhorn transport plan	Barycentric projection
m	Number of support points or particles	OT approximation
I	Number of Sinkhorn iterations	Complexity analysis
α	Conservative barycentric interpolation weight	Alignment step
M	PLT trace capacity	Memory complexity
d_z	Retrieval-key dimension	Memory / retrieval complexity
Q	Number of queried states using PLT fusion	Runtime complexity

Table 5: **Notation for the LLM-as-policy instantiation and overhead analysis.**

Symbol	Meaning	Context
$\mathcal{V}$	Vocabulary / discrete action space	LLM-as-policy
V	Vocabulary size $\lvert\mathcal{V}\rvert$	Complexity analysis
Δ^{V-1}	Probability simplex over vocabulary tokens	Next-token policy space
n	Training iteration index	LLM training
t	Token position / time index	Sequence modeling
$x_{\le t}$	Prefix up to position t	Autoregressive policy input
y_t	Ground-truth next token	Cross-entropy target
$p^{\mathrm{out}}_{n,t}$	Model output distribution $p_{\theta_n}(\cdot \mid x_{\le t})$	LLM policy
$h_{n,t}$	Contextual hidden representation	Retrieval key
d_h	Hidden-state dimension	Memory / retrieval complexity
$\mathcal{M}_\ell$	Position-conditioned memory bank	LLM PLT memory
$L_{\max}$	Maximum sequence length / number of positions	LLM memory organization
$s_{n,t}$	Surprisal or difficulty signal $-\log p^{\mathrm{out}}_{n,t}(y_t)$	Hardness gating
τ_{ret}	Retrieval temperature	LLM memory retrieval
q_i	Stored next-token distribution in memory	Memory item
$p^{\mathrm{mem}}_{n,t}$	Memory-fused distribution from retrieved neighbors	Local policy prior
E	Token embedding matrix	Semantic transport proxy
e_a	Embedding of token a	Vocabulary geometry
d_e	Token embedding dimension	Semantic proxy
$\mu(p)$	Embedding mean $p^\top E$	Semantic transport representation
$d_{\mathrm{sem}}(p,q)$	Semantic distance $\lVert\mu(p)-\mu(q)\rVert_2^2$	Scalable OT proxy
$p^\star_{n,t}$	Ideal proximal PLT target distribution	Variational interpretation
$R(q;p^{\mathrm{mem}})$	Transport or semantic regularization term	LLM PLT objective
$g_{n,t}$	Inner fusion gate	Output-memory interpolation
$\gamma_{n,t}$	Hardness gate for applying PLT	Sparse token fusion
β	Sharpness of hardness gate	Gating hyperparameter
τ_{hard}	Hardness threshold	Gating hyperparameter
$p^{\mathrm{PLT}}_{n,t}$	Inner PLT-fused distribution	LLM PLT output
$p^{\mathrm{final}}_{n,t}$	Final output distribution after hardness gating	Training loss
$\mathcal{L}_{\mathrm{CE}}$	Cross-entropy loss	Supervised LLM training
B	Batch size	Complexity analysis
N	Number of stored memory entries	LLM memory complexity
d_{sem}	Compressed semantic-policy dimension	LLM storage / fusion complexity
Q	Number of tokens selected by hardness gate	Sparse PLT runtime

Table 6: **Notation used in the theoretical analysis.**

Symbol	Meaning	Context
$G_t = (V_t, E_t)$	Experience-induced graph at time t	Graph Laplacian analysis
$w_{ij} = w_{ji}$	Symmetric nonnegative edge weight	Dirichlet energy
$\mathcal{S}_t = \{s_i\}_{i=1}^{t}$	Visited or replay-buffer state set	Historical samples
π_i	Shorthand for $\pi(\cdot \mid s_i)$	Policy graph node
Λ_s	Realizable policy set at state s	Policy submanifold
$\ell_t(\mu)$	Task-driven local loss functional	Proximal update
$F_{\#}\mu$	Pushforward of μ under F	Measure transport
Id	Identity map on $\mathcal{A}$	Wasserstein logarithm
$T_{\mu\to\nu}$	Optimal transport map from μ to ν	OT geometry
$T_\mu\mathcal{P}_2(\mathcal{A})$	Wasserstein/Otto tangent space at μ	First-variation calculus
$L^2(\mu)$	Square-integrable vector fields under μ	Tangent-space inner product
$\langle u, v\rangle_{L^2(\mu)}$	$\int u(a)^\top v(a)\, d\mu(a)$	Otto calculus
∂^{W_2}	Wasserstein subdifferential	Non-smooth / non-unique OT case
E_{mem}	Wasserstein graph Dirichlet energy	Laplacian smoothing
$E_{\mathrm{mem}}^{(p)}$	p-Dirichlet graph energy	p-Laplacian analysis
$\mathrm{prox}_{\tau\Phi_t}^{W_2}$	Wasserstein proximal operator	Stability / contraction proof
τ	Proximal flow step size	OT-Fuse / theory

D OT-FUSE: EFFICIENT APPROXIMATE SOLVER FOR PROXIMAL FUSION

Based on the Policy Laplacian learning concepts introduced in Section 4, we briefly present OT-FUSE, a lightweight routine that approximately solves this variational update using a small number of Sinkhorn iterations. After turns the patch into a Wasserstein–Dirichlet potential $\Phi_t(\pi)$ that penalizes geometric roughness relative to nearby historical policies. Then updates the current policy by solving the proximal fusion objective (Eq. 15), which couples the task signal with Laplacian-style smoothing in distribution space.

Directly minimizing Eq. 15 can be expensive because it combines a task functional, multiple Wasserstein terms, and a trust-region constraint in policy space. We therefore adopt a two-step approximation: first form a *task-leading* policy $\tilde{\pi}_t$ by applying the base (myopic) update,

$$\tilde{\pi}_t \;=\; \mathcal{U}_t(\pi_t; \mathcal{T}), \tag{39}$$

and then fuse $\tilde{\pi}_t$ with the local geometric prior induced by $\mathcal{H}_t$. Concretely, we approximate Eq. 15 by the anchored barycentric proximal problem

$$\pi_{t+1} \;=\; \arg\min_{\pi \in \mathcal{P}_2(\mathcal{A})} \lambda\, \Phi_t(\pi) \;+\; \frac{1}{\eta}\, \mathrm{KL}(\pi \| \tilde{\pi}_t), \tag{40}$$

where $\Phi_t(\pi) = \frac{1}{2} \sum_{j \in \mathcal{N}_k(s_t)} w_{tj} W^2_{2,\varepsilon}(\pi, \pi_j)$. Intuitively, Eq. equation 40 moves the task-leading policy toward a local Wasserstein barycentric configuration of historical neighbors, while the KL term keeps the step conservative.

The resulting OT-Fuse iteration admits a geometric optimization interpretation: it implements an explicit Euler step of the entropic Wasserstein gradient flow induced by Φ_t, and provides a first-order approximation to the corresponding Wasserstein proximal point (Proposition H.12). Assume $\mathcal{A} = \{a_1, \ldots, a_n\}$ with ground cost matrix $C \in \mathbb{R}_+^{n \times n}$, e.g., $C_{uv} = d_{\mathcal{A}}(a_u, a_v)^2$. Let $K \coloneqq \exp(-C/\varepsilon)$ be the entropic OT kernel. For entropic OT, a coupling between a candidate π and a neighbor π_j takes the Sinkhorn form $P_j = \mathrm{diag}(u_j)\, K\, \mathrm{diag}(v_j)$ with marginals $P_j \mathbf{1} = \pi$ and $P_j^\top \mathbf{1} = \pi_j$. The anchored barycenter equation 40 can be approximated by a few iterations of a Sinkhorn-barycenter fixed point, warm-started from $\tilde{\pi}_t$.

Algorithm 1 OT-FUSE: few-step entropic barycenter with KL anchor (discrete actions)

Require: Anchor policy $\tilde{\pi} \in \Delta^n$, neighbors $\{\pi_j\}_{j=1}^k \subset \Delta^n$, weights $w_j \geq 0$ ($\sum_j w_j = 1$), cost C, entropic ε, inner steps L, anchor exponent $\beta \in (0, 1)$.
Ensure: Fused policy $\pi_{\text{fuse}} \in \Delta^n$

$K \leftarrow \exp(-C/\varepsilon)$; $\pi^{(0)} \leftarrow \tilde{\pi}$ *// Sinkhorn kernel + init*
for $\ell = 0, 1, \ldots, L-1$ **do**
  for $j = 1, \ldots, k$ **do**
    $v_j \leftarrow \pi_j \oslash (K^\top u_j)$; $\quad u_j \leftarrow \pi^{(\ell)} \oslash (K v_j)$
    $m_j \leftarrow K v_j$ *// message to barycenter*
  end for
  $\pi^{(\ell+1)} \leftarrow \mathrm{Normalize}\Big(\tilde{\pi}^\beta \odot \prod_{j=1}^k m_j^{(1-\beta) w_j}\Big)$
end for
return $\pi_{\text{fuse}} \leftarrow \pi^{(L)}$

Here $\odot$ and exponents are taken elementwise, $\oslash$ denotes elementwise division, and $\mathrm{Normalize}(\cdot)$ rescales a positive vector to sum to one. The parameter β controls the strength of the KL anchor: larger β keeps π_{t+1} closer to $\tilde{\pi}_t$, while smaller β increases geometric fusion. In practice, L can be very small (e.g., $L \in \{1, 2, 3\}$) with warm-started scalings, making OT-FUSE a lightweight drop-in solver.

OT-FUSE can be understood as a fast approximate proximal step toward a local Wasserstein barycenter induced by the neighborhood $\{\pi_j\}$. Viewed through the minimizing-movement perspective, Eq. 40 is an implicit (proximal) time discretization of a drift–diffusion dynamics on the policy manifold: the task step produces $\tilde{\pi}_t$ (drift), while the Dirichlet potential Φ_t induces Laplacian-style smoothing (diffusion) in $W_{2,\varepsilon}$ geometry. OT-FUSE implements this step approximately using entropic OT and a few Sinkhorn iterations.

Algorithm 2 Policy Laplacian Trace (PLT): task update + Wasserstein Laplacian fusion

Require: Stepsize η, geometry weight λ, mixed trust weight γ, entropic OT ε, capacity M, kNN size k, similarity kernel κ, initial params θ_0.

1: $\mathcal{H} \leftarrow \emptyset$; initialize policy π_{θ_0}
2: **for** $t = 0, 1, 2, \dots$ **do**
3: Observe context/state s_t; current policy $\pi_t \leftarrow \pi_{\theta_t}(\cdot \mid s_t)$
4: Log trace item $\mathcal{E}_t \leftarrow (s_t, \pi_t, \xi_t)$ *(optional ξ_t: advantage/grad summary)*
5: $\mathcal{H} \leftarrow \textsc{Insert}(\mathcal{H}, \mathcal{E}_t; M)$ *(FIFO / similarity-prune)*
6: Retrieve neighbors $\mathcal{N} \leftarrow \textsc{kNN}(s_t, \mathcal{H}, \kappa, k)$; weights $w_j \propto \kappa(s_t, s_j)$ for $j \in \mathcal{N}$ (normalize to $\sum_j w_j = 1$)
7: Task-leading update (any base learner): $\tilde{\theta}_t \leftarrow \textsc{BaseUpdate}(\theta_t)$; $\tilde{\pi}_t \leftarrow \pi_{\tilde{\theta}_t}(\cdot \mid s_t)$
8: Fuse in distribution space:

$$\pi_{t+1} \leftarrow \begin{cases} \text{OT-}\textsc{Fuse}(\tilde{\pi}_t, \{\pi_j\}_{j\in\mathcal{N}}, \{w_j\}, \lambda, \varepsilon) & \mathcal{N} \neq \emptyset \\ \tilde{\pi}_t & \mathcal{N} = \emptyset \end{cases} \tag{41}$$

9: Project back to parametric policy (distillation / projection):

$$\theta_{t+1} \leftarrow \arg\min_{\theta} \; D(\pi_{t+1} \,\|\, \pi_\theta(\cdot \mid s_t)), \quad D = KL + \frac{\gamma}{2} W_{2,\varepsilon}^2 \tag{42}$$

10: **end for**

E Pseudocode for Applied PLT Algorithm

E.1 PPO + PLT

Algorithm 3 PPO with Policy Laplacian Trace

Hyper: clip ϵ, neighbors k, batch size b
Initialize policy π_θ, PLT memory $\mathcal{H}$ (capacity M), Reply Buffer $\mathcal{B}$ (capacity T)
for step $t = 1, 2, \dots$ **do**
  Observe s_t, take $a_t \sim \pi_\theta(\cdot|s_t)$, receive (r_t, s_{t+1}); store $(s_t, a_t, r_t, s_{t+1}, \log \pi_{\theta_{\text{old}}}(a_t|s_t), \hat{A}_t)$ in $\mathcal{B}$
  if step $t \geq$ start learning step **then**
    Sampling $(s_i, a_i, r_i, s_{i+1}, \log \pi_{\theta_{\text{old}}}(a_i|s_i), \hat{A}_i)_{i=1}^b$ from $\mathcal{B}$
    if $|\mathcal{H}| < k$ **then**
      $\pi_{\text{align}}(\cdot|s) \leftarrow \pi_{\theta_{\text{old}}}(\cdot|s)$ ▷ fallback
      continue
    else
      Retrieve $I(s_i) =$ top-k neighbors in $\mathcal{H}$ and compute $w_i(s_i)$ (cf.19)
      if π_θ is discrete (categorical) **then**
        $\pi_{\text{hist}} \leftarrow \sum_{i\in I(s_i)} w_i \, p_i$ (cf.20)
      else if π_θ is continuous (gaussian) **then**
        $\pi_{\text{hist}} \leftarrow \sum_{i\in I(s_i)} w_i \, \mathcal{N}(\mu_i, \sigma_i^2)$ (cf.21)
        Estimate π_{align} through Sinkhorn plan and barycentric projection (cf.22 $\sim$ cf.24)
      end if
    end if
    Policy Update (cf.25):

$$\mathbb{E}_{(s,a,\hat{A})\sim\mathcal{B}}\Big[\min\Big(\tfrac{\pi_{\text{align}}(a|s)}{\pi_{\theta_{\text{old}}}(a|s)}\hat{A},\ \text{clip}(\cdot, 1\pm\epsilon)\hat{A}\Big)\Big]$$

    Update θ and Push $(s_t, \varphi_t \coloneqq \pi_{\theta_{\text{old}}}(\cdot|s_t))$ into PLT $\mathcal{H}$
  end if
end for

E.2 LLM + PLT

Algorithm 4 LLM Training with PLT

Input: dataset $\mathcal{D}$, parameters θ_0, memory $\mathcal{M} \leftarrow \emptyset$, min-memory $M_{\min}$, kNN size k, hardness threshold τ, gates α, β, step size η.
for $t = 0, 1, \ldots, T$ **do**
  Sample a minibatch $\{(x_b, y_b)\}_{b=1}^{B} \sim \mathcal{D}$.
  Compute logits $z_b = f_\theta(x_b)$ and output distributions $p_b^{\text{out}} = \mathrm{Softmax}(z_b)$.
  for $b = 1, \ldots, B$ **do**
    if $|\mathcal{M}| \geq M_{\min}$ **then**
      Retrieve $\mathcal{N}_k(x_b) = \{(h_i, p_i)\}$ by kNN over keys (h_b).
      Weights $w_i = \mathrm{Softmax}(\kappa(h_b, h_i))$ and memory barycenter $p_b^{\text{mem}} = \sum_{i \in \mathcal{N}_k} w_i \,\mathrm{stopgrad}(p_i)$.
      Fusion gate $g_b^{\text{fuse}} = \sigma\Big(\mathrm{MLP}(h_b) - \alpha\, d_{\mathrm{OT}}(p_b^{\text{out}}, p_b^{\text{mem}})\Big)$.
      Fused distribution (no gradient): $p_b^{\text{fuse}} = \mathrm{stopgrad}\big((1 - g_b^{\text{fuse}})p_b^{\text{out}} + g_b^{\text{fuse}} p_b^{\text{mem}}\big)$.
    else
      $p_b^{\text{fuse}} \leftarrow \mathrm{stopgrad}(p_b^{\text{out}})$.
    end if
    Surprisal $s_b = -\log p_b^{\text{out}}(y_b)$ and hardness gate $\gamma_b = \sigma(\beta(s_b - \tau))$.
    Geometry-aware target $\pi_b^{\text{target}} = (1 - \gamma_b)\,\mathrm{stopgrad}(p_b^{\text{out}}) + \gamma_b\, p_b^{\text{fuse}}$.
  end for
  Optimize with supervised CE plus distributional regularization:

$$\mathcal{L}(\theta) = \frac{1}{B} \sum_{b=1}^{B} \Big(\underbrace{-\log p_b^{\text{out}}(y_b)}_{\text{CE to labels}} + \lambda \underbrace{H(\pi_b^{\text{target}},\, p_b^{\text{out}})}_{\text{PLT distillation}} \Big),$$

  $\theta_{t+1} \leftarrow \theta_t - \eta \nabla_\theta \mathcal{L}(\theta)$.
  Log experience: $\mathcal{M} \leftarrow \mathcal{M} \cup \{(\mathrm{stopgrad}(h_b),\, \mathrm{stopgrad}(p_b^{\text{out}}))\}_{b=1}^{B}$.
end for

F More Experiment

F.1 Hyperparameter

Table 7: **Key hyperparameters for W2/PLT.** We defer full training settings to Appendix F.1.

Module	Hyperparameter	Value	Description
LLM	memory_size	20000	Memory buffer capacity
LLM	λ_{W_2}	1.0	Weight of the internal W_2 loss
LLM	inner_loss_max_weight	0.1	Maximum contribution of the inner loss
LLM	sinkhorn_reg (ε)	0.02	Entropic Sinkhorn regularization strength
LLM	memory_warmup_steps	2000	Steps before activating memory
RL	horizon $\times$ parallel_envs	128×40	Rollout buffer size
RL	ot_bank_capacity	100	OT memory bank capacity
RL	ot_update_freq	5	OT update frequency (iterations)
RL	clip_range	0.2	PPO clipping range
RL	γ, λ_{GAE}	0.99, 0.95	Discount factor and GAE coefficient

F.2 More Experiment Results

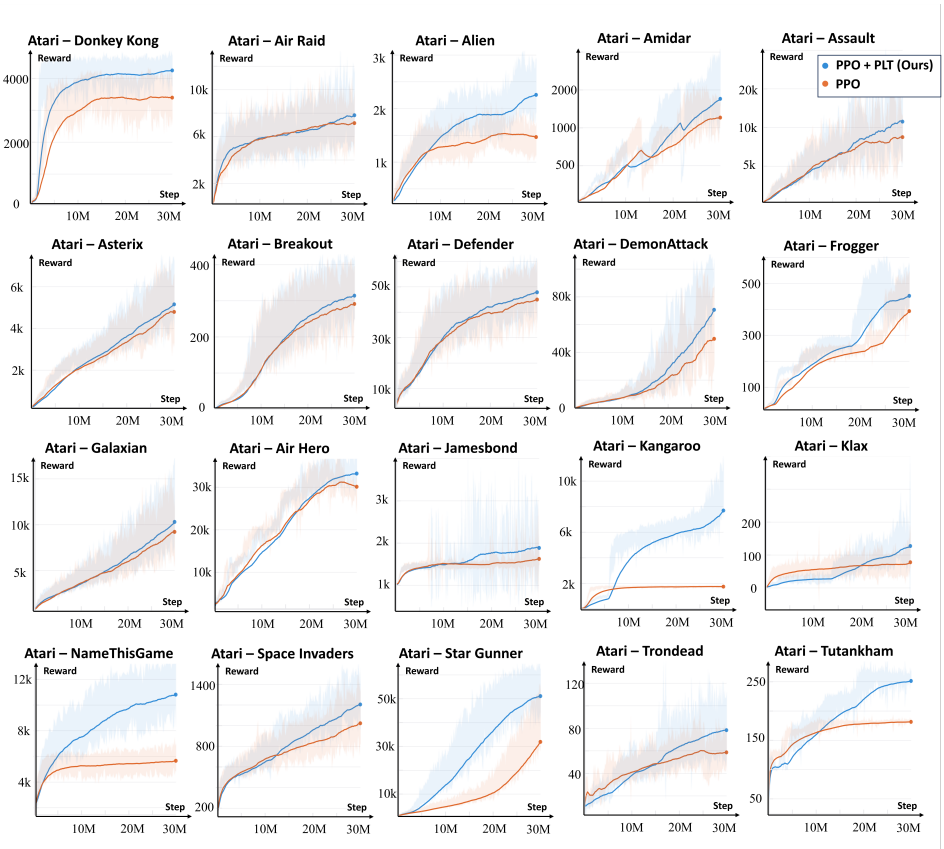

Figure 3: Learning curves with PLT compare with PPO baseline cumulative returns on Atari Games.

Table 8: **Continuous-control results with PPO+MNEMO.** We report episodic return as mean ± standard deviation after 2M environment steps. Higher is better. Green indicates relative improvement over PPO, computed as $(\text{MNEMO} - \text{PPO})/|\text{PPO}|$.

Environment	PPO	PPO+PLT
MuJoCo tasks (2M steps)		
Walker2d	3209.14 ± 1374.56	4586.33 ± 2110.27(+42.9%)
Reacher	-5.52 ± 0.58	-5.26 ± 1.53(+4.7%)
Swimmer	92.29 ± 0.87	111.10 ± 5.88(+20.4%)
HalfCheetah	5052.39 ± 1999.43	6659.97 ± 1316.39(+31.8%)
HumanoidStandup	58355.59 ± 2351.22	117143.71 ± 48593.62(+100.7%)
Humanoid	311.98 ± 28.98	444.68 ± 62.63(+42.5%)
Ant	2250.26 ± 270.81	2667.34 ± 297.72(+18.5%)
Hopper	1851.90 ± 1522.27	2378.05 ± 862.16(+28.4%)
Box2D tasks (2M steps)		
BipedalWalker	95.27 ± 78.52	264.86 ± 36.13(+178.0%)

Table 9: **Comprehensive Performance Analysis.** "+ PLT." denotes our Wasserstein method. **Bold** indicates the best performance in each category. Green values represent significant gains over the baseline, and shaded cells highlight key zero-to-one breakthroughs.

Model	Param.	General Acc.		Retrieval Acc.		Official Acc.		Conflict Acc.	
		Base	+ PLT.	Base	+ PLT.	Base	+ PLT.	Base	+ PLT.
GPT-Nano	2.5M	8.38%	**9.09%** (+8.5%)	8.38%	**10.62%** (+26.7%)	0.0%	**20.0%** (+∞%)	10.53%	10.53%
GPT-Micro	5M	6.11%	**10.39%** (+70.0%)	7.02%	**11.20%** (+59.5%)	0.0%	0.0%	6.98%	**11.26%** (+61.3%)
GPT-Mini	10M	8.35%	**10.15%** (+21.6%)	9.71%	**11.49%** (+18.3%)	0.0%	0.0%	10.00%	**11.22%** (+12.2%)
Gopher	44M	6.05%	**11.00%** (+81.8%)	7.10%	**12.50%** (+76.1%)	0.0%	0.0%	7.28%	**12.23%** (+68.0%)
OpenAI-GPT	117M	9.23%	**10.22%** (+10.7%)	10.53%	**11.78%** (+11.9%)	0.0%	**12.5%** (+∞%)	10.61%	**12.86%** (+21.2%)
GPT2	124M	10.83%	**12.20%** (+12.7%)	12.32%	**13.77%** (+11.8%)	0.0%	0.0%	8.79%	**13.11%** (+49.1%)
GPT2-Medium	355M	11.28%	11.28%	12.00%	**12.65%** (+5.4%)	0.0%	**25.0%** (+∞%)	**11.89%**	10.64%
GPT2-Large	774M	11.26%	**15.82%** (+40.5%)	10.00%	**18.18%** (+81.8%)	40.0%	**40.0%**	8.33%	**17.06%** (+104.8%)

Table 10: **Training & Long-Context Bucket Performance.** "+PLT" denotes our Policy Laplacian Trace variant. **Bold** indicates the better result within each metric (higher is better for accuracies; lower is better for loss/NLL). Green denotes improvements over baseline.

	Loss (Late) ↓		Move Acc ↑		Pass. Acc ↑		Token Acc (W) ↑		NLL ↓ (Std)	
Model	Base	+PLT	Base	+PLT	Base	+PLT	Base	+PLT	Base	+PLT
Gopher	1.84	**1.16**	45.6%	**46.6%**	32.4%	**34.0%**	54.5%	**58.0%**	2.87 (—)	**2.81 (1.83)**
GPT-Nano	1.58	**1.01**	34.1%	**34.6%**	41.8%	**42.7%**	56.9%	**59.6%**	2.96 (—)	**2.81 (1.90)**
GPT-Micro	2.03	**1.20**	48.8%	**49.3%**	13.2%	**22.5%**	48.8%	**51.8%**	3.19 (—)	**2.97 (1.82)**
GPT-Mini	1.35	**1.05**	46.1%	**47.7%**	35.7%	**40.5%**	58.7%	**59.3%**	2.56 (—)	**2.44 (1.68)**
OpenAI GPT	1.80	**1.68**	45.2%	**47.9%**	33.8%	**40.7%**	58.7%	**60.0%**	2.80 (—)	**2.71 (2.00)**
GPT-2	1.71	**1.01**	41.8%	**43.9%**	30.4%	**32.3%**	56.3%	**59.7%**	2.92 (—)	**2.58 (1.86)**
GPT-2 Medium	1.70	**1.23**	40.6%	**42.5%**	37.8%	**42.8%**	54.6%	**55.8%**	2.90 (—)	**2.58 (1.86)**

Table 11: **Policy Laplacian Trace (PLT) Enhanced Few-Shot Learning. +PLT** denotes our Wasserstein memory method with Policy Laplacian Trace. **Bold** indicates the best performance within each row. Green values denote absolute improvements over baseline, green shading highlights significant few-shot gains.

Model	**Param.**	**Base Categories**		**Few-Shot (New Cat.)**		**Overall**		**Few-Shot Gain**	
		Baseline	+PLT	Baseline	+PLT	Baseline	+PLT	Abs. Δ	Rel.
GPT-Micro	5M	1.0000	1.0000	0.1078	**0.3922**	0.78	**0.85**	+0.2844	×3.64
GPT-Nano	2.5M	1.0000	1.0000	0.0000	0.0000	0.70	0.70	0.0000	–
GPT-Mini	10M	1.0000	1.0000	0.1386	**0.1683**	0.82	**0.83**	+0.0297	×1.21
Gopher-44M	44M	0.9794	**1.0000**	0.0097	**0.1359**	0.85	**0.90**	+0.1262	×14.0
OpenAI-GPT	117M	0.9697	**1.0000**	0.0000	**0.0099**	0.88	**0.93**	+0.0099	∞
GPT-2	124M	0.9906	0.9906	0.0213	**0.1489**	0.89	**0.90**	+0.1276	×6.99
GPT-2-Medium	355M	0.9794	**1.0000**	0.0000	0.0000	0.90	**0.93**	0.0000	–
GPT-2-Large	774M	1.0000	0.9703	0.0000	**0.0101**	0.91	0.87	+0.0101	∞

F.3 Ablation on neighborhood size k

We study the sensitivity of our method to the neighborhood size k used for retrieving historical policies / constructing local neighborhood relations in Wasserstein space. Figure 4 reports results on Atari Alien with $k \in \{5, 10, 20, 50\}$. Overall, performance improves as k increases: k=20 and k=50 consistently outperform smaller neighborhoods, with k=50 achieving the best final return. In contrast, very small neighborhoods (e.g., k=5) yield weaker late-stage performance, suggesting that overly sparse local graphs provide noisy or incomplete estimates of the underlying policy geometry, which reduces the effectiveness of Laplacian-style regularization and neighborhood-based updates. Notably, compared to the baseline, our method continues to improve beyond mid training, whereas the baseline tends to plateau. These observations support the intuition that reliable local neighborhood structure in Wasserstein space is important for our approach. In practice, we find that moderate-to-large k (e.g., 20) provides a favorable trade-off between performance and computation.

F.4 Ablation on the Alignment Kernel in Eq. 22

We conduct an additional ablation on the alignment kernel used in Eq. 22 to assess the effect of different similarity/regularization choices. Concretely, we compare (i) a pure KL alignment, (ii) a Wasserstein-only (PLT) variant, and (iii) the combined objective in Eq. 22. We instantiate all variants within PPO and evaluate on the Atari *DonkeyKong* environment.

Table 12 reports mean episodic return and standard deviation across seeds. The combined alignment in Eq. 22 achieves the best average performance and the lowest variance, supporting our motivation for coupling local stability (KL anchoring) with transport-aware alignment (Wasserstein) rather than relying on either component alone.

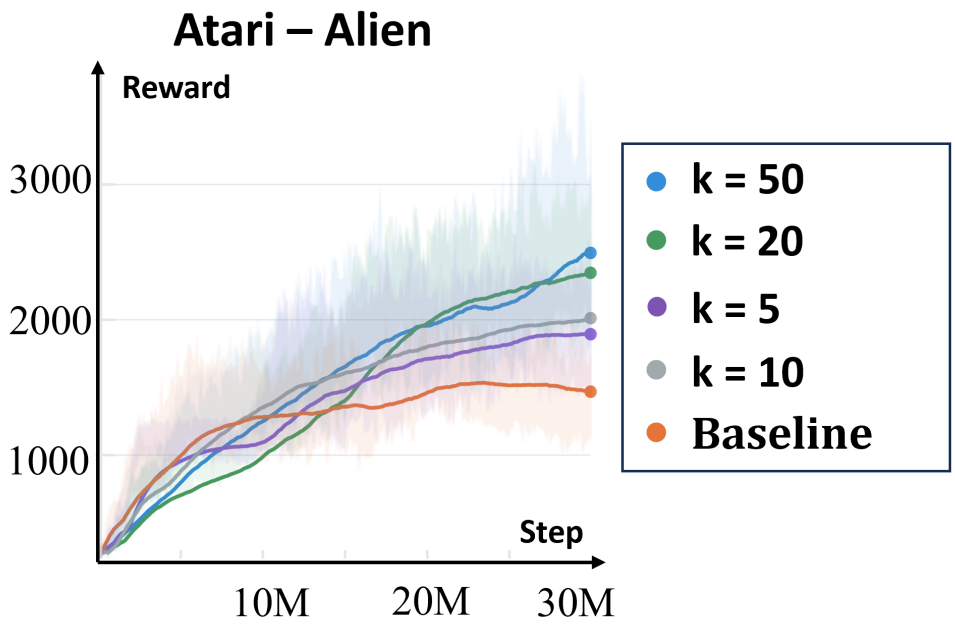

Figure 4: Ablation on the neighborhood size k on ATARI ALIEN. Curves show mean evaluation return over training steps; shaded areas indicate variability across seeds.

Table 12: **Kernel ablation on Atari *DonkeyKong*.** Mean $\pm$ std episodic return over seeds. The combined alignment in Eq. 22 performs best and yields lower variance.

Method (PPO)	**Return (mean $\pm$ std)**
PLT + KL (Eq. 22)	$\mathbf{4249.84 \pm 458.58}$
PLT (*Wasserstein-only*)	4207.00 ± 627.33
PLT + KL (KL-only alignment)	4140.70 ± 718.25
Baseline PPO	3397.04 ± 774.55

F.5 PLT VS. SIMPLER HISTORY-AWARE BASELINES

To isolate whether PLT's benefits stem from its Wasserstein-Laplacian structure or merely from smoothing with historical information, we compare against two simpler baselines on CarRacing-v2: *Parameter EMA* applies a uniform exponential moving average to θ_t, independent of the current state; *Output EMA* smooths the action distribution in simplex geometry, also with a state-independent coefficient. Table 13 shows that both baselines score negatively, indicating that uniform history smoothing does not help, and can actively hurt, in a continuous control task with rapidly changing state contexts. PLT, by contrast, achieves a positive return. Two factors contribute: (i) PLT retrieves state-specific historical neighbours, so the smoothing direction adapts to the current input, and (ii) the Wasserstein barycenter respects the geometry of the action space, penalising large transport displacements rather than mixing probabilities in simplex coordinates. This suggests that neither history-awareness alone nor transport geometry alone suffices; their combination is what makes PLT effective.

Table 13: CarRacing-v2: PLT vs. simpler history-aware baselines. Higher is better.

Method	Smooth space	Input-conditioned	Return ↑
Parameter EMA	Θ (parameter)	No	-41.26
Output EMA	Δ^{V-1} (simplex)	No	-34.43
PLT (ours)	$\mathcal{P}_2(\mathcal{A})$ (Wasserstein)	Yes	$\mathbf{14.86}$

F.6 Low-Dimensionality and Manifold Diagnostics

Motivation. A central premise of this work is that, for each input x, the realized policy trajectory $\{\pi_{\theta_t}(\cdot \mid x)\}_t$ does not fill the ambient parameter space but instead concentrates on an *experience-induced low-dimensional subset* of $\mathcal{P}_2(\mathcal{A})$ (idealized as a submanifold). To empirically substantiate this premise and to quantify the geometric effect of PLT, we conduct diagnostics on policy snapshots along training and report three complementary geometry indicators: (i) *effective dimension*, (ii) *local (intrinsic) dimension*, and (iii) *trajectory curvature* in Wasserstein space.[3]

Evidence for experience-induced low-dimensional structure. Despite operating in an ambient parameter space of roughly 10^6 dimensions, the policy trajectory exhibits a markedly low effective dimension: the estimated effective dimension is approximately 46, corresponding to a compression ratio exceeding 2×10^4:1. This provides direct empirical support that the learned policies concentrate on a low-dimensional structure, consistent with the "experience-induced Wasserstein policy manifold" hypothesis.

Geometry regularization induced by PLT. We compare the above diagnostics between the baseline training and +PLT. Across matched training budgets, PLT consistently yields a *smoother* and *more concentrated* trajectory in Wasserstein space:

- **Effective dimension:** decreases by 0.4% (paired t-test: $t(31) = 2.98$, $p = 0.007$), indicating slightly stronger concentration of policy iterates.
- **Trajectory curvature:** decreases by 9.9% ($t(31) = 3.12$, $p = 0.004$), suggesting that PLT substantially suppresses jagged, geometry-unaware jumps and promotes smoother policy evolution.
- **Local (intrinsic) dimension:** decreases by 4.7% ($t(31) = 0.93$, $p = 0.360$); while not statistically significant at $\alpha = 0.05$, the trend is directionally consistent with the above measures.

Statistical testing protocol. All tests use paired, two-sided t-tests across $n = 32$ paired runs (degrees of freedom 31), with significance level $\alpha = 0.05$. The resulting statistics are:

- Effective dimension: $t(31) = 2.98$, $p = 0.007$.
- Trajectory curvature: $t(31) = 3.12$, $p = 0.004$.
- Local dimension: $t(31) = 0.93$, $p = 0.360$.

Takeaway. Together, these diagnostics provide converging empirical evidence that (i) policy learning trajectories concentrate on a low-dimensional structure in Wasserstein space, and (ii) PLT acts as a Laplacian-style geometric regularizer that makes this structure *more coherent* (lower effective dimension) and the trajectory *smoother* (lower curvature). This supports the paper's key claim that one can *learn and exploit policy geometry from experience* rather than relying solely on parameter- or KL-scale control.

F.7 Additional Evaluation on Qwen3-4B

To examine whether the benefits of PLT extend beyond the GPT-family models, we evaluate Qwen3-4B on Long-Range Factual Recall using three independent seeds. This experiment extends the controlled recall task described in Section 5.2 and Appendix I.2.2. The task requires the model to recover the latest valid factual update under controlled query–evidence distances, overwrites, and distractors. We compare the baseline model with its PLT-enhanced counterpart.

As shown in Table 14, PLT improves Weighted Token Accuracy by 4.83 percentage points and Move Accuracy by 6.75 percentage points. It also reduces Final NLL from 3.085 to 2.053, with NLL reductions observed in all three seeds. These results extend the empirical evidence for PLT to Qwen3-4B in this controlled recall setting. The corresponding runtime and memory measurements are reported in Appendix G.3.

[3] We report mean $\pm$ s.d. over seeds and perform paired two-sided t-tests at $\alpha = 0.05$.

Table 14: Qwen3-4B results on Long-Range Factual Recall from three independent seeds. Accuracy changes are reported in percentage points (pp). The NLL change is an absolute difference. Bold indicates the better result for each metric.

Metric	Baseline	+PLT	Change
Weighted Token Accuracy ↑	39.86%	**44.69%**	+4.83 pp
Move Accuracy ↑	49.31%	**56.06%**	+6.75 pp
Final NLL ↓	3.085	**2.053**	−1.032

This additional evaluation covers Long-Range Factual Recall only. It does not establish performance on the other two stress tests at the Qwen3-4B scale or within a complete RLHF/RLVR pipeline. Such extensions require separate evaluation.

G Overhead of PLT: Memory and Computation

Scope. This appendix summarizes the additional *memory* and *compute* overhead introduced by Policy Laplacian Trace (PLT) relative to the base optimizer, separating (i) RL/MDP policies and (ii) LLMs-as-policies. We emphasize that PLT is modular: its overhead is controlled by the trace capacity, the stored representation, and the OT approximation used for fusion.

Empirical wall-clock overhead. To complement the asymptotic analysis, we also report a representative wall-clock comparison on Atari NameThisGame-v5 under the same training budget and hardware setting. The baseline PPO run takes 1d 18h 28m 31s, while PPO+PLT takes 1d 19h 49m 35s. This corresponds to an additional 1h 21m 04s, or a relative overhead of approximately 3.18%.

Table 15: **Representative wall-clock overhead on Atari NameThisGame-v5.** Both runs use the same environment budget and hardware configuration.

Method	Wall-clock time (NVIDIA 4060(8G))	Absolute overhead	Relative overhead
PPO Baseline	42h 28m 31s	–	–
PPO+PLT	43h 49m 35s	+1h 21m 04s	+3.18%

G.1 RL/MDP Setting

Storage. With trace capacity M, PLT stores (i) a state embedding $z(s) \in \mathbb{R}^{d_z}$ and (ii) a compact representation of the policy at that state (e.g., distribution parameters) of dimension d_ϕ. The asymptotic storage is

$$\mathcal{O}\big(M(d_z + d_\phi)\big), \qquad d_\phi = \begin{cases} 2d_a, & \text{continuous actions (e.g., Gaussian mean/scale)} \\ |\mathcal{A}|, & \text{discrete actions (probability vector).} \end{cases} \tag{43}$$

Example. For $M{=}10^4$, $d_z{=}64$, $d_a{=}6$ (continuous), the additional footprint is $10^4(64 + 2 \cdot 6) = 7.6 \times 10^5$ floats ≈ 3.0 MB with float32.

Compute. Let Q denote the number of queried states for which PLT fusion is enabled in one base update. For each query, PLT adds three operations: retrieval from the trace, local aggregation of retrieved policy representations, and a small Sinkhorn-based OT fusion step. With exact nearest-neighbor search, the additional cost per update is

$$\mathcal{O}\big(QMd_z + Qkd_\phi + Qm^2 I\big)\,, \tag{44}$$

where the three terms correspond to trace retrieval, local historical-policy construction, and entropic OT fusion, respectively. With approximate nearest-neighbor indexing or sparse fusion, the retrieval and fusion costs can be further amortized. In our Atari NameThisGame-v5 run, this theoretical overhead translates to only a 3.18% wall-clock increase over PPO, suggesting that the dominant training cost remains the base rollout and optimization pipeline rather than PLT fusion.

G.2 LLM Setting

Storage. For LLMs, naively storing full next-token distributions is prohibitive. If PLT stores N trace entries (e.g., $N \approx L \times K$ for sequence length L and K entries per position) and records both a hidden-state key of dimension d_h and a full vocabulary distribution of size V, the storage scales as

$$\mathcal{O}\big(N(d_h + V)\big), \tag{45}$$

which can reach multi-GB for typical $V \approx 50$ k. In practice, our implementation uses a *compressed* representation (e.g., a semantic embedding proxy) of dimension $d_{\text{sem}} \approx d_h$ instead of storing full distributions, reducing storage to

$$\mathcal{O}\big(N(d_h + d_{\text{sem}})\big). \tag{46}$$

Example (GPT-2 Small). With $L{=}1024$, $K{=}10$ so $N{=}10{,}240$, $d_h{=}768$, $V{=}50{,}257$: storing full distributions would be ≈ 2.1 GB (float32), whereas a compressed $(d_h{+}d_{\text{sem}}) \approx 1536$ representation is ≈ 63 MB.

Compute. Let B be the batch size, T the sequence length, and $Q \leq BT$ the number of tokens selected by the hardness gate for PLT fusion. In the worst case, $Q = BT$; in practice, sparse gating makes Q substantially smaller. For each selected token, PLT performs memory retrieval over hidden-state keys and fuses the retrieved distributions in a compressed semantic space. With exact retrieval, the additional cost is

$$\mathcal{O}(QNd_h + Qkd_{\text{sem}}), \tag{47}$$

where N is the number of memory entries, d_h is the hidden-state dimension, k is the number of retrieved neighbors, and d_{sem} is the compressed semantic-policy dimension. If PLT were applied to all tokens without sparse gating, this becomes $\mathcal{O}(BTNd_h + BTkd_{\text{sem}})$. Exact vocabulary-level OT would require a $V \times V$ transport cost and is therefore avoided; the semantic proxy reduces fusion from vocabulary-scale transport to $\mathcal{O}(kd_{\text{sem}})$ per selected token.

Practical overhead. In practice, the overhead is controlled by the trace size, query ratio, neighborhood size, and Sinkhorn parameters. For the RL setting, a representative Atari NAMETHISGAME-V5 run shows a wall-clock increase of only 3.18% over PPO. For LLMs, the overhead depends more strongly on the gating ratio $Q/(BT)$ and memory size N; compressed storage and sparse token-level fusion are therefore important for scalability.

Key implementation knobs. To balance performance and overhead, PLT exposes: (i) fixed-capacity FIFO or similarity-prioritized trace, (ii) compressed policy representations (semantic proxies), (iii) sparse retrieval/fusion via gating, and (iv) small-batch entropic OT with $m \in [8, 32]$.

Deployment note. PLT is designed to be optional at inference: one can disable fusion at test time to recover the base model's latency profile, while retaining PLT benefits during training.

G.3 EMPIRICAL OVERHEAD ON QWEN3-4B

We measure the computational overhead of PLT in the Qwen3-4B experiment described in Appendix F.7. Table 16 reports median training-step time, throughput, and peak GPU memory under the tested H100 configuration. Relative changes are computed with respect to the baseline.

Table 16: Runtime and GPU-memory measurements for Qwen3-4B under the tested H100 configuration. Bold indicates the better result for each metric.

Metric	Baseline	+PLT	Relative change
Median step time (ms) ↓	**314.21**	325.39	+3.56%
Throughput (tokens/s) ↑	**4,124.6**	3,982.9	−3.44%
Peak GPU memory (GB) ↓	**8.789**	9.362	+6.52%

PLT increases median step time from 314.21 to 325.39 ms. Throughput decreases from 4,124.6 to 3,982.9 tokens/s, while peak GPU memory increases from 8.789 to 9.362 GB. These measurements indicate modest additional training overhead under the tested configuration.

Memory scaling. We also measure retrieval and aggregation time at two trace capacities. As shown in Table 17, increasing the trace from 2,000 to 20,000 records changes the measured time from 2.093 to 2.324 ms. The trace containing 20,000 records requires approximately 217.9 MB of storage. This trace-storage measurement is distinct from the total peak GPU memory reported in Table 16.

Table 17: Retrieval and aggregation time at two trace capacities under the tested Qwen3-4B/H100 configuration.

Trace records	Retrieval/aggregation time (ms)
2,000	2.093
20,000	2.324

These measurements characterize the runtime and storage costs of the tested implementation and memory capacities. They do not imply constant-time retrieval as the trace grows. Together with Appendix F.7, they report the observed performance–overhead trade-off, but do not establish a performance advantage under an equal wall-clock training budget.

H THEORETICAL PROOF

Notation $\mathcal{S}$ denotes the state space and $\mathcal{A}$ the action space equipped with a norm $\|\cdot\|$. $\mathcal{P}_2(\mathcal{A})$ is the set of Borel probability measures on $\mathcal{A}$ with finite second moment. $W_2(\cdot,\cdot)$ is the 2-Wasserstein distance on $\mathcal{P}_2(\mathcal{A})$, and $W_{2,\varepsilon}$ is its entropic-regularized surrogate (e.g., Sinkhorn OT) with regularization strength $\varepsilon > 0$. A stochastic policy is written as $\pi(\cdot \mid s)$, and when parameterized we write $\pi_\theta(\cdot \mid s)$ with parameter $\theta \in \Theta \subset \mathbb{R}^d$. We use t to index optimization iterations / interaction steps, and i, j to index samples stored in a buffer. The visited (or replay-buffer) state set at time t is $\mathcal{S}_t = \{s_i\}_{i=1}^t$, and we abbreviate $\pi_i \coloneqq \pi(\cdot \mid s_i)$ (or $\pi_{\theta_i}(\cdot \mid s_i)$ when emphasizing parameters). A state representation for similarity is $\varphi : \mathcal{S} \to \mathbb{R}^m$, $\kappa(\cdot)$ is a kernel, $h > 0$ is its bandwidth, and k denotes the neighborhood size for kNN graphs. The experience-induced graph is $G_t = (V_t, E_t)$ with node set $V_t = \mathcal{S}_t$, edge set E_t, and symmetric nonnegative weights $w_{ij} = w_{ji} \geq 0$; $\mathcal{N}_k(s_i)$ is the k-nearest-neighbor set of node s_i in G_t. $\Lambda_s \subset \mathcal{P}_2(\mathcal{A})$ denotes the realizable policy set at state s. $\mathrm{KL}(\mu\|\nu)$ is the Kullback–Leibler divergence between measures μ and ν (assumed finite when used). $\lambda \geq 0$ is the geometric-regularization coefficient, and $\eta > 0$ is a step size. $\ell_t(\mu)$ denotes a task-driven local objective or surrogate loss functional evaluated at a candidate policy distribution μ. For a measurable map $F : \mathcal{A} \to \mathcal{A}$, $F_{\#}\mu$ denotes the pushforward measure. $\mathrm{Id} : \mathcal{A} \to \mathcal{A}$ is the identity map. For $\mu \in \mathcal{P}_2(\mathcal{A})$, $T_\mu\mathcal{P}_2(\mathcal{A})$ denotes the (Otto) tangent space used in Wasserstein first-variation calculus. $L^2(\mu)$ denotes square-integrable vector fields and $\langle u, v\rangle_{L^2(\mu)} \coloneqq \int u(a)^\top v(a)\, d\mu(a)$. We write $\mathrm{Log}_\mu(\nu)$ for the (possibly set-valued) Wasserstein logarithm direction; when the OT map $T_{\mu\to\nu}$ is unique, $\mathrm{Log}_\mu(\nu) = T_{\mu\to\nu} - \mathrm{Id}$.

H.1 EXPERIENCE-INDUCED GEOMETRY AND THE WASSERSTEIN POLICY OPERATOR

Scope and purpose This subsection fixes the mathematical objects used throughout the paper. The goal is to make precise (i) *what* the policy distributions live on, (ii) *what* geometric structure is induced by experience, and (iii) *how* the proposed update can be viewed as a proximal/geometric step. Proofs are deferred unless explicitly stated.

(see, e.g., (Ambrosio et al., 2005; Villani et al., 2008) for the first-variation formula and Otto calculus; (Cuturi, 2013; Peyré et al., 2019) for entropic OT smoothness and dual potentials).

Definition H.1 (Policy as a point in Wasserstein space)**.** Let $(\mathcal{A}, \|\cdot\|)$ be the action space and $\mathcal{P}_2(\mathcal{A})$ the set of Borel probability measures with finite second moment. A (stochastic) policy is a state-indexed family of measures

$$\pi(\cdot \mid s) \in \mathcal{P}_2(\mathcal{A}), \qquad s \in \mathcal{S}, \tag{48}$$

equipped with the 2-Wasserstein metric W_2 on $\mathcal{P}_2(\mathcal{A})$. Optionally, we use the entropic OT surrogate $W_{2,\varepsilon}$ as a computationally tractable approximation of W_2.

Assumption H.2 (Realizable policy family as a benign finite-dimensional set)**.** Let $\Theta \subset \mathbb{R}^d$ be a parameter space and $\{\pi_\theta(\cdot \mid s)\}_{\theta\in\Theta}$ a parameterized policy family. For each state s, define the realizable set

$$\Lambda_s \coloneqq \big\{\pi_\theta(\cdot \mid s) : \theta \in \Theta\big\} \subset \mathcal{P}_2(\mathcal{A}). \tag{49}$$

We assume Λ_s is locally well-behaved (e.g., a finite-dimensional C^1 embedded submanifold of $\mathcal{P}_2(\mathcal{A})$ or admits a locally smooth chart/retraction), so that local geodesic approximations and projections/retractions onto Λ_s are meaningful.

Definition H.3 (Experience-induced state graph)**.** Let $\mathcal{S}_t = \{s_i\}_{i=1}^t$ denote the set of visited states (or a replay buffer subset) and let $\varphi : \mathcal{S} \to \mathbb{R}^m$ be a state representation used for similarity. We build an undirected (typically kNN) graph $G_t = (V_t, E_t)$ with nodes $V_t = \mathcal{S}_t$ and symmetric weights

$$w_{ij} \geq 0, \qquad w_{ij} = 0 \text{ if } (i,j) \notin E_t, \qquad w_{ij} = w_{ji}, \tag{50}$$

e.g., $w_{ij} = \kappa(\|\varphi(s_i) - \varphi(s_j)\|/h)$ for a kernel κ and bandwidth h. We write $\mathcal{N}_k(s_i)$ for the neighborhood of s_i in G_t.

Definition H.4 (Experience-induced policy Dirichlet energy)**.** Associate each visited state s_i with its policy measure $\pi_i \coloneqq \pi(\cdot \mid s_i) \in \mathcal{P}_2(\mathcal{A})$. The (quadratic) memory/geometry energy is the Wasserstein graph Dirichlet energy

$$E_{\mathrm{mem}}(\{\pi_i\}_{i=1}^t) \coloneqq \frac{1}{2}\sum_{i,j=1}^{t} w_{ij}\, W_2^2(\pi_i, \pi_j). \tag{51}$$

When using entropic OT, we replace W_2 by $W_{2,\varepsilon}$. More generally, the p-Dirichlet version is

$$E_{\mathrm{mem}}^{(p)}(\{\pi_i\}) \coloneqq \frac{1}{p}\sum_{i,j} w_{ij}\, W_2^p(\pi_i,\pi_j), \qquad p \geq 1, \tag{52}$$

which induces a nonlinear (graph) p-Laplacian geometry.

Lemma H.5 (First variation induces a Wasserstein graph Laplacian)**.** *Let $\mathcal{A} \subset \mathbb{R}^d$ be convex and equip $\mathcal{P}_2(\mathcal{A})$ with the quadratic-cost 2-Wasserstein metric W_2 induced by $c(a,b) = \|a-b\|_2^2$. Consider the weighted Dirichlet energy on a collection of policies $\{\pi_\ell\}_{\ell=1}^n \subset \mathcal{P}_2(\mathcal{A})$:*

$$E_{\mathrm{mem}}(\{\pi_\ell\}_{\ell=1}^n) \;\coloneqq\; \frac{1}{2}\sum_{\ell,m=1}^{n} w_{\ell m}\, W_2^2(\pi_\ell,\pi_m), \qquad w_{\ell m} = w_{m\ell} \geq 0. \tag{53}$$

Fix a node i and assume:

(1) Let π_i be absolutely continuous w.r.t. Lebesgue measure on $\mathcal{A}$;

(2) for every j with $w_{ij} > 0$, the optimal transport from π_i to π_j is induced by an (a.e.) unique map $T_{i\to j} : \mathcal{A} \to \mathcal{A}$.

Define the (single-valued) Wasserstein logarithm direction

$$\log_{\pi_i}(\pi_j) \;\coloneqq\; T_{i\to j} - \mathrm{Id} \in L^2(\pi_i). \tag{54}$$

Let $v \in T_{\pi_i}\mathcal{P}_2(\mathcal{A})$ and consider the perturbation $\pi_i^\tau \coloneqq (\mathrm{Id}+\tau v)_{\#}\pi_i$ for τ small.

Then the directional derivative of E_{mem} along v exists and satisfies

$$\frac{d}{d\tau}\Big|_{\tau=0} E_{\mathrm{mem}}(\pi_1,\dots,\pi_i^\tau,\dots,\pi_n) = -\Big\langle \sum_{j=1}^{n} w_{ij}\, \log_{\pi_i}(\pi_j),\, v \Big\rangle_{L^2(\pi_i)}. \tag{55}$$

Equivalently, the Wasserstein–Riemannian (Otto) gradient of E_{mem} with respect to π_i is

$$\nabla_{\pi_i}^{W_2} E_{\mathrm{mem}} = -\sum_{j=1}^{n} w_{ij}\, \log_{\pi_i}(\pi_j) = \sum_{j=1}^{n} w_{ij}\,(\mathrm{Id} - T_{i\to j}). \tag{56}$$

Proof. Write $E_{\mathrm{mem}} = \frac{1}{2}\sum_{\ell,m} w_{\ell m} W_2^2(\pi_\ell,\pi_m)$ and differentiate only the terms that depend on π_i. Using symmetry $w_{\ell m} = w_{m\ell}$ and $W_2(\pi_\ell,\pi_m) = W_2(\pi_m,\pi_\ell)$, we can rewrite the π_i-dependent part as

$$E_{\mathrm{mem}}(\pi_1,\dots,\pi_i,\dots,\pi_n) = \sum_{j=1}^{n} w_{ij}\, W_2^2(\pi_i,\pi_j) \;+\; (\text{terms independent of } \pi_i). \tag{57}$$

Hence,

$$\frac{d}{d\tau}\Big|_{\tau=0} E_{\mathrm{mem}}(\pi_1,\dots,\pi_i^\tau,\dots,\pi_n) = \sum_{j=1}^{n} w_{ij}\;\; \frac{d}{d\tau}\Big|_{\tau=0} W_2^2(\pi_i^\tau,\pi_j). \tag{58}$$

We now invoke the standard Otto-calculus first-variation identity for the squared 2-Wasserstein distance (see, e.g., (, Ch. 8–10) (Villani et al., 2008)): under Assumptions (1)–(2), the functional $\mu \mapsto \frac{1}{2}W_2^2(\mu,\nu)$ has Wasserstein (Otto) gradient at μ equal to

$$\nabla_{\mu}^{W_2}\left(\frac{1}{2}W_2^2(\mu,\nu)\right) = -\,\log_\mu(\nu) = \mathrm{Id} - T_{\mu\to\nu}, \tag{59}$$

and consequently, for the perturbation $\mu^\tau = (\mathrm{Id}+\tau v)_{\#}\mu$,

$$\frac{d}{d\tau}\Big|_{\tau=0}\left(\frac{1}{2}W_2^2(\mu^\tau,\nu)\right) = \big\langle \nabla_\mu^{W_2}\big(\tfrac{1}{2}W_2^2(\mu,\nu)\big)\,,\, v\big\rangle_{L^2(\mu)} = -\langle \log_\mu(\nu),\, v\rangle_{L^2(\mu)}. \tag{60}$$

Applying equation 60 with $\mu = \pi_i$ and $\nu = \pi_j$ gives

$$\frac{d}{d\tau}\Big|_{\tau=0} W_2^2(\pi_i^\tau, \pi_j) = -2\langle \log_{\pi_i}(\pi_j),\, v\rangle_{L^2(\pi_i)}. \tag{61}$$

Substituting equation 61 into equation 58 yields

$$\frac{d}{d\tau}\Big|_{\tau=0} E_{\mathrm{mem}}(\pi_1, \ldots, \pi_i^\tau, \ldots, \pi_n) = -2\Big\langle \sum_{j=1}^{n} w_{ij}\ \log_{\pi_i}(\pi_j),\, v \Big\rangle_{L^2(\pi_i)}. \tag{62}$$

Finally, note that our definition of E_{mem} contains a prefactor $\frac{1}{2}$, while the π_i-dependent part in equation 57 contributes two symmetric copies (the (i, j) and (j, i) terms). These cancel the factor 2 above, yielding exactly

$$\frac{d}{d\tau}\Big|_{\tau=0} E_{\mathrm{mem}}(\pi_1, \ldots, \pi_i^\tau, \ldots, \pi_n) = -\Big\langle \sum_{j=1}^{n} w_{ij}\ \log_{\pi_i}(\pi_j),\, v \Big\rangle_{L^2(\pi_i)}. \tag{63}$$

By the Riesz representation of the differential in the Otto metric, this identifies the Wasserstein gradient as $\nabla^{W_2}_{\pi_i} E_{\mathrm{mem}} = -\sum_j w_{ij} \log_{\pi_i}(\pi_j) = \sum_j w_{ij}(\mathrm{Id} - T_{i\to j})$. □

Remark H.6 (Non-uniqueness and discrete policies: subgradients / entropic smoothing). If the OT map from π_i to π_j is not unique, the Wasserstein logarithmic direction becomes set-valued (equivalently, the optimal coupling may be non-unique), so the Otto gradient in Lemma H.5 should be interpreted in a subdifferential sense (Ambrosio et al., 2005; Divol et al., 2025). Let $\partial^{W_2}_{\pi_i}$ denote the metric (Wasserstein) subdifferential. Then one can replace Lemma H.5 by the inclusion

$$\partial^{W_2}_{\pi_i} E_{\mathrm{mem}} \ \ni\ -\sum_{j=1}^{n} w_{ij}\, \xi_{i\to j}, \qquad \xi_{i\to j} \in \log_{\pi_i}(\pi_j), \tag{64}$$

i.e., any selection of optimal couplings (or optimal maps when they exist) yields a valid element of the subdifferential and hence a valid descent direction for the Wasserstein gradient flow.

For discrete policies (finite action space), a convenient alternative is to replace W_2^2 by the entropic OT surrogate $W^2_{2,\varepsilon}$ with $\varepsilon > 0$. In a discrete action space, $W^2_{2,\varepsilon}$ is everywhere differentiable with respect to the probability vector on the simplex, and its gradient is given by the (centered) Sinkhorn dual potentials returned by the Sinkhorn algorithm (equivalently, by the gradient of the entropic OT dual objective with respect to the source marginal) (Cuturi, 2013; Peyré et al., 2019). Consequently, the entropic energy

$$E^{\varepsilon}_{\mathrm{mem}}(\{\pi_\ell\}) \ \coloneqq\ \frac{1}{2} \sum_{\ell,m=1}^{n} w_{\ell m}\, W^2_{2,\varepsilon}(\pi_\ell, \pi_m) \tag{65}$$

is differentiable on the simplex, and the resulting "Laplacian force" at node i is simply the sum of the neighbor-wise entropic OT gradients:

$$\nabla_{\pi_i} E^{\varepsilon}_{\mathrm{mem}} = \sum_{j=1}^{n} w_{ij}\, \nabla_{\pi_i}\left(\frac{1}{2}\, W^2_{2,\varepsilon}(\pi_i, \pi_j)\right), \tag{66}$$

which directly matches the gradients implicitly computed by our Sinkhorn-based OT-fuse implementation.

Definition H.7 (Policy Laplacian Trace (PLT) as an update map)**.** Fix a time t and a query state s_t with current policy $\pi_t(\cdot \mid s_t)$. Let $\mathcal{N}_k(s_t)$ be its neighborhood in the experience-induced graph, with corresponding neighbor policies $\{\pi_j(\cdot \mid s_j)\}_{j\in\mathcal{N}_k(s_t)}$ and weights $\{w_{tj}\}$. Given a task-driven local signal denoted by $\ell_t(\cdot)$, define the fused/updated policy distribution as

$$\pi_t^{\mathrm{fuse}} \ \in\ \arg\min_{\mu\in\Lambda_{s_t}} \Big\{ \ell_t(\mu) \ +\ \frac{\lambda}{2} \sum_{j\in\mathcal{N}_k(s_t)} w_{tj}\, W_2^2(\mu, \pi_j) \ +\ \frac{1}{\eta}\, \mathrm{KL}(\mu\|\pi_t) \Big\}. \tag{67}$$

We define the operator

$$\mathrm{PLT}_t :\ (\pi_t, \{\pi_j\}_{j\in\mathcal{N}_k(s_t)}, \ell_t) \ \mapsto\ \pi_t^{\mathrm{fuse}}. \tag{68}$$

Lemma H.8 (PLT is the implicit Euler (proximal) step of a composite geometric energy)**.** *Fix an iteration t and a state s_t. Let Λ_{s_t} be a nonempty, closed, convex subset of the probability simplex Δ^{m-1} (finite action space with m actions).*[4] *Let $\pi_t \in \Lambda_{s_t}$ be the current policy at s_t.*

Let $\ell_t : \Lambda_{s_t} \to \mathbb{R} \cup \{+\infty\}$ be proper, lower semicontinuous, and convex. Let $\{(\pi_i, w_i)\}_{i\in\mathcal{N}_t}$ be the retrieved neighbour policies with weights $w_i \geq 0$ and $\sum_{i\in\mathcal{N}_t} w_i = 1$. Let $d : \Lambda_{s_t} \times \Lambda_{s_t} \to \mathbb{R}_+$ be a metric such that the map $\pi \mapsto \frac{1}{2}d^2(\pi, \pi_i)$ is convex on Λ_{s_t} for each $i \in \mathcal{N}_t$ (e.g., $d = W_{2,\varepsilon}$ with fixed ground cost and $\varepsilon > 0$). Define the memory potential

$$\Phi_t(\pi) \;\coloneqq\; \frac{1}{2}\sum_{i\in\mathcal{N}_t} w_i\, d^2(\pi, \pi_i), \qquad \mathcal{J}_t(\pi) \;\coloneqq\; \ell_t(\pi) + \lambda\,\Phi_t(\pi),$$

for some $\lambda \geq 0$.

For any stepsize $\eta > 0$, define the PLT update as the unique minimizer

$$\pi_{t+1} \;\coloneqq\; \arg\min_{\pi\in\Lambda_{s_t}} \Big\{\mathcal{J}_t(\pi) \;+\; \frac{1}{\eta}KL(\pi\|\pi_t)\Big\}. \tag{69}$$

Then:

1. ***(Existence and uniqueness)*** *The minimization problem equation 69 admits a unique solution $\pi_{t+1} \in \Lambda_{s_t}$.*
2. ***(Implicit Euler / proximal step)*** *π_{t+1} is exactly the implicit Euler step of the D_{KL}-gradient flow of $\mathcal{J}_t$ on Λ_{s_t}, i.e., it satisfies the first-order optimality condition*

$$0 \in \partial\mathcal{J}_t(\pi_{t+1}) \;+\; \frac{1}{\eta}\,\partial_\pi D_{\mathrm{KL}}(\pi\|\pi_t)\big|_{\pi=\pi_{t+1}} \;+\; N_{\Lambda_{s_t}}(\pi_{t+1}), \tag{70}$$

   where $N_{\Lambda_{s_t}}(\cdot)$ denotes the normal cone of Λ_{s_t}.

3. ***(Trust region in distribution space)*** *Moreover,*

$$\mathcal{J}_t(\pi_{t+1}) + \frac{1}{\eta}D_{\mathrm{KL}}(\pi_{t+1}\|\pi_t) \;\leq\; \mathcal{J}_t(\pi_t), \tag{71}$$

   so the step enforces a D_{KL}-proximal trust region while decreasing the composite energy.

Proposition H.9 (Cumulative W_2 drift control)**.** *Fix a state s and let $\Lambda_s \subset \mathcal{P}_2(\mathcal{A})$ satisfy Hypothesis H.2 with diameter $D < \infty$, sectional curvatures bounded in $[-\kappa_{\min}, \kappa_{\max}]$, and intrinsic dimension d_s. Suppose the PLT update (Eq. 15) is run for T steps with step size η small enough that each π_{t+1} remains within the injectivity radius of π_t on Λ_s. Then there exists a constant $C = C(D, \kappa_{\max}) > 0$ such that*

$$\sum_{t=0}^{T-1} W_2^2(\pi_{t+1}, \pi_t) \;\leq\; C\,\eta\,\mathcal{J}_0(\pi_0) \;+\; C\,\eta\,\lambda\sum_{t=1}^{T-1}\big|\Phi_t(\pi_t) - \Phi_{t-1}(\pi_t)\big| \;+\; C\,\eta\sum_{t=1}^{T-1}\big|\ell_t(\pi_t) - \ell_{t-1}(\pi_t)\big|.$$

In particular, if the neighbourhood structure evolves slowly—i.e., $\sum_{t=1}^{T-1}|\Phi_t(\pi_t) - \Phi_{t-1}(\pi_t)| = o(T)$ and the task loss is Lipschitz with bounded gradient along the trajectory—then

$$\sum_{t=0}^{T-1} W_2^2(\pi_{t+1}, \pi_t) \;\leq\; C\,\eta\,\mathcal{J}_0(\pi_0) + o(\eta\,T).$$

Proof sketch. The argument has three steps.

Step 1: Local KL–W_2 comparability. By Hypothesis H.2, the visited policies $\{\pi_t\}_{t=0}^T$ lie on a compact C^2 submanifold $\Lambda_s \subset \mathcal{P}_2(\mathcal{A})$. On any compact Riemannian manifold, the Fisher–Riemannian metric g_F and the metric g_W induced by the ambient W_2 geometry are uniformly equivalent (Amari, 2016; Ay et al., 2017): there exists $c > 0$ such that $g_W \leq c\,g_F$ pointwise on Λ_s. Since KL is locally the

[4] For Gaussian policies, replace Δ^{m-1} by the parameter manifold and interpret KL as a Bregman divergence induced by the log-partition; the statement remains identical.

squared Fisher geodesic distance, $\mathrm{KL}(\mu\|\nu) = \frac{1}{2}\|v\|^2_{g_F} + O(\|v\|^3_{g_F})$ where v is the tangent vector from μ to ν, the injectivity-radius assumption ensures the cubic remainder is dominated, giving

$$W_2^2(\pi_{t+1}, \pi_t) \;\le\; c\,\mathrm{KL}(\pi_{t+1}\|\pi_t) \qquad \text{for all } t.$$

Step 2: Per-step descent. Lemma H.8 gives $\mathrm{KL}(\pi_{t+1}\|\pi_t) \le \eta\,[\mathcal{J}_t(\pi_t) - \mathcal{J}_t(\pi_{t+1})]$. Combining with equation H.1,

$$W_2^2(\pi_{t+1}, \pi_t) \;\le\; c\,\eta\,[\mathcal{J}_t(\pi_t) - \mathcal{J}_t(\pi_{t+1})].$$

Step 3: Summation and neighbourhood drift. Summing equation H.1 over $t = 0, \dots, T-1$:

$$\begin{aligned}
\sum_{t=0}^{T-1} W_2^2(\pi_{t+1}, \pi_t) &\le c\,\eta \sum_{t=0}^{T-1} [\mathcal{J}_t(\pi_t) - \mathcal{J}_t(\pi_{t+1})] \\
&= c\,\eta\Big[\mathcal{J}_0(\pi_0) - \mathcal{J}_{T-1}(\pi_T) + \sum_{t=1}^{T-1} \big(\mathcal{J}_t(\pi_t) - \mathcal{J}_{t-1}(\pi_t)\big)\Big] \qquad (72) \\
&\le c\,\eta\,\mathcal{J}_0(\pi_0) + c\,\eta \sum_{t=1}^{T-1} \big|\mathcal{J}_t(\pi_t) - \mathcal{J}_{t-1}(\pi_t)\big|,
\end{aligned}$$

where equation 72 follows from re-indexing the second sum. Expanding $\mathcal{J}_t = \ell_t + \lambda\Phi_t$ and applying the triangle inequality yields equation H.9. The simplified bound equation H.9 follows when the neighbourhood drift terms are $o(T)$, which holds when the PLT buffer is sufficiently large and retrieval is stable. □

Corollary H.10 (Small-step expansion: task drift + Laplacian-type smoothing)**.** *Assume in addition that Λ_{s_t} is a smooth Riemannian manifold endowed with a distance d such that, on a geodesically convex neighbourhood $\mathcal{U} \subset \Lambda_{s_t}$ containing π_t, each $\pi_i \in \mathcal{U}$ is connected to π by a unique minimizing geodesic and*

$$\nabla_\pi\Big(\frac{1}{2}d^2(\pi, \pi_i)\Big) \;=\; -\log_\pi(\pi_i), \qquad \forall \pi \in \mathcal{U}, \tag{73}$$

where $\log_\pi(\pi_i) \in T_\pi\Lambda_{s_t}$ denotes the Riemannian logarithm map. Assume ℓ_t is C^1 on $\mathcal{U}$ and $\pi_{t+1} \in \mathcal{U}$.

Then π_{t+1} satisfies the implicit equation

$$\frac{1}{\eta}\,\mathrm{grad}_{D_{\mathrm{KL}}} D_{\mathrm{KL}}(\cdot\|\pi_t)\big|_{\pi_{t+1}} \;+\; \mathrm{grad}_{D_{\mathrm{KL}}}\ell_t(\pi_{t+1}) \;-\; \lambda \sum_{i\in\mathcal{N}_t} w_i\,\log_{\pi_{t+1}}(\pi_i) \;=\; 0, \tag{74}$$

and, as $\eta \to 0$, admits the first-order expansion in $T_{\pi_t}\Lambda_{s_t}$:

$$\pi_{t+1} = \pi_t - \eta\Big(\mathrm{grad}_{D_{\mathrm{KL}}}\ell_t(\pi_t) \;-\; \lambda \sum_{i\in\mathcal{N}_t} w_i\,\log_{\pi_t}(\pi_i)\Big) + o(\eta). \tag{75}$$

In particular, the memory term $\sum_i w_i\,\log_{\pi_t}(\pi_i)$ is a Laplacian-type smoothing direction: it points from π_t towards the (weighted) barycentric neighbourhood in the geometry induced by d.

Remark (operator vs. storage). Definitions H.4–H.7 emphasize that memory enters learning through an *operator* (a variational update rule) rather than a particular storage mechanism: any buffer that provides $\{(s_i, \pi(\cdot \mid s_i))\}$ together with a graph construction yields the same mathematical object.

H.2 Metric Mismatch: KL Divergence and Wasserstein Distance

To motivate the choice of the Wasserstein metric for our analysis, we first demonstrate a fundamental limitation of the standard KL divergence trust region. Specifically, we show that constraining the KL divergence does not imply a bound on the Wasserstein-2 displacement, which is the geometric quantity governing the transport cost.

Lemma H.11 (KL does not control W_2 without additional structure)**.** *There is no universal constant $C > 0$ such that $W_2(\nu, \mu) \le C\sqrt{D_{\mathrm{KL}}(\nu\|\mu)}$ holds for all pairs $(\nu, \mu) \in \mathcal{P}_2(\mathbb{R}^d)$. In particular, one can construct a sequence of (Gaussian) measures with $D_{\mathrm{KL}}(\nu_n\|\mu_n) \to 0$ while $W_2(\nu_n, \mu_n) \to \infty$.*

Remark (When KL *does* control W_2). Such a control can hold under additional regularity/compactness assumptions, e.g., when the reference measure satisfies a T_2/log-Sobolev-type condition, or when supports / second moments are uniformly bounded, yielding bounds of the form $W_2^2(\nu,\mu) \lesssim D_{\mathrm{KL}}(\nu\|\mu)$ with problem-dependent constants. Our lemma rules out only a *structure-free* (universal) control.

Proof. We prove this by constructing an explicit counterexample using Gaussian distributions with shifting means and expanding variances. For each $n \in \mathbb{N}$, define:

$$P_n = \mathcal{N}(n, n^4), \qquad Q_n = \mathcal{N}(0, n^4). \tag{76}$$

Note that both distributions share the same variance $\sigma_n^2 = n^4$ but have means separated by $\mu_{P_n} - \mu_{Q_n} = n$.

1. KL divergence vanishes The KL divergence between two univariate Gaussians $P = \mathcal{N}(\mu_1, \sigma^2)$ and $Q = \mathcal{N}(\mu_2, \sigma^2)$ is given by the formula $D_{\mathrm{KL}}(P\|Q) = \frac{(\mu_1-\mu_2)^2}{2\sigma^2}$. Substituting the parameters for our sequence:

$$D_{\mathrm{KL}}(P_n\|Q_n) = \frac{(n-0)^2}{2\cdot n^4} = \frac{n^2}{2n^4} = \frac{1}{2n^2}. \tag{77}$$

Clearly, as $n \to \infty$,

$$D_{\mathrm{KL}}(P_n\|Q_n) \to 0. \tag{78}$$

2. Wasserstein distance diverges. The squared Wasserstein-2 distance between two Gaussians is given by $W_2^2(\mathcal{N}(\mu_1,\sigma_1^2),\mathcal{N}(\mu_2,\sigma_2^2)) = (\mu_1-\mu_2)^2 + (\sigma_1-\sigma_2)^2$. Since $\sigma_1 = \sigma_2 = n^2$, the variance term vanishes, and the distance is purely determined by the shift in means:

$$W_2^2(P_n, Q_n) = (n-0)^2 + (n^2-n^2)^2 = n^2. \tag{79}$$

Taking the square root, we have:

$$W_2(P_n, Q_n) = n. \tag{80}$$

Thus, as $n \to \infty$,

$$W_2(P_n, Q_n) \to +\infty. \tag{81}$$

Conclusion. Suppose, for the sake of contradiction, that a constant C exists such that $W_2(P,Q) \le C\sqrt{D_{\mathrm{KL}}(P\|Q)}$ for all pairs. Applying this to our sequence would imply:

$$n \le C\sqrt{\frac{1}{2n^2}} = \frac{C}{\sqrt{2}n} \implies n^2 \le \frac{C}{\sqrt{2}}. \tag{82}$$

This inequality clearly fails for sufficiently large n. The construction demonstrates that when the reference distribution Q_n becomes sufficiently "flat" (large variance), the KL divergence becomes insensitive to significant shifts in the mean, whereas the Wasserstein distance correctly captures the geometric displacement. □

H.3 Geometric Optimization Principle of OT-Fuse

We now provide a theoretical foundation for the OT-fuse module. While the fusion rule (constructing the mixture coupling) is computationally efficient, we show here that it formally constitutes an approximate proximal update step on the Wasserstein manifold. This characterizes OT-fuse not as a heuristic, but as a principled geometric optimization procedure.

Proposition H.12 (OT-Fuse as an explicit Euler step for the entropic Wasserstein gradient flow)**.** *Let $\mathcal{A} = \{1,\dots,m\}$ be a finite action set and let Δ^{m-1} denote the probability simplex. Fix $\varepsilon > 0$ and equip Δ^{m-1} with the entropic OT transport cost $W_{2,\varepsilon}^2(\cdot,\cdot)$ induced by a ground cost matrix $C \in \mathbb{R}_+^{m\times m}$. Let $\{\pi_i\}_{i\in\mathcal{N}}$ be retrieved neighbour policies with weights $w_i \ge 0$, $\sum_{i\in\mathcal{N}} w_i = 1$. Define the memory potential*

$$\Phi_t(\pi) \;:=\; \frac{1}{2}\sum_{i\in\mathcal{N}} w_i\, W_{2,\varepsilon}^2(\pi,\pi_i), \qquad \pi \in \Delta^{m-1}. \tag{83}$$

(i) Explicit Euler interpretation. *Assume $\pi_t \in \mathrm{int}(\Delta^{m-1})$. Then Φ_t is continuously differentiable on $\mathrm{int}(\Delta^{m-1})$ and the gradient flow*

$$\dot{\pi}(\tau) \;=\; -\nabla\Phi_t(\pi(\tau)) \tag{84}$$

is well-defined locally. Moreover, the OT-Fuse update

$$\pi_t^{\mathrm{fuse}} \;:=\; \pi_t \;-\; \tau \sum_{i\in\mathcal{N}} w_i \,\nabla_\pi\!\left(\frac{1}{2}W_{2,\varepsilon}^2(\pi,\pi_i)\right)\Big|_{\pi=\pi_t} \tag{85}$$

implements a single explicit Euler step *of equation 84 with stepsize $\tau > 0$.*

(ii) First-order approximation to the proximal point (implicit Euler). *Define the entropic Wasserstein proximal point of Φ_t at π_t by*

$$\pi^\star \;:=\; \arg\min_{\pi\in\Delta^{m-1}} \left\{\frac{1}{2\tau}W_{2,\varepsilon}^2(\pi,\pi_t) \;+\; \Phi_t(\pi)\right\}. \tag{86}$$

Assume in addition that $\nabla\Phi_t$ is locally Lipschitz in a neighbourhood of π_t (with constant L) and that the minimizer $\pi^\star$ exists in $\mathrm{int}(\Delta^{m-1})$. Then, for sufficiently small τ,

$$\|\pi_t^{\mathrm{fuse}} - \pi^\star\| \;\le\; c\,\tau^2, \tag{87}$$

for some constant c depending on L and local smoothness constants of $W_{2,\varepsilon}^2$. Consequently, π_t^{fuse} is a first-order approximation to the implicit proximal point $\pi^\star$.

(iii) Link to Sinkhorn dual potentials (algorithmic correspondence). *For each neighbour π_i, let $(f_{t\to i}^\varepsilon, g_{t\to i}^\varepsilon)$ be the optimal entropic OT dual potentials between (π_t,π_i). Then on $\mathrm{int}(\Delta^{m-1})$,*

$$\nabla_\pi\!\left(\frac{1}{2}W_{2,\varepsilon}^2(\pi,\pi_i)\right)\Big|_{\pi=\pi_t} \;=\; \mathrm{center}(f_{t\to i}^\varepsilon), \tag{88}$$

where $\mathrm{center}(\cdot)$ subtracts an additive constant to make the gradient lie in the tangent space of the simplex (sum of coordinates equal to zero). Thus, OT-Fuse can be implemented by computing Sinkhorn dual potentials for each pair (π_t,π_i) and taking their weighted sum, which matches the update rule equation 85.

Proof. **Step 1: Differentiability and the explicit Euler form** For $\varepsilon > 0$ on a finite action space, the entropic OT cost $W_{2,\varepsilon}^2(\cdot,\nu)$ is C^1 on $\mathrm{int}(\Delta^{m-1})$ (and in fact smooth) with respect to the source marginal; moreover its gradient is given by the optimal Sinkhorn dual potential, up to an additive constant (Peyré et al., 2019; Cuturi, 2013, see e.g. Ch. 4). Since Φ_t is a finite weighted sum of such terms, it is C^1 on $\mathrm{int}(\Delta^{m-1})$, and the gradient flow equation 84 is locally well-posed. By definition of Φ_t, we have

$$\nabla\Phi_t(\pi) = \sum_{i\in\mathcal{N}} w_i\,\nabla_\pi\!\left(\frac{1}{2}W_{2,\varepsilon}^2(\pi,\pi_i)\right), \tag{89}$$

so the explicit Euler discretization of equation 84 with step τ is exactly equation 85.

Step 2: Implicit Euler equals the proximal point The proximal point equation 86 is the standard implicit Euler discretization of the gradient flow associated with Φ_t under the metric induced by $W_{2,\varepsilon}$ (i.e., the proximal point algorithm / minimizing movement scheme). Its first-order optimality condition is

$$\frac{1}{\tau}\,\nabla_\pi\!\left(\frac{1}{2}W_{2,\varepsilon}^2(\pi,\pi_t)\right)\Big|_{\pi=\pi^\star} \;+\; \nabla\Phi_t(\pi^\star) \;=\; 0. \tag{90}$$

Step 3: Explicit vs. implicit: a second-order local discrepancy Assume $\nabla\Phi_t$ is L-Lipschitz near π_t and $\pi^\star$ stays in this neighbourhood. Using a Taylor expansion around π_t,

$$\nabla\Phi_t(\pi^\star) = \nabla\Phi_t(\pi_t) + O(\|\pi^\star - \pi_t\|). \tag{91}$$

Similarly, smoothness of $W_{2,\varepsilon}^2$ implies

$$\nabla_\pi\big(\tfrac{1}{2}W_{2,\varepsilon}^2(\pi,\pi_t)\big)\Big|_{\pi=\pi^\star} = (\pi^\star - \pi_t) + O(\|\pi^\star - \pi_t\|^2) \tag{92}$$

in local coordinates on the simplex.[5] Substituting these expansions into equation 90 yields

$$\frac{1}{\tau}(\pi^\star - \pi_t) + \nabla\Phi_t(\pi_t) = O(\tau), \tag{93}$$

so $\pi^\star = \pi_t - \tau\nabla\Phi_t(\pi_t) + O(\tau^2)$. Comparing with the explicit Euler step $\pi_t^{\text{fuse}} = \pi_t - \tau\nabla\Phi_t(\pi_t)$ gives equation 87.

Step 4: Sinkhorn dual potentials implement the gradients Finally, equation 88 is the standard sensitivity result for entropic OT: the derivative of the entropic OT objective with respect to the source marginal equals the optimal dual potential (up to a constant), hence after centering it yields a valid gradient in the simplex tangent space (Peyré et al., 2019; Cuturi, 2013). This establishes the algorithmic correspondence. □

H.4 Theoretical Stability Guarantee: Non-Expansiveness

A central concern in policy fusion is stability: can the fusion process amplify input noise or lead to mode collapse? We answer this negatively by proving that the memory potential is geodesically convex, rendering the OT-fuse operator a contraction mapping in the Wasserstein metric.

Proposition H.13 (Geodesic Convexity and Contraction)**.** *Let $\mathcal{P}_2(\mathcal{A})$ be the space of probability measures on $\mathbb{R}^d$ with finite second moments. Consider the memory potential $\Phi_t(\pi) = \frac{1}{2}\sum_i w_i W_2^2(\pi, \pi_i)$ with $\sum w_i = 1$.*

(i) ***1-Convexity****: The potential Φ_t is 1-geodesically strongly convex on $\mathcal{P}_2(\mathcal{A})$.*

(ii) ***Non-Expansiveness****: The proximal operator $\pi \mapsto \text{prox}_{\tau\Phi_t}^{W_2}(\pi)$ is a contraction on $(\mathcal{P}_2(\mathcal{A}), W_2)$ with Lipschitz constant $(1+\tau)^{-1}$. Specifically, for any two policies $\pi, \pi' \in \mathcal{P}_2(\mathcal{A})$:*

$$W_2\left(\text{prox}_{\tau\Phi_t}^{W_2}(\pi),\, \text{prox}_{\tau\Phi_t}^{W_2}(\pi')\right) \le \frac{1}{1+\tau} W_2(\pi, \pi'). \tag{94}$$

Proof. **Part (i): Strong Geodesic Convexity** The squared Wasserstein distance $\nu \mapsto W_2^2(\nu, \mu)$ is known to be 1-geodesically convex (or displacement convex) on $\mathcal{P}_2(\mathbb{R}^d)$ for any fixed μ. Since Φ_t is a convex combination of such functions ($\sum w_i = 1$), it preserves the convexity modulus. Thus, for any constant-speed geodesic $\gamma : [0,1] \to \mathcal{P}_2(\mathcal{A})$, the following inequality holds strictly unless γ is a constant curve:

$$\Phi_t(\gamma(s)) \le (1-s)\Phi_t(\gamma(0)) + s\Phi_t(\gamma(1)) - \frac{1}{2}s(1-s)W_2^2(\gamma(0), \gamma(1)). \tag{95}$$

Part (ii): Contraction of the Proximal Map Let $T(\pi) = \text{prox}_{\tau\Phi_t}^{W_2}(\pi)$. By definition, $u = T(\pi)$ minimizes the objective functional:

$$J_\pi(\nu) \coloneqq \frac{1}{2\tau}W_2^2(\nu, \pi) + \Phi_t(\nu). \tag{96}$$

Since $\frac{1}{2\tau}W_2^2(\cdot, \pi)$ is $\frac{1}{\tau}$-strongly convex and Φ_t is 1-strongly convex, the total objective J_π is σ-strongly convex with modulus $\sigma = 1 + \frac{1}{\tau}$.

Consider two inputs π, π' and their mapped outputs $u = T(\pi), u' = T(\pi')$. The first-order optimality condition for geodesically convex functionals (characterized by the vanishing Wasserstein subgradient) implies:

$$0 \in \partial J_\pi(u) \quad \text{and} \quad 0 \in \partial J_{\pi'}(u'). \tag{97}$$

Using the strong monotonicity of the subgradients of σ-strongly convex functions on Wasserstein space, we have:

$$\langle \nabla J_\pi(u) - \nabla J_\pi(u'),\, \text{Log}_u(u')\rangle_\pi \ge \sigma W_2^2(u, u'), \tag{98}$$

where the inner product is on the tangent bundle. Since $\nabla J_\pi(u) = 0$, this simplifies to:

$$\langle -\nabla J_\pi(u'),\, \text{Log}_u(u')\rangle \ge \sigma W_2^2(u, u'). \tag{99}$$

[5] On the simplex, this should be interpreted in the tangent space (sum-to-zero constraint); the centering operation removes the additive constant ambiguity of dual potentials.

Note that $\nabla J_\pi(\nu) = \frac{1}{\tau}\nabla(\frac{1}{2}W_2^2(\nu,\pi)) + \nabla\Phi_t(\nu)$. Substituting this into the inequality:

$$\langle -\frac{1}{\tau}\mathrm{Log}_{u'}(\pi) - \nabla\Phi_t(u'),\, \mathrm{Log}_u(u')\rangle \geq \sigma W_2^2(u,u'). \tag{100}$$

From the optimality of u' for $J_{\pi'}$, we know $\nabla\Phi_t(u') = -\frac{1}{\tau}\mathrm{Log}_{u'}(\pi')$. Substituting this back:

$$\frac{1}{\tau}\langle \mathrm{Log}_{u'}(\pi') - \mathrm{Log}_{u'}(\pi),\, \mathrm{Log}_u(u')\rangle \geq \sigma W_2^2(u,u'). \tag{101}$$

By the generalized Cauchy-Schwarz inequality on the tangent space:

$$\frac{1}{\tau}W_2(\pi,\pi')W_2(u,u') \geq \frac{1}{\tau}\langle \mathrm{Log}_{u'}(\pi') - \mathrm{Log}_{u'}(\pi),\, \mathrm{Log}_u(u')\rangle. \tag{102}$$

Combining the inequalities:

$$\frac{1}{\tau}W_2(\pi,\pi')W_2(u,u') \geq \left(1+\frac{1}{\tau}\right) W_2^2(u,u'). \tag{103}$$

Dividing by $W_2(u,u')$ (assuming $u \neq u'$) and multiplying by τ:

$$W_2(\pi,\pi') \geq (1+\tau)W_2(u,u') \implies W_2(u,u') \leq \frac{1}{1+\tau}W_2(\pi,\pi'). \tag{104}$$

□

Corollary H.14 (Robustness)**.** *Since $\tau > 0$ (and $\tau = 1$ in our implementation), the OT-fuse operator strictly contracts the Wasserstein distance between any perturbed policy and the ideal policy. This guarantees that (1) the fusion process is stable against input noise, and (2) recursive application of the module leads to exponential convergence towards the memory consensus, preventing distributional divergence.*

H.5 Discrete Dirichlet Energy and Laplacian Smoothing

To understand the global effect of the OT-fuse mechanism, we analyze it through the lens of functional analysis on graphs. We define a *discrete Dirichlet energy* over the graph of policies, which measures the total geometric smoothness of the policy field.

Proposition H.15 (Discrete Dirichlet Form Induces Wasserstein Graph Laplacian)**.** *Let $G = (V, E)$ be an undirected graph with vertices $V = \{1,\dots,n\}$ and symmetric edge weights $w_{ij} \geq 0$. Let $\{\pi_i\}_{i=1}^n \subset \mathcal{P}_2(\mathcal{A})$ be a collection of policy distributions over a convex action space $\mathcal{A} \subseteq \mathbb{R}^d$. Define the discrete Dirichlet energy on the graph of distributions:*

$$E_{\mathrm{mem}}(\pi_1,\dots,\pi_n) \coloneqq \frac{1}{4}\sum_{i,j} w_{ij}\, W_2^2(\pi_i,\pi_j). \tag{105}$$

Under the regularity assumptions of Lemma H.5 (absolute continuity and unique OT maps), the first variation of E_{mem} with respect to π_i satisfies:

$$\frac{\delta E_{\mathrm{mem}}}{\delta \pi_i} = -\sum_j w_{ij}\,\varphi_{ij} + C_i, \tag{106}$$

where φ_{ij} is the Kantorovich potential from π_i to π_j (i.e., $\nabla\varphi_{ij}$ is the optimal transport map T_{ij}), and C_i is a constant. Consequently, the Wasserstein gradient flow of E_{mem} is:

$$\partial_t \pi_i = \mathrm{div}\left(\pi_i \sum_j w_{ij}\,\mathrm{Log}_{\pi_i}(\pi_j)\right), \qquad i = 1,\dots,n, \tag{107}$$

where $\mathrm{Log}_{\pi_i}(\pi_j) = T_{ij} - \mathrm{id}$.

Moreover, the stationary points of (7) are exactly the discrete Wasserstein harmonic maps *satisfying the graph Laplacian condition:*

$$\sum_j w_{ij}\,\mathrm{Log}_{\pi_i}(\pi_j) = 0 \quad \textit{for all } i. \tag{108}$$

Equivalently, each π_i is a Wasserstein barycenter *of its neighbors with weights $\{w_{ij}\}_{j=1}^n$.*

Proof. We compute the first variation of E_{mem} with respect to a single π_i. Since E_{mem} is symmetric and $w_{ij} = w_{ji}$, we have:

$$\frac{\delta E_{\mathrm{mem}}}{\delta \pi_i} = \frac{1}{2}\sum_j w_{ij}\,\frac{\delta}{\delta \pi_i} W_2^2(\pi_i, \pi_j) + \frac{1}{2}\sum_j w_{ji}\,\frac{\delta}{\delta \pi_i} W_2^2(\pi_j, \pi_i). \tag{109}$$

By the symmetry of the Wasserstein distance, $W_2^2(\pi_j, \pi_i) = W_2^2(\pi_i, \pi_j)$, and the first variation of $W_2^2(\cdot, \pi_j)$ at π_i is well-known:

$$\frac{\delta}{\delta \pi_i}\left(\frac{1}{2} W_2^2(\pi_i, \pi_j)\right) = -\varphi_{ij}, \tag{110}$$

where φ_{ij} is a Kantorovich potential for the transport from π_i to π_j, determined up to an additive constant. Similarly,

$$\frac{\delta}{\delta \pi_i}\left(\frac{1}{2} W_2^2(\pi_j, \pi_i)\right) = -\psi_{ji}, \tag{111}$$

where ψ_{ji} is a potential for the transport from π_j to π_i. However, by duality, $\psi_{ji} = \varphi_{ij}^c$, the c-transform of φ_{ij}. Under the assumption of unique OT maps, we have $\varphi_{ij}^c = \varphi_{ij}$ up to a constant. Thus, both terms contribute equally, yielding:

$$\frac{\delta E_{\mathrm{mem}}}{\delta \pi_i} = -\sum_j w_{ij}\varphi_{ij} + C_i, \tag{112}$$

where C_i is a constant ensuring the normalization of the variation.

The Wasserstein gradient flow (in the Otto calculus) is then given by the continuity equation:

$$\partial_t \pi_i = \mathrm{div}\left(\pi_i \nabla \frac{\delta E_{\mathrm{mem}}}{\delta \pi_i}\right). \tag{113}$$

Substituting the expression for the first variation, and noting that $\nabla \varphi_{ij} = T_{ij}$ and $\mathrm{Log}_{\pi_i}(\pi_j) = T_{ij} - \mathrm{id}$, we obtain (7).

To see the barycenter interpretation, observe that the stationary condition $\partial_t \pi_i = 0$ implies $\sum_j w_{ij} \mathrm{Log}_{\pi_i}(\pi_j) = 0$ (in the tangent space at π_i). This is precisely the first-order condition for π_i being a barycenter of $\{\pi_j\}$ with weights $\{w_{ij}\}$ in the Wasserstein space Cuturi (2013). □

Remark H.16 (Laplacian smoothing of policy fields). Equation (7) can be viewed as a *nonlinear heat equation* on the graph of distributions. Indeed, linearizing around a uniform policy $\bar{\pi}$ (i.e., setting $\pi_i = \bar{\pi} + \varepsilon \rho_i$ with small ε), and using the Taylor expansion $\mathrm{Log}_{\pi_i}(\pi_j) \approx \nabla(\psi_j - \psi_i)$ for some potentials ψ_i, we recover the classical graph Laplacian acting on the potentials:

$$\partial_t \psi_i \approx \sum_j w_{ij}(\psi_j - \psi_i). \tag{114}$$

Thus, E_{mem} induces a smoothing flow that dissipates variations between neighboring policies, analogous to Laplacian smoothing of scalar functions on graphs.

Corollary H.17 (Stabilization via Dirichlet energy minimization)**.** *Fix a query node/state s_t with current policy $\pi_t \in \Lambda_{s_t}$ and neighbors $\{\pi_j\}_{j \in \mathcal{N}_k(s_t)}$ with weights $\{w_{tj}\}_{j \in \mathcal{N}_k(s_t)}$. Define the* local *Dirichlet (memory) energy on Λ_{s_t} by*

$$\Phi_t(\mu) \;\coloneqq\; \frac{\lambda}{2} \sum_{j \in \mathcal{N}_k(s_t)} w_{tj}\, W_{2,\varepsilon}^2(\mu, \pi_j). \tag{115}$$

Consider the (local) Wasserstein gradient flow

$$\partial_\tau \mu_\tau \;=\; -\nabla^{W_{2,\varepsilon}} \Phi_t(\mu_\tau), \qquad \mu_\tau \in \Lambda_{s_t}. \tag{116}$$

Then the implicit Euler discretization of equation 116 with step size $\tau > 0$ is given by the entropic Wasserstein proximal map

$$\mu^+ \;\in\; \arg\min_{\mu \in \Lambda_{s_t}} \left\{ \frac{1}{2\tau} W_{2,\varepsilon}^2(\mu, \pi_t) + \Phi_t(\mu) \right\}. \tag{117}$$

In particular, the OT-fuse update (with $\tau = 1$ in our implementation) can be interpreted as computing $\pi_t^{\mathrm{fuse}} \approx \mu^+$, i.e., one implicit Euler step that decreases the local Dirichlet energy. Moreover, by optimality of μ^+,

$$\Phi_t(\mu^+) + \frac{1}{2\tau} W_{2,\varepsilon}^2(\mu^+, \pi_t) \;\le\; \Phi_t(\pi_t), \tag{118}$$

so the fusion step is stabilizing: it cannot increase the local geometric energy and it stays within a controlled distributional displacement from π_t.

I Additional Experimental Details

I.1 PLT in RL: Environments Information and Evaluation Protocols

I.1.1 PPO and Atari Games

In Xuance, the Atari benchmark consists of a collection of Atari 2600 environments simulated by Stella and interfaced through the Arcade Learning Environment (ALE). In this paper, we conduct comparative experiments on 20 representative tasks, with several examples illustrated in the Figure 5. The evaluation reward definitions are environment-specific, and the detailed specifications can be found in AtariAge.



Figure 5: Atari Game Example: Donkey King, Frogger, Name This Game, Kangaroo

I.1.2 MAPPO and StarCraft II

The StarCraft Multi-Agent Challenge (SMAC) serves as WhiRL's benchmark environment for studying cooperative multi-agent reinforcement learning (MARL) algorithms. SMAC is built on StarCraft II, a real-time strategy game developed by Blizzard Entertainment, which provides a challenging and realistic testbed for cooperative control. In this paper, we conduct experiments on four combat maps, where performance is evaluated by the win rate, defined as the ability of the allied units to overcome the enemy forces.

I.2 PLT in LLM: Design Rationale and Evaluation Protocols

High-level goal. We treat an LLM as a conditional policy that maps an input x (prompt/state/context) to an action distribution over next tokens. Our LLM experiments are designed to stress three brittle failure modes of *LLMs-as-policies* that directly reflect the paper's claims: (i) **stability** under long-range, contradictory evidence, (ii) **robustness** under adversarial fact–rumor and credibility shifts, and (iii) **sample efficiency** when acquiring novel concepts without erasing existing ones. Across all experiments, we match training budgets and evaluation protocols between baselines and +PLT and report multi-seed statistics (mean $\pm$ std / 95% CI), so observed gains are unlikely to be artifacts of scaling or noise.

Our LLM evaluations are designed as controlled *stress tests* that isolate failure modes implied by the geometry-mismatch hypothesis, rather than to claim broad benchmark SOTA. **Counterfactual Trust** targets fact-verification and misinformation robustness under explicit fact–rumor conflicts and source-credibility shifts, aligning in spirit with established truthfulness/fact-checking evaluations such as *FEVER: A Large-scale Dataset for Fact Extraction and VERification* (Thorne et al., 2018) and *TruthfulQA: Measuring How Models Mimic Human Falsehoods* (Lin et al., 2022). **Long-Range Factual Recall** isolates distance-driven degradation in long-context use under controlled overwrites and distractors, complementing long-context benchmarks and analyses such as *LongBench: A Bilingual, Multitask Benchmark for Long Context Understanding* (Bai et al., 2024) and *Lost in the Middle: How Language Models Use Long Contexts* (Liu et al., 2024). Finally, **Few-Shot Learning** mirrors few-shot adaptation and continual-learning concerns by measuring novel-class acquisition under extreme scarcity while tracking base-class retention, connecting to in-context few-shot evaluation (Brown et al., 2020) and catastrophic-forgetting/continual-learning formulations (French, 1999). The advantage of these controlled tasks is that they hold confounds fixed (distance, conflict structure, and data scarcity), making them sensitive to update-trajectory stability rather than model scale alone.

I.2.1 COUNTERFACTUAL TRUST TASK

Motivation. This task targets **decision stability** and **robustness** under a common real-world pathology: the model must make a *retrieval-conditioned decision* when evidence is (i) long-range, (ii) partially counterfactual, and (iii) adversarially conflicting (fact–rumor), with an additional *source credibility shift* dimension. The design intentionally separates "can the model retrieve/use evidence" from "does the model keep stable preferences under conflict."

Task setup. Each example consists of (a) a query requiring a trust/choice decision, (b) an in-context narrative containing both official and non-official claims, and (c) a retrieved context block placed at a controlled long distance (e.g., a 500-token retrieval window). We construct conditions that vary whether the decisive evidence is (i) present only in retrieval, (ii) contradicted by a rumor, and (iii) associated with a high-credibility or low-credibility source.

Metrics and what they measure. We report four complementary metrics:

- **General Acc.** Overall decision accuracy across all evaluation conditions. *Measures end-to-end policy correctness.*
- **Retrieval Acc.** Accuracy on the subset where the correct decision *requires* using retrieved evidence (e.g., when decisive facts appear only in the retrieval block at long range). *Measures long-range evidence integration as an action policy.*
- **Conflict Acc.** Accuracy under explicit fact–rumor contradictions (adversarial conflicts). *Measures robustness/stability under contradictory evidence.*
- **Official Acc.** Accuracy (or preference rate) for selecting the official/high-credibility source when credibility is the determining factor. *Measures whether the policy maintains stable credibility priors under distribution shift.*

Scoring rule. Each example is mapped to a discrete decision (e.g., select A/B, trust official vs rumor, etc.). A prediction is correct iff it matches the ground-truth decision under that condition. For **Retrieval Acc.** and **Conflict Acc.**, we evaluate only the pre-specified subsets (retrieval-required vs conflict-present), so the metric isolates the intended failure mode rather than being diluted by easy cases.

Why this design supports our claims. If PLT improves **geometry-aware stability** rather than merely fitting spurious patterns, we should observe: (i) larger gains on **Retrieval Acc.** than on easy in-context cases, and (ii) improved **Conflict Acc.** and **Official Acc.** when evidence is contradictory or credibility-shifted. These are precisely conditions where "small" parameter-space updates often produce disproportionate behavioral flips, and where using historical policies as a regularizer should stabilize policy preferences.

I.2.2 LONG-RANGE FACTUAL RECALL (OVERWRITE & DISTRACTORS)

Motivation. This benchmark isolates **distance-driven degradation** (a stability failure) in a controlled setting: facts are repeatedly overwritten, and the model must retrieve the *latest* update at a known token distance under distractors. Unlike natural corpora, the overwrite structure gives an unambiguous ground-truth "final state" and makes failure modes attributable to long-range credit assignment rather than ambiguity.

Task setup. Each story contains at least two overwriteable variables (e.g., a moving object's location and a vault passcode). The final supporting update is inserted at controlled positions, inducing exact query–evidence token gaps. We optionally include distractors (irrelevant facts, misleading mentions, or outdated values) to stress robustness of late-evidence selection.

Metrics and what they measure. We score *answer spans only* (targets-only) to avoid conflating narrative modeling with factual retrieval:

- *Loss (Late)*: negative log-likelihood restricted to the final answer span when the decisive evidence is far from the query. *Measures whether late evidence is actually used.*

- *Move Acc / Pass. Acc*: exact-match accuracy for each overwritten variable. *Measures factual correctness under overwrite.*
- *Token Acc (W)*: length-weighted token-level accuracy on the answer span. *Measures partial correctness and robustness to answer length.*
- *NLL (Std)*: calibration/stability statistic across distance buckets (e.g., variance/dispersion of NLL). *Measures whether improvements come with unstable confidence or degraded calibration.*

Scoring rule and distance buckets. We group examples by query–evidence token distance (predefined buckets) and report both per-bucket and aggregate results. Targets-only scoring ensures improvements reflect *evidence retrieval and selection*, not generic language modeling.

Why this design supports our claims. A geometry-aware, experience-regularized update should reduce sharp behavioral drift that appears as "forgetting the latest overwrite" at long distance. Thus, the expected signature is lower *Loss (Late)* and higher accuracy at large token gaps, while maintaining (or improving) calibration (*NLL (Std)* not exploding). This directly operationalizes **stability** for LLM policies.

I.2.3 Few-Shot Learning (Base Retention + Novel Acquisition)

Motivation. Few-shot learning is a canonical **sample efficiency** test and also exposes **catastrophic forgetting**: we want fast acquisition of novel classes from extremely few examples *without* degrading performance on base classes.

Task setup. We use a controlled class-incremental protocol with 20 base classes (many labeled examples each) and 5 novel classes (few labeled examples each; e.g., 5-shot). Evaluation uses a mixed test set containing both base and novel classes, so improvements cannot come from simply shifting probability mass toward novel classes at the expense of base performance.

Metrics and what they measure.

- **Base Acc.**: accuracy on base classes after training with novel examples. *Measures retention / absence of catastrophic forgetting.*
- **Novel Acc.**: few-shot accuracy on novel classes. *Measures sample-efficient acquisition.*
- **Overall Acc.**: accuracy on the mixed test set. *Measures net utility under realistic mixture.*

Scoring rule. Predictions are correct iff the model outputs the correct class label (or canonicalized label string) under a fixed decoding scheme. We report multi-seed statistics and keep training budgets identical between baseline and +PLT.

Why this design supports our claims. If PLT is functioning as an experience-induced geometric regularizer on the policy trajectory, it should improve novel-class learning (higher **Novel Acc.**) *without* sacrificing the learned base geometry (stable **Base Acc.**). This is exactly the "sample efficiency without forgetting" signature.

J Experimental Setup and Compute.

The most experiments are run on a single NVIDIA RTX 3090 (24GB). A typical run takes approximately 2 hours end-to-end under our default settings, while the peak GPU memory footprint varies with the backbone and policy parameterization (e.g., network width/depth, action distribution head, and batch size). Unless otherwise specified, we choose model sizes to fit within the 24GB budget and adjust architecture and batch/sequence lengths accordingly; for larger backbones or higher-capacity variants commonly used in top-tier benchmarks, we reduce batch size and/or sequence length (and, when needed, use gradient accumulation) to ensure comparable optimization dynamics under the same hardware constraints.