# Revisiting Laplacian Representations for Value Function Approximation in Deep RL

**Priyesh Vijayan, Padideh Nouri, Rishav Rishav, Sarath Chandar, Yash Chandak, Mathieu Reymond, Samira E. Kahou, Doina Precup**

**Keywords:** Laplacian, Eigenvectors, Successor Representations, Proto-value networks, PQN, ALE

## Summary

Proto-value functions originally introduced Laplacian eigenvectors as an effective feature basis for learning value functions. While recent advances have extended Laplacian embeddings to high-dimensional observations—primarily leveraging them as intrinsic rewards for option discovery—they are typically used as auxiliary signals rather than as direct inputs for value estimation. In this work, we take a more direct approach by integrating learned Laplacian representations into the value function and empirically evaluating their effectiveness across 23 Atari games.

## Contribution(s)

1. We empirically demonstrate that Laplacian eigenvectors learned online can be useful features for model-free RL in high-dimensional, large-scale environments, such as ALE (Bellemare et al., 2013).
   **Context:** Prior model-free online RL work has relied on Laplacian features only in tabular settings or via offline pre-training, or treated them merely as auxiliary reward signals.
2. We show that state-action representations can provide benefit over state-only q-learning methods in ALE. We report the performance of methods based on state-action representations under a behavior policy and a uniform-random policy.
   **Context:** State–action representations have traditionally been confined to continuous or variable-action domains and remain underexplored in discrete-action settings.
3. We explain the contribution of different embeddings within our Fusion architecture using Shapley values (Shapley et al., 1953).
   **Context:** To assess the importance of Laplacian within our Fusion architecture, we use Shapley values as a principled metric to attribute the impact of each head in the Q-value prediction.

# Revisiting Laplacian Representations for Value Function Approximation in Deep RL

**Priyesh Vijayan**[1, 2†], **Padideh Nouri**[1,2], **Rishav Rishav**[2,3], **Sarath Chandar**[2,4,6], **Yash Chandak**[5], **Mathieu Reymond**[2,4], **Samira E. Kahou**[2,3,6], **Doina Precup**[1, 2, 6]
vijayanp@mila.quebec

[1]McGill University, [2]Mila - Quebec AI Institute, [3]University of Calgary,
[4]Polytechnique Montreal, [5]Cohere, [6] Canada CIFAR AI Chair

## Abstract

Proto-value functions (PVFs) (Mahadevan & Maggioni, 2007) introduced Laplacian embeddings as an effective feature basis for value-function approximation; however, their utility remained limited to small, fully known state spaces. Recent work has scaled Laplacian embeddings to high-dimensional inputs, using them for reward shaping and option discovery in goal-directed tasks, yet only as auxiliary signals, rather than directly using them as features for value functions. In this paper, we learn Laplacian eigenvectors online and employ them as features for Q-learning in 23 Atari games. We empirically demonstrate that these online–learned embeddings substantially improve model-free RL in large, high-dimensional domains. We demonstrate that enriching state representations with action embeddings yields additional gains under both behavior-policy and uniform-random policies. Additionally, we introduce the Fusion architecture, which augments the representation with useful inductive bias at the embedding-level. To assess the usefulness of each embedding used in the Fusion architecture, we use Shapley values analysis.

## 1 Introduction

Representation learning has become a cornerstone of modern machine learning, enabling models to extract compact, informative, and generalizable features from high-dimensional data (Bengio et al., 2013; LeCun et al., 2015). In supervised learning, well-structured representations have led to significant improvements in generalization under limited data. In Reinforcement Learning (RL), however, the challenge of representation learning is compounded by the sequential nature of decision making, sparse reward signals, and the non-i.i.d. structure of data collected from agent–environment interactions (Dulac-Arnold et al., 2021; Lake et al., 2017).

A critical open question in RL is how to learn representations that capture the underlying structure of the environment in a way that facilitates policy learning. Traditional value-based methods, such as Q-learning or actor-critic algorithms, rely heavily on reward signals to shape representations (Mnih et al., 2015; Schulman et al., 2017). However, in sparse or delayed reward settings, this can lead to inefficient or unstable learning. To address this, recent work has investigated auxiliary learning objectives—unsupervised or self-supervised tasks used to shape representations that are useful for control. These include tasks such as predictive modeling (Gelada et al., 2019), contrastive learning (Laskin et al., 2020), mutual information maximization (Laskin et al., 2022), and spectral learning (Machado et al., 2018b; Wu et al., 2019).

Among these, manifold learning techniques, particularly those based on spectral graph theory, offer a powerful inductive bias for capturing the geometry of the state space (Belkin & Niyogi, 2003; Coifman & Lafon, 2006; Chandak et al., 2023). These methods postulate that the high-dimensional observations in an environment lie on a lower-dimensional manifold with meaningful structure—e.g.,

connectivity, bottlenecks, or clusters of behaviorally similar states. By embedding the state space into a smooth, low-dimensional latent space that reflects these geometric properties, manifold learning can enhance exploration (Klissarov & Machado, 2023; Machado et al., 2018b), generalization (Adhikary et al., 2024), and reward propagation (Klissarov & Precup, 2020).

In this context, the Graph Laplacian provides a principled operator for encoding the geometry of a graph induced by the environment's transition dynamics. Laplacian-based methods, such as Proto-Value Functions (PVFs) (Mahadevan & Maggioni, 2007), utilize the eigenvectors of the Laplacian of the transition graph to derive smooth basis functions over the state space, which can be used for approximating value functions. These representations capture long-range dependencies and global connectivity patterns, and can be learned even in the absence of reward signals, making them especially useful as auxiliary tasks.

Despite their promise, Laplacian representations have traditionally relied on matrix-based eigendecompositions under a fixed, often uniform policy (Mahadevan & Maggioni, 2007). This limits their applicability in high-dimensional, non-stationary or online settings, which are typical of modern RL applications. Recent advances in learning eigenvectors for kernels (Wu et al., 2019; Wang et al., 2021; Gomez et al., 2024) address these limitations by enabling scalable, online learning of Laplacian embeddings using sampled transitions. This opens the door to learning spectral representations in a model-free RL setting, where the agent lacks access to a known transition model and must rely solely on its interactions with the environment.

Recent work has demonstrated the utility of Laplacian-based representations in deep RL (Klissarov & Machado, 2023; Farebrother et al., 2023). Klissarov & Machado (2023) introduced Deep Covering Eigenoptions (DCEO), which learn eigenvectors online for guiding exploratory option selection, but do not use them for value estimation. In contrast, Proto-Value Networks (Farebrother et al., 2023) employ Laplacian features as inputs to the value function, but only under a uniformly random policy and using offline data.

Building on this progress, our work unifies and extends these approaches by learning Laplacian representations for both the evolving behavior policy and the uniformly random policy, using data collected solely from the behavior policy. Unlike prior methods, we incorporate these representations directly into online value function approximation in a model-free setting. Specifically, we adopt the Parallelized Q-Networks (PQN) framework (Gallici et al., 2025), which enables efficient deep Q-learning without the need for a replay buffer or target networks. Within this framework, we demonstrate that Laplacian representations can be learned jointly with value estimates during training. This formulation broadens the applicability of Laplacian-based auxiliary objectives and provides a scalable method for learning structured representations in high-dimensional environments.

## 2 Background

We consider the model-free RL setting, where an agent interacts with the environment without access to its transition dynamics—neither given nor learned explicitly. While traditional model-free methods do not explicitly leverage the environment's transition structure, our goal is to extract useful structural information from interaction data in the form of temporal representations, which can be used to improve sample efficiency in value-based learning.

**Markov Decison Process:** Let $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma)$ be a Markov Decision Process (Puterman, 1994) with set of states $\mathcal{S}$, action set $\mathcal{A}$, a scalar reward function $\mathcal{R}$, probability transition function $\mathcal{P}$ and a discount factor $\gamma \in [0, 1)$. For simplicity, let $\mathcal{S}$ and $\mathcal{A}$ be finite, $R \in \mathbb{R}^{|\mathcal{S}|}$ be the reward vector and $\mathcal{P} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|\times|\mathcal{S}|}$ be the transition tensor, where $\mathcal{P}(s, a, s') = \Pr(s' \mid s, a)$, and let $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ be a policy.

Let $V_\pi(s) \in \mathbb{R}$ denote the value of a state associated with policy $\pi$ and it is defined as following:

$$V_\pi(s) = \mathbb{E}_{\pi,\mathcal{P}}\left[\sum_{k=0}^{\infty} \gamma^k R(S_{t+k+1}) | S_t = s\right]. \tag{1}$$

The goal of RL is to find the policy that maximizes the expected cumulative discounted reward, i.e., $\pi^* = \text{argmax}_\pi V_\pi(s_0)$, where $s_0$ follows the initial state distribution of the environment.

**Parallelized Q-network:** We define state-action values as Q-values, i.e., $Q_\pi(s,a) = \mathbb{E}_{\pi,\mathcal{P}}\left[\sum_{k=0}^{\infty}\gamma^k R(S_{t+k+1})|S_t = s, A_t = a\right]$. One approach to learn $Q_{\pi^*}$ is through the *Bellman equation* (Sutton & Barto, 2018), $Q(s_t,a_t) \leftarrow (1-\alpha)Q(s_t,a_t) + \alpha\delta$, where $\delta = r_t + \gamma \max_a Q(s_{t+1},a)$ is often referred to as the *temporal-difference target*. Parallelized Q-network (PQN, Gallici et al. 2025) combines parallelized environments, layer-normalized neural networks, and $\lambda$-returns to learn $Q_{\pi^*}$ in MDPs with large state-spaces. The agent interacts with I parallel environments for T timesteps, following an $\epsilon$-greedy policy, collecting trajectories $\tau^i = (s_t^i, a_t^i, r_t^i, s_{t+1}^i, \dots, s_{t+T}^i)$, and updating a neural network $Q_\psi$ parametrized by $\psi$, by minimizing $(Q_\psi(s_t,a_t) - [\![\delta_\psi]\!])^2$ over batches sampled from $\tau$. We define $[\![\cdot]\!]$ as the stop-gradient operator.

**Successor Matrix** The value function under a fixed policy $\pi$ can be expressed recursively in matrix form as:

$$\begin{aligned} V_\pi &= R + \gamma P_\pi V_\pi \\ V_\pi &= \underbrace{(I - \gamma P_\pi)^{-1}}_{M_{\pi,\gamma}} R \end{aligned} \tag{2}$$

where $M_{\pi,\gamma} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ is known as the *successor matrix* and $P_\pi(s'|s) = \sum_a \pi(a|s)P(s'|s,a)$ denotes the stochastic *state transition matrix* corresponding to policy $\pi$'s Markov chain. This formulation enables computation of the value function using only the immediate reward vector:

$$V(s) = \langle M_{\pi,\gamma}(s,:), R\rangle = \sum_{s'} M_{\pi,\gamma}(s,s')R(s') \tag{3}$$

In this expression, the row $M_{\pi,\gamma}(s,:) \in \mathbb{R}^{|\mathcal{S}|}$ serves as a rich state representation where the entries $M(s,s')$ encode the discounted expected number of visits to state $s'$ starting form state $s$. This follows from the expansion of the geometric series i.e $M_{\pi,\gamma}(s,s') = \sum_{k=0}^{\infty}\gamma^k P_\pi^t(s,s')$. However, computing and storing $M_{\pi,\gamma}$ becomes intractable in large or continuous state-spaces.

**Laplacian representations:** One approach to reflect the long-term dynamics encoded in $M_{\pi,\gamma}$ without explicitly constructing it is to leverage the spectral structure of the environment dynamics, via the transition matrix $P_\pi$. The Laplacian of $P_\pi$ is defined as $L_\pi = I - P_\pi$, and provides a graph-based view of the environment, where states are nodes and transitions define edges. Importantly, $P_\pi$, $L_\pi$, and $M_\pi$ share the same eigenvectors, though they differ in their eigenvalues (Stachenfeld et al., 2014). This spectral alignment enables the use of any of these matrices to derive meaningful low-dimensional representations of states. We focus on the eigenvectors of $L_\pi$ to construct state embeddings. When $P_\pi$ corresponds to an irreducible and aperiodic Markov chain, its spectral decomposition exists and reveals useful structural information about the environment. The eigenvectors associated with the smallest $d$ eigenvalues of the Laplacian yield smooth, global representations over the state space, capturing properties such as bottlenecks, clusters, and overall connectivity. These embeddings, denoted $\phi(s) \in \mathbb{R}^d$, are then used as input to a function approximator $f_a$ for estimating Q-values:

$$Q_\pi(s,a) = Q_\psi(\phi(s)). \tag{4}$$

**Online Laplacian representations:** In this work, we utilize Laplacian representations to serve as both state and state-action embeddings, in the spirit of Proto-Value Functions (PVFs), but applied to significantly larger state spaces with high-dimensional pixel observations. Recent advances (Deng et al., 2022; Wu et al., 2019) in spectral learning with neural networks have introduced optimization techniques that approximate eigenvectors of kernels using only sampled transitions, making them practical in environments where the full transition matrix is unknown or too large to store.

We specifically leverage recent advances (Wu et al., 2019; Wang et al., 2021; Gomez et al., 2024) built upon the graph drawing objective (Koren, 2003) to extract eigenvectors of the graph Laplacian

kernel $L$ as given below. The $d$ minimal values ($f_i, \forall i = 1, ..., d$) of the objective approximate the $d$ smallest orthonormal eigenvectors of $L$, where $\delta_{ij}$ is the Kronecker delta:

$$\begin{aligned} \underset{f \in \mathbb{R}^{d \times |S|}}{\text{minimize}} \quad & \sum_{i=1}^{d} f_i^T L f_i \\ \text{s.t.} \quad & f_i^T f_j = \delta_{ij}, \forall i, j = 1, ..., d. \end{aligned} \tag{5}$$

To handle settings where the state space $\mathcal{S}$ is continuous or uncountable (e.g., $\mathbb{R}^n$) and inputs are high-dimensional (e.g., raw pixels), we employ the Augmented Lagrangian Laplacian Objective (ALLO) (Gomez et al., 2024). ALLO generalizes the graph drawing objective by replacing the finite vectors $f_i$ with real-valued functions $f_i : \mathcal{S} \to \mathbb{R}$, making it suitable for function approximation via neural networks. The objective is:

$$\max_{\boldsymbol{\beta}} \min_{\mathbf{f} \in \mathbb{R}^{d \times |\mathcal{S}|}} \sum_{i=1}^{d} \langle \mathbf{f}_i, L\mathbf{f}_i \rangle + \sum_{m=1}^{d} \sum_{m=1}^{m} \beta_{mn} \left( \langle \mathbf{f}_m, [\![\mathbf{f}_n]\!] \rangle - \delta_{mn} \right) + b \sum_{m=1}^{d} \sum_{n=1}^{m} \left( \langle \mathbf{f}_m, [\![\mathbf{f}_n]\!] \rangle - \delta_{mn} \right)^2, \tag{6}$$

where b is the barrier coefficient and $\beta \in \mathbb{R}^{d(d+1)/2}$ is the lower-triangular matrix of dual variables, both penalizing deviation from orthonormality constraints (the second and third term in the loss). And the first term is a similarity constraint term, which forces nodes that are connected to be similar in the L2 space. The minimum of ALLO's objective uniquely corresponds to the exact eigenvectors, unlike its predecessors' works, which do not contain the eigenvectors as unique solutions, but only provide the subspace they span or correspond to the rotations of the eigenvectors.

Although originally designed for symmetric Laplacian matrices under fixed policies, we empirically evaluate the ALLO method in asymmetric settings, specifically on the ALE benchmark, where the agent's behavior policy, which provides data for learning the Laplacian embeddings, changes over time with learning.

## 3 Related work

Despite recent successes of RL, sample inefficiency remains a considerable challenge, especially for problems with large state-spaces such as raw pixel data. Learning state representations to improve sample efficiency has thus become an essential aspect of RL research (Tomar et al., 2023). Many self-supervised approaches aim to learn representations by implicitly modeling dynamics. One successful approach is through contrastive learning (Laskin et al., 2020; Schwarzer et al., 2021; Mazoure et al., 2020; Zheng et al., 2023), or using latent dynamic models (Zhang et al., 2018; Guo et al., 2020; Zhao et al., 2023). However, these methods do not explicitly leverage the global diffusion properties of the MDP's transition dynamics, which are central to Laplacian embeddings, and require careful design of the optimization dynamics to learn meaningful representations (Tang et al., 2023).

Alternatively, Successor features (SFs, Barreto et al. 2017) model the expected future occupancy of features under a policy, enabling generalization across tasks. However, Touati et al. (2023) investigate the zero-shot RL capacities of SFs when using various representation learning methods to construct the basic feature maps, which embeds the underlying states. They argue that SFs show inconsistent performance when the basic feature maps are derived from auto-encoders, inverse curiosity (Pathak et al., 2017), transition models, low-rank transition matrix (Ren et al., 2023), contrastive learning (Chen et al., 2020; Tian, 2022), or diversity (Eysenbach et al., 2018; Hansen et al., 2020). They highlight that the SFs are most effective when paired with representations based on Forward-Backward (FB) learning (Touati & Ollivier, 2021) and Laplacian eigenfunctions (Wu et al., 2019), the focus of our work.

Separately, Forward–Backward (FB) representations have emerged as an effective alternative to successor features, capturing richer dynamics by jointly modeling transitions in both temporal direc-

tions (Touati & Ollivier, 2021; Sun et al., 2025; Jeen et al., 2024). However, unlike Klissarov & Machado (2023) and our method, FB-based approaches typically rely on alternating phases of data collection and representation learning, which limits their applicability in fully online reinforcement learning settings.

Motivated by recent advances in spectral optimization using graph drawing objectives (Gomez et al., 2024), we investigate the feasibility of learning Laplacian eigenfunctions in an online, model-free RL setting where the *behavior policy is non-stationary*. Additionally, drawing inspiration from PVFs, we extend this framework to learn representations corresponding to a *uniform random policy* over the *state-action* space. These are obtained using importance sampling corrections applied to data collected under the non-uniform behavior policy, without requiring access to a uniformly random data collection policy or control over initial state distributions.

## 4 Methodology

In this section, we build on the Augmented Lagrangian Laplacian Objective (ALLO, Gomez et al. 2024) to define a *multi-hop graph drawing objective* for learning Laplacian representations online using batches of k-step transitions sampled from a nonstationary behavior policy. We then extend this formulation to the *state-action graph*, showing how the d smoothest eigenvectors of its Laplacian under a *uniform random policy* can be learned using data collected from an evolving policy without requiring a uniformly random data collection strategy.

**State Laplacian eigenvectors:** For this work, we adopt PQN (Gallici et al., 2025), using the partial trajectories $\{\tau^1_{t:t+T}, \ldots, \tau^I_{t:t+T}\}$ collected to update $Q_\psi$ to also learn Laplacian representations. Specifically, we define a $K$-hop Laplacian loss using pairs of states $(s_t, s_{t+k})$ from each environment trajectory, where $1 \leq k \leq K \leq T$. The first term of the graph drawing loss is defined over these temporal edges. We use $\{\tau^1_{t:t+T}, \ldots, \tau^I_{t:t+T}\}$ to perform one update to the Laplacian network's parameters $\theta$. The orthogonality constraints in ALLO (linear and quadratic penalties) are computed only for the initial state $s_t$ in each trajectory.

$$\min_\theta \sum_{k=1}^{K} (f_\theta(s_t) - f_\theta(s_{t+k}))^2 \tag{6}$$

**State-action Laplacian eigenvectors:** We next extend this formulation to the state-action space. Here, each node in the graph corresponds to a state-action pair, and the agent's policy defines the edge transitions. For an $\epsilon$-greedy behavior policy, the temporal edges connect $(s_t, a_t \sim \pi(s_t))$ to $(s_{t+1}, a_{t+1} \sim \pi(s_{t+1}))$. The similarity terms for the $\epsilon$-greedy is given below.

$$\min_\theta \sum_{k=1}^{K} (f_\theta(s_t, a_t) - f_\theta(s_{t+k}, a_{t+k}))^2 \tag{7}$$

tIn the uniformly random policy case, the Laplacian is defined over a graph where each state-action node $(s_t, a_t)$ is connected to all state-action nodes $\{(s_{t+1}, a)$ for every action $a \in \mathcal{A}$, with uniform edge weights $\}$. However, in practice, the data is collected under a non-uniform, non-stationary behavior policy $\mu$, such as $\epsilon$-greedy. Therefore, the loss must be corrected using importance sampling (Precup et al., 2000) to account for the discrepancy between the behavior policy and the uniform target policy. Given a $k$-step trajectory $(s_0, a_0, s_1, a_1, \ldots, s_k, a_k)$, the importance sampling weight to correct from the behavior policy $\mu$ to the uniform random policy $\pi_{\text{uni}}(a|s) = 1/|\mathcal{A}|$ is: $w_k = \prod_{j=0}^{k} \frac{\pi_{\text{uni}}(a_j|s_j)}{\mu(a_j|s_j)}$. The corrected similarity term for the uniformly random policy is:

$$\min_\theta \sum_{k=1}^{K} \frac{(1/|\mathcal{A}|)^{k+1}}{\prod_{j=0}^{k} p(s_j, a_j)} \left( f_\theta(s_i, a_i) - \frac{1}{|\mathcal{A}|} \sum_a f_\theta(s_{i+k}, a) \right)^2 \tag{8}$$

We implement the state-action embedding using Film (Perez et al., 2018) technique where the modulated embedding is defined as $f_{s,a} = \alpha e_s + \beta$, with $\alpha = f_1(e_a)$ and $\beta = f_2(e_a)$ where $e_a$ is a continuous action embedding and $e_s$ is a continuous state embedding. For illustrations of the models, refer to the supplementary material: Figure 3 shows the baseline PQN-S architecture; Figure 4 illustrates the state-action variant; and Figure 5 details the FiLM conditioning module.

**Laplacian-based models:** We explore two architectural approaches to integrate Laplacian eigenvectors with Q-learning while preventing the TD-error gradients from altering the Laplacian representation.

- **Laplacian-Only models** (see Appendix A, Figure 6): The Laplacian embeddings are used as input features to an MLP that predicts Q-values. This is a decoupled architecture, where only the MLP (Q-network) is updated during training; the Laplacian encoder remains fixed.
- **Fusion models** (see Appendix A, Figure 7): Both the Laplacian and PQN-derived embeddings are computed separately and concatenated before being passed through a shared MLP. In this case, only the PQN encoder is updated via TD-loss gradients, while the Laplacian encoder remains unchanged.

## 5 Experiments

We evaluate the impact of learning Laplacian state-action representations in an online setting. To isolate and understand the contribution of each component in our methodology, we design multiple ablation baselines. Our evaluation is conducted on 23 Atari games from the Atari Learning Environment (ALE) (Bellemare et al., 2013; Machado et al., 2018a), using five random seeds per game. This subset includes 10 games known to be representative of the full suite (Aitchison et al., 2023) and 13 additional games with diverse characteristics (e.g., stationary, dynamic, and progressive environments) as categorized in (Delfosse et al., 2024). Following standard practices in Atari benchmarks, we report human-normalized scores (HNS) and use the aggregation metrics from Agarwal et al. (2021), computed with the publicly available library rliable.

### 5.1 Methods

We evaluate three families of methods, all of which share the same convolutional encoder architecture applied to four stacked frames (Mnih et al., 2015):

- **PQN-only**: Standard Q-learning using the Parallelized Q-Network (PQN) framework without auxiliary representation learning. These methods are prefixed with `PQN:`. Models with doubled encoder capacity (for fair comparison against fusion models) are suffixed with `-2x`.
- **Laplacian-only**: Q-values are predicted directly from fixed Laplacian representations learned via the ALLO objective. These methods are prefixed with `L:`.
- **Fusion**: Combines PQN embeddings with Laplacian embeddings via feature concatenation. Only the PQN branch is updated with the TD error. These methods are prefixed with `F:`.

Each method is further categorized by whether it uses **state-only (S)** or **state-action (SA)** representations. Laplacian features are trained either under the **behavior policy (BP)** or using **uniform-policy alignment (UNI)** via importance sampling corrections.

### 5.2 Results

We present the aggregate results in Figure 1 and provide per-game performance in Table 1 of the supplementary material.

**Laplacian-Only models:** Overall, we observe that all Laplacian-only methods—S-BP, SA-BP, and SA-UNI—fail to reach human-level performance on average, though they achieve strong results in a subset of games. Unlike prior work that integrates representation learning as an auxiliary objective—allowing the learned embeddings to be shaped by the temporal-difference (TD) loss—we

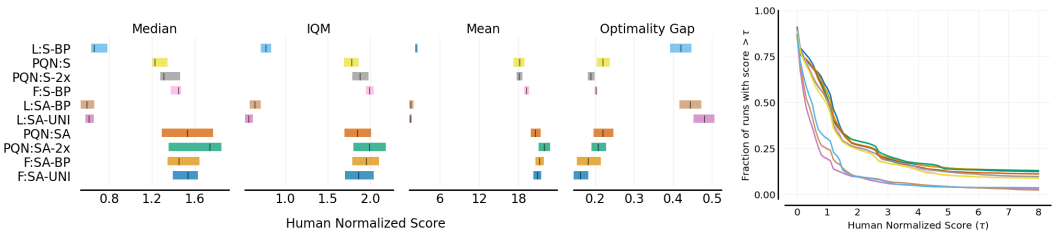


Figure 1: Comparison of PQN-only (PQN:), Laplacian-only (L:), and Fusion (F:) architectures on 23 Atari games with Human-Normalized Scores (HNS). State-action (SA) consistently outperforms state-only (S) models. Laplacian-only methods occasionally succeed but fail to reach human-level performance overall. Fusion models benefit from the inductive bias of Laplacian structure and are more robust, especially in games where PQN struggles, as indicated by the reduced optimality gap.

train Laplacian representations independently of the value function. While this separation preserves the spectral structure of the learned features, it limits their ability to adapt to task-specific reward signals. We hypothesize that the Laplacian embeddings may capture spurious or task-irrelevant temporal structures, especially under a tight dimensionality budget (512). This was supported empirically: increasing the embedding dimension led to improved performance in several games, yet in others, performance plateaued early or failed to improve regardless of dimensionality.

**PQN and Fusion Models:** In contrast to the Laplacian-only variants, methods based on PQN and Fusion architectures consistently learn high-performing policies across the Atari suite.

Among *state-based methods*, the fusion variant *F:S-BP outperforms both PQN:S and PQN:S-2x across all aggregate metrics*—including median, interquartile mean (IQM), mean, and optimality gap—demonstrating the utility of incorporating fixed Laplacian features even without direct gradient updates from the TD loss.

Across all model types, *state-action* variants (e.g., PQN:SA, F:SA-BP, F:SA-UNI) consistently outperform their state-only counterparts. This suggests that incorporating action-conditioning improves the quality of representation and downstream Q-values.

Among the top-performing models, PQN:SA, F:SA-BP, and F:SA-UNI achieve similar performance in terms of mean, median, and IQM. However, *F:SA-UNI*, which incorporates uniform policy Laplacian representations, *yields a noticeably lower optimality gap*. Examining the performance profiles in Figure 1 (right), F:SA-UNI achieves the highest proportion of runs with human-normalized score (HNS) exceeding thresholds $\tau \in [0, 1.5]$, indicating more robust performance. Notably, while PQN:SA occasionally fails to learn meaningful policies in a few games, F:SA-UNI avoids these failures. This suggests that the Laplacian component in the Fusion architecture helps mitigate policy collapse, likely by providing a stable temporal structure as an inductive bias.

Notably, this improvement is not simply due to increased network capacity: PQN-SA-2x, which matches F:SA-UNI in architecture and parameter size but lacks Laplacian input, does not match its optimality gap. Nevertheless, the added parameters in PQN-SA-2x do benefit learning, as evidenced by its strong median score, the highest among all evaluated methods.

### 5.2.1 Impact of the Laplacian representation in the Fusion architecture

To better understand how each representation contributes to decision-making in the Fusion model, we analyze the influence of each head in our two-branch Q-network using Shapley values (Shapley et al., 1953). Originally developed in cooperative game theory, Shapley values attribute credit to individual agents in a collaborative setting; here, we treat each embedding head in the Q-network as a "player" contributing to the predicted Q-value.

In our setting, the Fusion Q-network receives two inputs: (i) a representation from the policy-driven encoder (denoted PQN), which is trained end-to-end with TD loss, and (ii) a fixed representation

from a Laplacian encoder (denoted LAP), which is pretrained via ALLO and kept frozen during TD training. The combined embeddings are concatenated and passed through a shared Q-network head.

We define the Shapley contribution (ShapC) of the LAP head for a given state-action pair $(s, a)$ as:

$$\mathrm{SC}_{\mathrm{LAP}}(s,a) = \frac{1}{2}\Big([Q(e_{\mathrm{LAP}}, \emptyset; s, a) - Q(\emptyset, \emptyset; s, a)] + [Q(e_{\mathrm{LAP}}, e_{\mathrm{PQN}}; s, a) - Q(\emptyset, e_{\mathrm{PQN}}; s, a)]\Big) \tag{9}$$

where $\emptyset$ denotes the replacement of an embedding with its expected value over the dataset (i.e., $\mathbb{E}_\pi[e.]$). $\mathrm{SC}_{\mathrm{PQN}}(s, a)$ is defined analogously. This attribution quantifies how much each embedding contributes to the final Q-value relative to a baseline of state-independent (mean) embeddings. We report these values under the data distribution induced by the policy at the end of training.

We report the average ShapC across 100 evaluation episodes and time steps, aggregated over five random seeds per game, in the supplementary section (Appendix C, Figure 8). In the main text, we highlight four representative games in Figure 2. In these examples, the Laplacian embedding head consistently contributes more to the final Q-value prediction than the PQN head. This aligns with the quantitative results in Table 1 of Appendix B, where Laplacian-only models (Lap:) outperform PQN-only models (PQN:) in these games.

For instance, in `Pong`, both PQN and Laplacian representations are sufficient to solve the task individually. However, Shapley analysis reveals that the final Q-network prediction relies almost entirely on the Laplacian branch. In `Skiing`, `Tutankham` and `Bowling`, where Lap:* methods outperform their PQN:* counterparts, the Laplacian head provides a more influential and beneficial inductive bias for Q-value estimation.

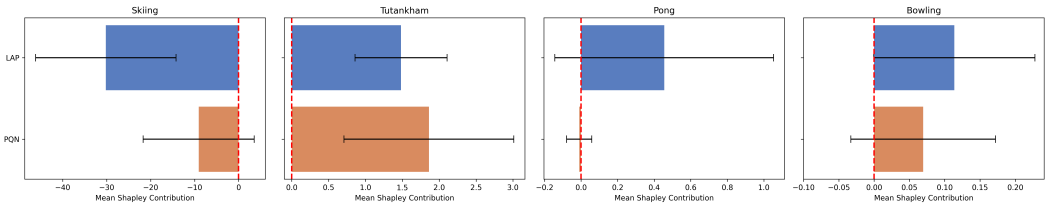


Figure 2: Average Shapley contribution (ShapC) of each head in the Fusion Q-network across four Atari games. A higher magnitude ShapC indicates stronger influence on the final Q-value prediction.

## 6 Conclusion and Future Work

In this work, we explored the integration of Laplacian-based representations into the value function approximation pipeline of reinforcement learning. Building on the foundation of Proto-Value Functions (PVFs), we demonstrated that spectral representations—learned online and over the state-action space—can serve as effective bases for deep Q-learning across a diverse set of environments in the Atari Learning Environment (ALE) benchmark. Our approach departs from prior work by learning these representations in a fully online manner, without relying on auxiliary rewards or cyclic training phases. Furthermore, we showed that Laplacian representations corresponding to both the behavior and uniform policies can be learned from interaction data alone and leveraged effectively in large-scale settings. Looking ahead, a promising direction for future work is to evaluate the impact of uniform-policy Laplacian representations in multi-task or transfer learning scenarios. Since these representations are task-independent, they are naturally suited for reuse across different tasks and environments, potentially offering improved sample efficiency and generalization.

## Acknowledgements

We thank Marlos C. Machado for his feedback throughout the project and also on this paper draft. We thank Diego Gomez, Martin Klissarov and Jesse Farebrother for participating in our discussions and providing feedback on an early version of the project.

## References


Sandesh Adhikary, Anqi Li, and Byron Boots. Beigemaps: behavioral eigenmaps for reinforcement learning from images. In *International Conference on Machine Learning*, 2024.

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in neural information processing systems*, 34:29304–29320, 2021.

Matthew Aitchison, Penny Sweetser, and Marcus Hutter. Atari-5: Distilling the arcade learning environment down to five games. In *International Conference on Machine Learning*, pp. 421–438. PMLR, 2023.

André Barreto, Will Dabney, Rémi Munos, Jonathan J Hunt, Tom Schaul, Hado P van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of artificial intelligence research*, 47: 253–279, 2013.

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.

Yash Chandak, Shantanu Thakoor, Zhaohan Daniel Guo, Yunhao Tang, Remi Munos, Will Dabney, and Diana L Borsa. Representations and exploration for deep reinforcement learning using singular value decomposition. In *International Conference on Machine Learning*, pp. 4009–4034. PMLR, 2023.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PmLR, 2020.

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

Quentin Delfosse, Patrick Schramowski, Martin Mundt, Alejandro Molina, and Kristian Kersting. Adaptive rational activations to boost deep reinforcement learning. In *International Conference on Learning Representations*, 2024.

Zhijie Deng, Jiaxin Shi, and Jun Zhu. Neuralef: Deconstructing kernels by deep neural networks. In *International Conference on Machine Learning*, pp. 4976–4992. PMLR, 2022.

Gabriel Dulac-Arnold, Nir Levine, Daniel J Mankowitz, Jerry Li, Cosmin Paduraru, Sven Gowal, and Todd Hester. Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. *Machine Learning*, 110(9):2419–2468, 2021.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *International Conference on Learning Representations*, 2018.

Jesse Farebrother, Joshua Greaves, Rishabh Agarwal, Charline Le Lan, Ross Goroshin, Pablo Samuel Castro, and Marc G Bellemare. Proto-value networks: Scaling representation learning with auxiliary tasks. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=oGDKSt9JrZi.

Matteo Gallici, Mattie Fellows, Benjamin Ellis, Bartomeu Pou, Ivan Masmitja, Jakob Nicolaus Foerster, and Mario Martin. Simplifying deep temporal difference learning. *ICLR*, 2025.

Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G Bellemare. Deepmdp: Learning continuous latent space models for representation learning. In *International conference on machine learning*, pp. 2170–2179. PMLR, 2019.

Diego Gomez, Michael Bowling, and Marlos C Machado. Proper laplacian representation learning. *International Conference on Learning Representations(ICLR)*, 2024.

Zhaohan Daniel Guo, Bernardo Avila Pires, Bilal Piot, Jean-Bastien Grill, Florent Altché, Rémi Munos, and Mohammad Gheshlaghi Azar. Bootstrap latent-predictive representations for multitask reinforcement learning. In *International Conference on Machine Learning*, pp. 3875–3886. PMLR, 2020.

Steven Hansen, Will Dabney, Andre Barreto, Tom Van de Wiele, David Warde-Farley, and Volodymyr Mnih. Fast task inference with variational intrinsic successor features. *International Conference on Learning Representations*, 2020.

Scott Jeen, Tom Bewley, and Jonathan Cullen. Zero-shot reinforcement learning from low quality data. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=79eWvkLjib.

Martin Klissarov and Marlos C. Machado. Deep Laplacian-based options for temporally-extended exploration. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17198–17217. PMLR, 23–29 Jul 2023.

Martin Klissarov and Doina Precup. Reward propagation using graph convolutional networks. *Advances in Neural Information Processing Systems*, 33:12895–12908, 2020.

Yehuda Koren. On spectral graph drawing. In *International Computing and Combinatorics Conference*, pp. 496–508. Springer, 2003.

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40:e253, 2017.

Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *International conference on machine learning*, pp. 5639–5650. PMLR, 2020.

Michael Laskin, Hao Liu, Xue Bin Peng, Denis Yarats, Aravind Rajeswaran, and Pieter Abbeel. Contrastive intrinsic control for unsupervised skill discovery. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pp. 34478–34491, 2022.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Marlos C. Machado, Marc G Bellemare, Erik Talvitie, Joel Veness, Matthew Hausknecht, and Michael Bowling. Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents. *Journal of Artificial Intelligence Research*, 61:523–562, 2018a.

Marlos C. Machado, Clemens Rosenbaum, Xiaoxiao Guo, Miao Liu, Gerald Tesauro, and Murray Campbell. Eigenoption discovery through the deep successor representation. In *International Conference on Learning Representations*, 2018b. URL https://openreview.net/forum?id=Bk8ZcAxR-.

Sridhar Mahadevan and Mauro Maggioni. Proto-value functions: A laplacian framework for learning representation and control in markov decision processes. *Journal of Machine Learning Research*, 8(10), 2007.

Bogdan Mazoure, Remi Tachet des Combes, Thang Long Doan, Philip Bachman, and R Devon Hjelm. Deep reinforcement and infomax learning. *Advances in Neural Information Processing Systems*, 33:3686–3698, 2020.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pp. 2778–2787. PMLR, 2017.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Doina Precup, Richard S Sutton, and Satinder Singh. Eligibility traces for off-policy policy evaluation. In *ICML*, volume 2000, pp. 759–766. Citeseer, 2000.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., USA, 1st edition, 1994. ISBN 978-0-471-61977-2.

Tongzheng Ren, Tianjun Zhang, Lisa Lee, Joseph E. Gonzalez, Dale Schuurmans, and Bo Dai. Spectral decomposition representation for reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=FBMLeaXpZN.

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

Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=uCQfPZwRaUu.

Lloyd S Shapley et al. A value for n-person games. 1953.

Kimberly L Stachenfeld, Matthew M Botvinick, and Samuel J Gershman. Design principles of the hippocampal cognitive map. *Advances in neural information processing systems*, 27, 2014.

Jingbo Sun, Songjun Tu, qichao Zhang, Haoran Li, Xin Liu, Yaran Chen, Ke Chen, and Dongbin Zhao. Unsupervised zero-shot reinforcement learning via dual-value forward-backward representation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=0QnKnt411O.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The mit press edition, 2018.

Yunhao Tang, Zhaohan Daniel Guo, Pierre Harvey Richemond, Bernardo Avila Pires, Yash Chandak, Rémi Munos, Mark Rowland, Mohammad Gheshlaghi Azar, Charline Le Lan, Clare Lyle, et al. Understanding self-predictive learning for reinforcement learning. In *International Conference on Machine Learning*, pp. 33632–33656. PMLR, 2023.

Yuandong Tian. Deep contrastive learning is provably (almost) principal component analysis. *Neural Information Processing Systems*, 2022.

Manan Tomar, Utkarsh Aashu Mishra, Amy Zhang, and Matthew E. Taylor. Learning representations for pixel-based control: What matters and why? *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=wIXHG8LZ2w.

Ahmed Touati and Yann Ollivier. Learning one representation to optimize all rewards. *Advances in Neural Information Processing Systems*, 34:13–23, 2021.

Ahmed Touati, Jérémy Rapin, and Yann Ollivier. Does zero-shot reinforcement learning exist? In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=MYEap_OcQI.

Kaixin Wang, Kuangqi Zhou, Qixin Zhang, Jie Shao, Bryan Hooi, and Jiashi Feng. Towards better laplacian representation in reinforcement learning with generalized graph drawing. In *International Conference on Machine Learning*, pp. 11003–11012. PMLR, 2021.

Yifan Wu, George Tucker, and Ofir Nachum. The laplacian in RL: Learning representations with efficient approximations. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJlNpoA5YQ.

Amy Zhang, Harsh Satija, and Joelle Pineau. Decoupling dynamics and reward for transfer learning, 2018. URL https://openreview.net/forum?id=H1aoddyvM.

Yi Zhao, Wenshuai Zhao, Rinu Boney, Juho Kannala, and Joni Pajarinen. Simplified temporal consistency reinforcement learning. In *International Conference on Machine Learning*, pp. 42227–42246. PMLR, 2023.

Ruijie Zheng, Xiyao Wang, Yanchao Sun, Shuang Ma, Jieyu Zhao, Huazhe Xu, Hal Daumé III, and Furong Huang. Taco: Temporal latent action-driven contrastive loss for visual reinforcement learning. *Advances in Neural Information Processing Systems*, 36:48203–48225, 2023.

# Supplementary Materials

*The following content was not necessarily subject to peer review.*

## A Architecture diagrams

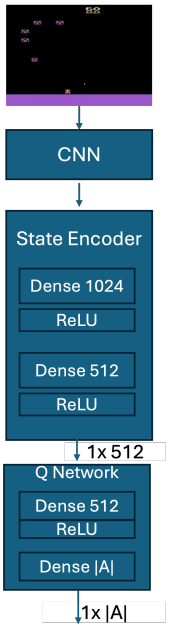


Figure 3: State PQN

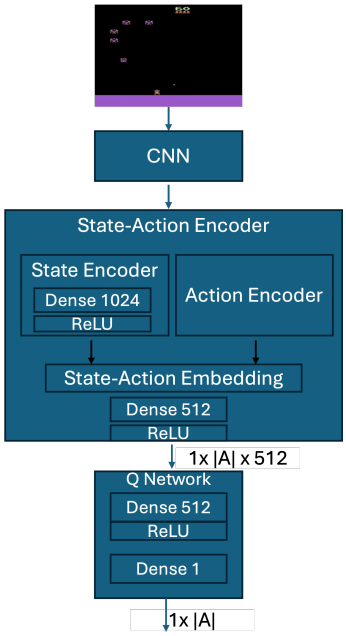


Figure 4: State-Action PQN

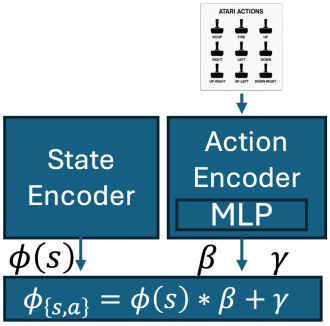


Figure 5: FiLM SA Encoder

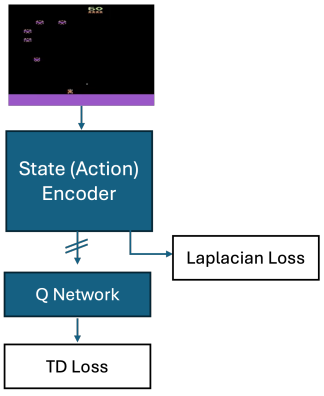


Figure 6: Laplacian-Only PQN, with stop-grad connection (arrow with crossed lines) between the encoder and Q-network

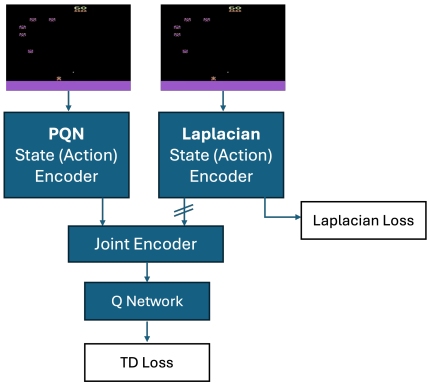


Figure 7: Fusion PQN, that concatenates TD-loss differentiable PQN embedding and fixed Laplacian embedding

## B Game-wise results of methods

Table 1: Average Human Normalized Scores across 23 Atari games.

| Games | State-Only Methods | | | | State-Action Methods | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **Laplacian** | **PQN** | | **Fusion** | **Laplacian** | | **PQN** | | **Fusion** | |
| | L:S-BP | PQN:S | PQN:S-2x | F:S-BP | L:SA-BP | L:SA-UNI | PQN:SA | PQN:SA-2x | F:SA-BP | F:SA-UNI |
| Amidar | 0.24 | 0.49 | 0.57 | 0.45 | 0.19 | 0.15 | 0.64 | 0.77 | 0.67 | 0.70 |
| Asterix | 1.05 | 34.26 | 33.00 | 31.48 | 1.30 | 1.12 | 36.26 | 40.01 | 28.74 | 29.40 |
| BattleZone | 0.64 | 1.19 | 1.13 | 1.18 | 0.59 | 0.66 | 1.22 | 1.16 | 1.22 | 1.23 |
| Bowling | 0.05 | 0.09 | 0.17 | 0.02 | 0.08 | 0.09 | -0.06 | -0.08 | 0.04 | 0.08 |
| Breakout | 4.12 | 19.58 | 20.02 | 18.36 | 4.90 | 5.46 | 17.17 | 19.63 | 17.56 | 16.60 |
| DoubleDunk | 2.70 | 2.07 | 1.35 | 2.58 | 2.89 | 2.30 | 1.95 | 2.84 | 1.55 | 1.31 |
| Enduro | 0.41 | 2.64 | 2.71 | 2.64 | 0.29 | 0.28 | 2.69 | 2.67 | 2.68 | 2.70 |
| Frostbite | 0.33 | 0.83 | 0.92 | 0.84 | 0.49 | 0.59 | 1.08 | 0.84 | 1.37 | 1.12 |
| Jamesbond | 1.66 | 5.73 | 5.99 | 4.85 | 1.54 | 1.42 | 5.59 | 5.04 | 5.57 | 5.60 |
| Kangaroo | 1.51 | 4.14 | 4.27 | 4.21 | 1.16 | 1.18 | 4.29 | 4.15 | 2.53 | 2.30 |
| KungFuMaster | 1.39 | 1.15 | 1.11 | 1.50 | 1.41 | 1.41 | 1.24 | 1.21 | 1.49 | 1.45 |
| NameThisGame | 0.54 | 0.57 | 0.77 | 0.69 | 0.34 | 0.19 | 1.48 | 1.96 | 1.40 | 1.59 |
| Phoenix | 0.70 | 5.04 | 9.90 | 6.80 | 0.60 | 0.58 | 11.50 | 15.64 | 18.19 | 22.47 |
| Pong | 1.16 | 1.18 | 1.18 | 1.16 | 1.18 | 1.17 | 1.18 | 1.18 | 1.16 | 1.18 |
| Qbert | 0.77 | 1.22 | 1.56 | 1.24 | 0.72 | 0.68 | 1.76 | 1.74 | 1.65 | 1.68 |
| Riverraid | 0.41 | 1.22 | 1.29 | 1.24 | 0.41 | 0.35 | 1.17 | 1.34 | 1.39 | 1.26 |
| Seaquest | 0.03 | 0.29 | 0.33 | 0.32 | 0.03 | 0.03 | 0.09 | 0.09 | 0.13 | 0.12 |
| Skiing | -1.02 | -1.01 | -1.01 | -1.03 | -0.52 | -0.49 | -0.98 | -1.01 | -0.27 | -0.24 |
| SpaceInvaders | 0.21 | 4.45 | 8.41 | 5.63 | 0.26 | 0.24 | 5.61 | 15.49 | 12.24 | 5.11 |
| Tennis | 1.37 | 0.98 | 1.34 | 1.47 | 0.26 | 0.28 | 1.23 | 1.30 | 0.74 | 1.30 |
| TimePilot | 0.74 | 2.83 | 3.15 | 3.33 | 0.47 | 0.02 | 3.15 | 3.61 | 4.28 | 4.53 |
| Tutankham | 0.63 | 1.28 | 1.32 | 1.33 | 1.13 | 0.75 | 0.80 | 1.26 | 0.93 | 1.54 |
| VideoPinball | 37.92 | 332.56 | 312.43 | 363.61 | 14.93 | 13.30 | 356.65 | 367.53 | 352.83 | 387.53 |

# C Shapley Contribution Scores

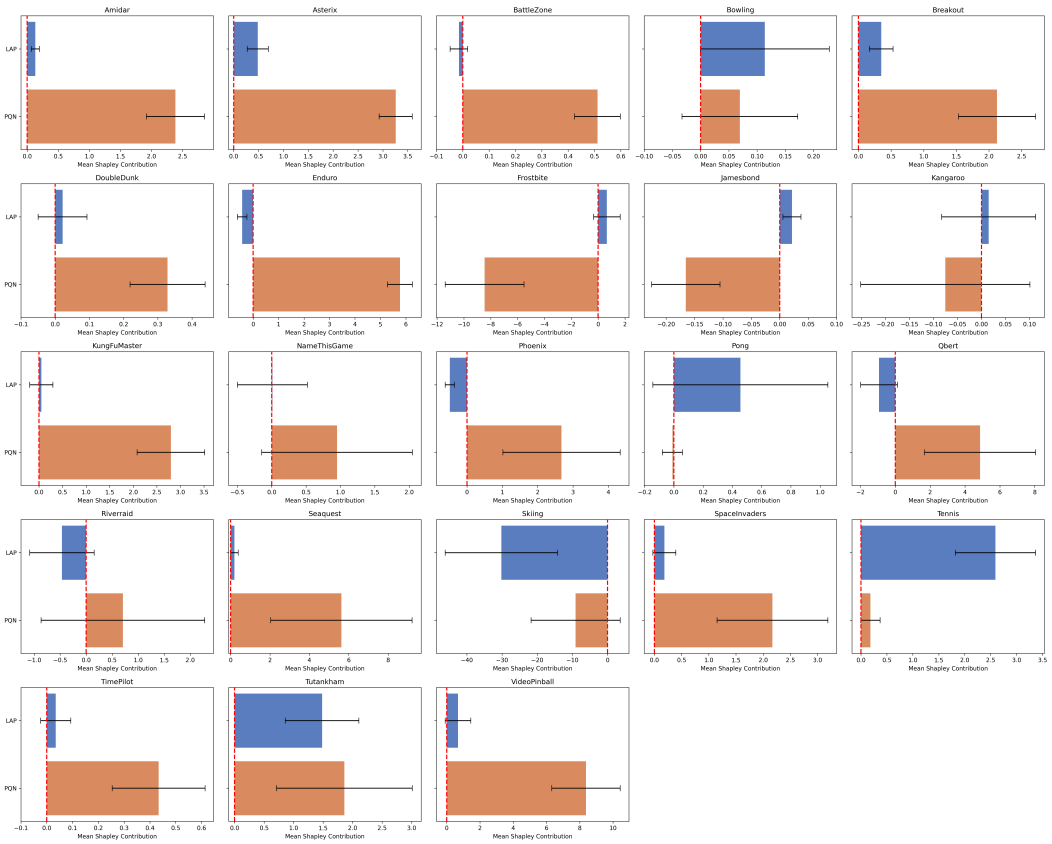


Figure 8: Average Shapley contribution (ShapC) values for each head in the Fusion model F:SA-UNI on Atari games. We compare the contributions of the standard PQN head and the Laplacian (LAP) head toward the final Q-value prediction. A higher positive ShapC indicates a stronger positive influence on the output Q-value. Each ShapC value represents the mean and standard deviation over 100 evaluation episodes (including all time steps), averaged across 5 training seeds per game.

# D Derivations

## D.1 State Representation Laplacian

The Graph Laplacian over states is defined as $L = D - W$ with $W \doteq \text{diag}(\rho)P^\pi$, $D = \text{diag}(\rho)$ is the degree matrix corresponding to $W$, and $P^\pi(s'|s) = \sum_a \pi(a|s)p(s'|s,a)$ is the transition matrix induced by policy $\pi$. Note that we assume our W is symmetric; for the scenario where it is not symmetric, we have $W = \frac{1}{2}(P^\pi + (P^\pi)^T)$. The vector $\rho$ denotes the stationary distribution of the Markov chain induced by $P^\pi$, i.e. $\rho^T P^\pi = \rho^T$. Below we derive the Equation 6 for the Laplacian of a 1-hop graph.

First we derive the marginal state-transition distribution under a uniform policy, sampled from a behavior policy. Let $\pi^{\text{uni}}$ and $\pi^{\text{B}}$ denote the uniform and behavior policies respectively.

$$
\begin{aligned}
P^{\pi^{\text{uni}}}(s'|s)\rho(s) &\doteq \mathbb{E}_{a\sim\pi^{\text{uni}}}[p(S'=s'|S=s,A=a)\rho(s)] \\
&= \sum_a \pi^{\text{uni}}(a|s)p(s'|s,a)\rho(s) \\
&= \sum_a \pi^{\text{uni}}(a|s)p(s'|s,a)\rho(s)\frac{\pi^{\text{B}}(a|s)}{\pi^{\text{B}}(a|s)} \\
&= \sum_a \frac{\pi^{\text{uni}}(a|s)}{\pi^{\text{B}}(a|s)}\pi^{\text{B}}(a|s)p(s'|s,a)\rho(s) \qquad\qquad , w(s,a) \doteq \frac{\pi^{\text{uni}}(a|s)}{\pi^{\text{B}}(a|s)} = \frac{1/|\mathcal{A}|}{\pi^{\text{B}}(a|s)} \\
&= \sum_a w(s,a)\pi^{\text{B}}(a|s)p(s'|s,a)\rho(s) \\
&= \mathbb{E}_{a\sim\pi^B}[w(s,a)p(s'|s,a)\rho(s)].
\end{aligned}
$$

We now derive the Laplacian objective under behavior policy:

$$
\begin{aligned}
< f, Lf > &= f^T L f && (10)\\
&= \sum_{s,s'} f_s(D-W)_{s,s'}f_{s'} && (11)\\
&= \sum_{s,s'} f_s f_{s'}\delta_{s,s'}\rho(s) - \sum_{s,s'} f_s f_{s'} P^\pi_{s,s'}\rho(s) && (12)\\
&= \sum_{s} f_s^2\rho(s) - \sum_{s,s'} f_s f_{s'} P^\pi_{s,s'}\rho(s) && (13)\\
&= \frac{1}{2}\sum_{s} f_s^2\rho(s) + \frac{1}{2}\sum_{s'} f_{s'}^2\rho(s') - \sum_{s,s'} f_s f_{s'} P^\pi_{s,s'}\rho(s) && (14)\\
&= \frac{1}{2}\sum_{s,s'} P^\pi_{s,s'}\rho(s) f_s^2 + \frac{1}{2}\sum_{s,s'} P^\pi_{s,s'}\rho(s) f_{s'}^2 - \sum_{s,s'} f_s f_{s'} P^\pi_{s,s'}\rho(s) && (15)\\
&= \frac{1}{2}\sum_{s,s'} P^\pi_{s,s'}\rho(s)(f_s - f_{s'})^2 = \frac{1}{2}\mathbb{E}_{\substack{s\sim\rho(S_0)\\ s'\sim P^\pi(.|S_0)}}[(f_s - f_{s'})^2] && (16)\\
&= \frac{1}{2}\sum_{s,s'}\sum_a \pi^{\text{uni}}(a|s)p(s'|s,a)\rho(s)(f_s - f_{s'})^2 && (17)\\
&= \frac{1}{2}\sum_{s,s'}\sum_a w(s,a)\pi^{\text{B}}(a|s)p(s'|s,a)\rho(s)(f_s - f_{s'})^2 && (18)\\
&= \frac{1}{2}\mathbb{E}_{\substack{s\sim\rho(S_0)\\ a\sim\pi^B(.|s)\\ s'\sim p(.|s,a)}}[w(s,a)(f_s - f_{s'})^2], && (19)
\end{aligned}
$$

where we used the fact that $\rho$ is the stationary distribution of $P^\pi$ in the 6th equality.

### D.2 Equation 7: State-Action Representation

Similar to state representation, we define the Graph Laplacian for state-action as $L = D - W$, where $D = \text{diag}(\rho)$ and $W = \text{diag}(\rho)P^\pi$, and $\pi$ is the uniform policy. The state–action representation modifies the underlying graph such that nodes now correspond to state-action pairs, hence the transition in such a graph becomes $P^\pi(s'a'|sa) = \pi(a'|s')p(s'|s,a)$ and the stationary distribution is $\rho(sa) \doteq \pi(a|s)\rho(s)$. The Laplacian objective for state-action representation under behavior policy can be derived as follow:

$$
\begin{aligned}
< f, Lf > &= f^T L f \\
&= \sum_{sa,s'a'} f_{sa}(D-W)_{sa,s'a'} f_{s'a'} \\
&= \sum_{sa,s'a'} f_{sa} f_{s'a'} \delta_{sa,s'a'} \rho(sa) - \sum_{sa,s'a'} f_{sa} f_{s'a'} P^{\pi}_{sa,s'a'} \rho(sa) \\
&= \sum_{sa} f^2_{sa} \rho(sa) - \sum_{sa,s'a'} f_{sa} f_{s'a'} P^{\pi}_{sa,s'a'} \rho(sa) \\
&= \frac{1}{2} \sum_{sa} f^2_{sa} \rho(sa) + \frac{1}{2} \sum_{s'a'} f^2_{s'a'} \rho(s'a') - \sum_{sa,s'a'} f_{sa} f_{s'a'} P^{\pi}_{sa,s'a'} \rho(sa) \\
&= \frac{1}{2} \sum_{sa,s'a'} P^{\pi}_{sa,s'a'} \rho(sa) f^2_{sa} + \frac{1}{2} \sum_{sa,s'a'} P^{\pi}_{sa,s'a'} \rho(sa) f^2_{s'a'} - \sum_{sa,s'a'} f_{sa} f_{s'a'} P^{\pi}_{sa,s'a'} \rho(sa) \\
&= \frac{1}{2} \sum_{sa,s'a'} P^{\pi}_{sa,s'a'} \rho(sa) (f_{sa} - f_{s'a'})^2 = \frac{1}{2} \mathbb{E}_{\substack{s_0 a_0 \sim \rho(S_0 A_0) \\ s'a' \sim P^{\pi}(.|S_0 A_0)}} [(f_{sa} - f_{s'a'})^2] \\
&= \frac{1}{2} \sum_{sa,s'a'} \pi^{\text{uni}}(a'|s') p(s'|s,a) \rho(sa) (f_{sa} - f_{s'a'})^2 \\
&= \frac{1}{2} \sum_{sa,s'a'} w(s',a') \pi^{\text{B}}(a'|s') p(s'|s,a) \rho(sa) (f_{sa} - f_{s'a'})^2 \\
&= \frac{1}{2} \mathbb{E}_{\substack{sa \sim \rho(S_0 A_0) \\ s' \sim p(.|s,a) \\ a' \sim \pi^B(.|s')}} [w(s',a') (f_{sa} - f_{s'a'})^2],
\end{aligned}
$$

We now expand the stationary distribution using the definition:

$$
\begin{aligned}
&\frac{1}{2} \sum_{sa,s'a'} \pi^{\text{uni}}(a'|s') p(s'|s,a) \rho(sa) (f_{sa} - f_{s'a'})^2 \\
&= \frac{1}{2} \sum_{sa,s'a'} \pi^{\text{uni}}(a'|s') p(s'|s,a) \pi^{\text{uni}}(a|s) \rho(s) (f_{sa} - f_{s'a'})^2 \\
&= \frac{1}{2} \sum_{sa,s'a'} \pi^{B}(a'|s') w(s'a') p(s'|s,a) \pi^{B}(a|s) w(s,a) \rho(s) (f_{sa} - f_{s'a'})^2 \\
&= \frac{1}{2} \mathbb{E}_{\substack{s \sim \rho(S_0) \\ a \sim \pi^B(.|s) \\ s' \sim p(.|s,a) \\ a' \sim \pi^B(.|s')}} [w(s,a) w(s'a') (f_{sa} - f_{s'a'})^2]
\end{aligned}
$$

Note that we can write the Laplacian objective as bellow where we assume $a'$ is sampled from uniform policy. Therefore, we do not need to approximate the expected action distribution via Monte Carlo, as it can be computed exactly by summing over all actions.

$$\frac{1}{2}\sum_{sa,s'a'}\pi^B(a'|s')w(s'a')p(s'|s,a)\pi^B(a|s)w(s,a)\rho(s)(f_{sa}-f_{s'a'})^2$$
$$=\frac{1}{2}\sum_{sa,s'}\sum_{a'}\pi^{\text{uni}}(a'|s')p(s'|s,a)\pi^B(a|s)w(s,a)\rho(s)(f_{sa}-f_{s'a'})^2$$
$$=\frac{1}{2}\sum_{sa,s'}\sum_{a'}\frac{1}{|\mathcal{A}|}p(s'|s,a)\pi^B(a|s)w(s,a)\rho(s)(f_{sa}-f_{s'a'})^2$$
$$=\frac{1}{2|\mathcal{A}|}\mathbb{E}_{\substack{s\sim\rho(S_0)\\ a\sim\pi^B(.|s)\\ s'\sim p(.|s,a)}}[w(s,a)\sum_{a'}(f_{sa}-f_{s'a'})^2]$$