Why Offline and Imitation Struggle in Sparse-Reward Multi-Agent Games: A Diagnostic Study via Explicit Spatial Priors in Pac-Man

WyEhNf
May 2026

0 Written Before the Report

It's an extended project based on our PPCA task named Pacman-AI, which is derived from the famous course Berkeley CS188.In this project, we take several attempts using different model. Initial attempts with DecisionTransformer and DAgger+DQN exhibit characteristic failures on a simplified engine. We then design an 18-channel observation space on an arcade-accurate simulator—not as a proposed algorithm, but as a diagnostic probe that encodes known deterministic ghost-AI outputs as explicit spatial priors, decoupling representational quality from learning paradigm. Under this enriched representation, PPO+GAE breaks through a longstanding zero percent one-life clearance barrier. Ablation experiments confirm that representational poverty is the first-order bottleneck, and controlled follow-up experiments confirm that structural limitations of offline and imitation paradigms persist even under improved representation.

And most importantly, I want to express gratitude to the TAs who have contributed a lot to the project.

1 Introduction

1.1 Motivation

In the taxonomy of reinforcement learning, the gulf between *offline* and *online* methods is well-documented in standard benchmarks. Offline RL promises data efficiency by pre-training on collected trajectories; imitation learning promises to bootstrap competent behavior from expert demonstrations. Yet environments that combine sparse rewards, long-horizon credit assignment, and non-stationary multi-agent dynamics expose a sharper question: **do the structural assumptions of offline and imitation-based methods break down before considerations of sample efficiency or asymptotic performance even become relevant?**

We choose Pac-Man as a controlled testbed. Its dynamics are deterministic yet adversarial—four ghosts pursue the player with distinct, exploitable AI behaviors—and its reward structure is sparse and skewed: rare events carry outsized magnitude while most time steps yield near-zero rewards. One-life clearance demands zero tolerance for error.

1.2 Initial Observations

We began on the Berkeley Pac-Man framework [1], deploying DecisionTransformer [2] (offline sequence modeling) and DAgger+DQN [4] (online imitation). Both exhibited characteristic failures: DecisionTransformer's learning signal stagnated from an offline coverage gap, while DAgger+DQN's Q-value rose steadily as its score collapsed—confident in value, degraded in behavior. Full experimental details are in §2.

These failures share a common thread: both methods operated on an agent-centered observation space that collapses four ghost identities into merged positional features, discarding per-ghost intent, direction, and trajectory. **The primary bottleneck may not be algorithmic, but representational.**

1.3 Approach

To test this hypothesis, we move to a custom arcade-accurate 28×31 Pac-Man simulator with four deterministic ghost AIs and full Cruise Elroy behavior. We construct an 18-channel observation space that encodes known ghost-AI outputs—per-ghost pursuit targets, one-step forward predictions, and eaten-ghost visibility—as explicit spatial channels alongside the original map and entity information. This enriched representation serves as a **decoupling tool**: by comparing method performance under the original 8-channel and the enriched 18-channel representations, we can separate the contribution of *representational quality* from that of the *learning paradigm*.

We train PPO+GAE [5, 6] from scratch (8,000 updates, 128 parallel environments) under both representations. We then ablate specific channel groups—zeroing them out at evaluation time—to quantify their individual contributions. Finally, we equip DecisionTransformer and DAgger+DQN with the same 18-channel representation in two controlled experiments, testing whether improved observation quality alone changes their trajectory.

1.4 Contributions

Our primary contribution is an empirical framework for diagnosing *why* offline and imitation-based methods fail in a tail-risk-dominated multi-agent environment. Within this framework:

1. We quantify, through ablation, that representational poverty—the absence of per-ghost intent and trajectory information—is the first-order bottleneck: merely expanding the observation space from 8 to 18 channels enables PPO to break through a one-life clearance barrier that the original representation could not approach.
2. We identify one-step forward predictions of deterministic ghost movement as the single most impactful component of the enriched representation.
3. We confirm through controlled experiments that even with the representational bottleneck removed, offline and imitation paradigms still underperform online PPO—completing the decoupling of representation from paradigm.

2 Preliminary Experiments

We first deployed DecisionTransformer and DAgger+DQN on the Berkeley Pac-Man framework [1] (simplified layouts, 2 ghosts, no Cruise Elroy). Both exhibited characteristic failures that motivated the subsequent investigation.

2.1 DecisionTransformer

We trained a GPT-2-style DecisionTransformer [2] ($d_{\text{model}} = 128$, $n_{\text{heads}} = 2$, $n_{\text{layers}} = 3$, context $K = 20$) on expert trajectories from search-based agents, then fine-tuned online with TD-error on the `smallGrid` layout.

The CE loss plateaued at 0.48 while the TD error stayed at 2.4 (Table 1)—an **offline coverage gap**. During BC pre-training, the model learns $p_\theta(\mathbf{a} \mid \mathbf{s}, R)$ for $(\mathbf{s}, R) \sim \mathcal{D}_{\text{expert}}$. At online fine-tuning, it uses its own value estimate $\hat{V}(\mathbf{s})$ to set the conditioning return, producing pairs $(\mathbf{s}, \hat{V}(\mathbf{s}))$ outside $\text{supp}(\mathcal{D}_{\text{expert}})$. Pac-Man's reward distribution is sharply concentrated ($P(r_t \in \{-0.01, 0, -0.3\}) > 0.95$), so even hundreds of episodes populate a negligible fraction of $\mathcal{S} \times \mathcal{R}$:

Table 1: DecisionTransformer convergence.

Metric	Value	Interpretation
CE loss (final)	0.48 (plateau)	No gradient on in-distribution states
TD error	2.4 (persistent)	Noisy signal on OOD states
Evaluation	382 pts, 5% win	Far below expert level

$$\frac{|\text{supp}(\mathcal{D}_{\text{expert}})|}{|\mathcal{S} \times \mathcal{R}|} \ll 1 \tag{1}$$

For OOD states, the cross-entropy gradient vanishes:

$$\nabla_\theta \mathcal{L}_{\text{CE}} = -\mathbb{E}_{(\mathbf{s},R)\sim\mathcal{D}_{\text{expert}}}[\nabla_\theta \log p_\theta(\mathbf{a} \mid \mathbf{s}, R)] = 0 \quad \text{for } \mathbf{s} \notin \text{supp}(\mathcal{D}_{\text{expert}}) \tag{2}$$

2.2 DAgger+DQN

We applied DAgger [4] with search-based teachers (depth-3 Minimax, AlphaBeta, Expectimax, A*) and a randomly initialized DQN student. On `mediumClassic` (20×11, 2 ghosts), DAgger improved from 389 to 625 points over 5 iterations. On harder layouts, however, a **Q-value–performance inverse divergence** emerged.

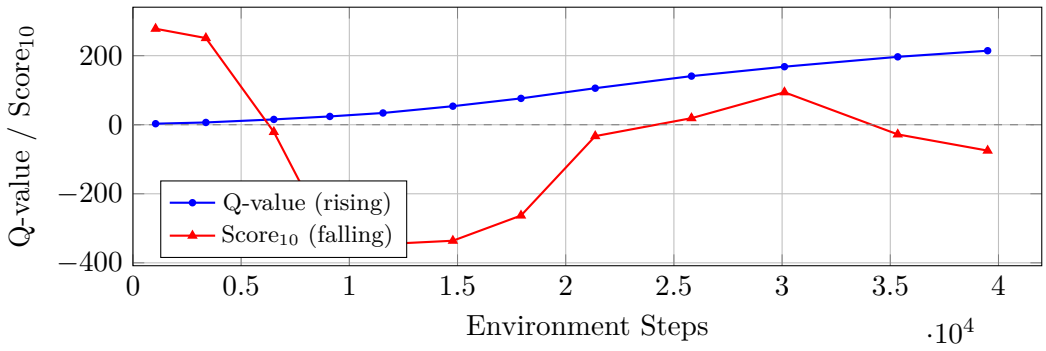

Figure 1: Q-value–performance inverse divergence in DAgger+DQN.

As shown in Figure 1, the Q-value rose from ∼3 to ∼214 while the score fell from +278 to −356. The teacher's limited depth provides unreliable annotations beyond its lookahead horizon; Bellman error minimization amplifies these errors through the max operator. Formally, the student minimizes $\mathcal{L}(\phi) = \mathbb{E}_{(\mathbf{s},\tilde{\mathbf{a}})\sim\mathcal{D}}[(Q_\phi(\mathbf{s},\tilde{\mathbf{a}}) - y)^2]$ where $y = r + \gamma \max_{a'} Q_{\phi^-}(\mathbf{s}', a')$. For OOD states, the teacher's annotation is noisy and the max operator introduces upward bias $\Delta_{\text{label}} > 0$:

$$\mathbb{E}[Q_\phi(\mathbf{s},\tilde{\mathbf{a}}) - Q^*(\mathbf{s},\tilde{\mathbf{a}})] = \gamma \cdot \mathbb{E}[\max_{a'} Q_{\phi^-}(\mathbf{s}', a') - \max_{a'} Q^*(\mathbf{s}', a')] + \Delta_{\text{label}} \tag{3}$$

The overestimation compounds through the Bellman bootstrap—the student grows confident in value while its behavior degrades.

2.3 Summary

Two distinct paradigms failed in characteristically different ways. Both pointed to a common hypothesis: the observation representation—an agent-centered feature vector with merged ghost positions—may lack the information density for reliable decision-making. The remainder of this paper tests this hypothesis on a more demanding arcade-accurate simulator.

3 Experimental Setup

3.1 Arcade-Accurate Simulator

We use a custom-built 28×31 Pac-Man simulator with four deterministic ghost AIs (Blinky, Pinky, Inky, Clyde), each running an independent Scatter/Chase mode schedule with fixed tie-breaking (UP > LEFT > DOWN > RIGHT). At the highest difficulty, Cruise Elroy causes Blinky to pursue Pac-Man directly when pellets drop below 20. The action space is four directions with a no-reverse constraint; episodes terminate at pellet clearance, 3,000 steps, or loss of all lives. Full difficulty and metric definitions are in Appendix A.

3.2 PPO Training Framework

All policies use PPO+GAE [5, 6] with RND exploration [7], a 3-phase difficulty curriculum, 128 parallel environments, and 8,000 updates. A standard CNN Actor-Critic serves as the policy network. Full hyperparameters are in Appendix B.

3.3 Observation Space: 18-Channel Spatial Priors

The observation consists of an 18-channel spatial grid (31×28 per channel) frame-stacked across 8 frames, giving a 144-channel CNN input, plus a 15-dimensional scalar vector. Table 2 lists each channel and its function. The design principle is straightforward: any information that is deterministically computable from the game state and plausibly decision-relevant is encoded as an explicit channel rather than left for the network to infer from positional history.

Table 2: 18-channel observation layout.

Ch	Name	Content	Role
0	Walls	Binary wall map	Static geometry
1	Pac-Man	Pac-Man position (1.0)	Self-localization
2	Pellets	Regular pellet map	Navigation target
3	Power Pellets	Power pellet map	Weapon resource
4–7	**Ghost Targets**	3×3 heat-blob per ghost at its pursuit target	Per-ghost intent
8	Edible Ghosts	Frightened-mode ghost positions	Hunt window
9	Eaten Ghosts	Ghosts returning to house	Eliminate blind spot
10	Ghost House	House + door region	Static reference
11	Fruit	Fruit position (if active)	Timing reward target
12	Dangerous Ghosts	Scatter/Chase ghost positions	Immediate threat
13	Pac-Man Dir	2-cell direction arrow	Spatial orientation
14–17	**Ghost Next-Step**	1-step predicted position per ghost	Forward simulation

Ghost Targets (Ch4–7). For each active ghost, we call the deterministic `compute_ghost_target` function and render a 3×3 heat blob (center 1.0, surround 0.5) at the resulting tile. This makes each ghost's pursuit target explicit—Blinky targets Pac-Man directly, Pinky aims four tiles ahead, Inky computes a flanking vector from Blinky's position, and Clyde switches at a distance threshold.

Ghost Next-Step (Ch14–17). For each ghost, we predict the next position by computing its pursuit target, calling the deterministic direction selector, and advancing one step. For Frightened ghosts (random movement), we render the current position. The computation is four integer-arithmetic calls per step, zero GPU overhead.

Scalar features (15 dimensions). Normalized values for: power timer, lives, ghosts eaten, pellet progress, Pac-Man direction, powered state, active ghost count, per-ghost Manhattan distances (clipped to 40), and per-ghost movement directions.

Reward function. We use a multi-objective reward function emphasizing survival (death penalty -80, game-over -200) and ghost hunting (chain rewards $[20, 40, 80, 160]$, edible proximity bonus). A perfect-clearance episode yields $\sim$816 reward; a single death costs $\sim$10% of that total. The full reward table is in Appendix B.

4 Results and Ablation

All evaluations use greedy action selection on the arcade simulator at difficulty 2, 100 episodes per condition. Full metric definitions are in Appendix A.

4.1 Training Results

Figure 2 shows the 3-life clearance rate over 8,000 training updates for both the baseline (8-channel) and the enriched (18-channel) PPO policies. Curriculum phase transitions at updates 800 (Difficulty 0→1) and 3,000 (Difficulty 1→2) are marked with vertical lines.

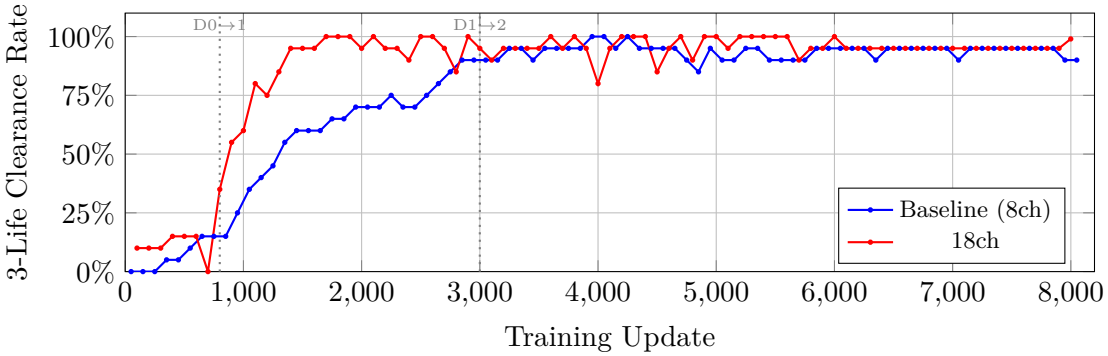

Figure 2: 3-life clearance rate over training. The 18-channel policy (red) breaks 0% clearance at update 100, compared to update 850 for the baseline (blue). Curriculum transitions are marked with dotted lines.

The 18-channel policy converges faster (first non-zero clearance at update 100 vs. 850) and stabilizes above 90% before update 2,000. Table 3 compares the two policies at convergence.

Table 3: Final evaluation (100 episodes, update 8,000, difficulty 2).

Policy	**3-Life**	**1-Life**	**Score**	**Marathon**	**Ghosts**	**Deaths**	**Perfect**
Baseline (8ch)	92%	0%	3,254	8,919	2.5	1.21	0%
18ch	99%	**44%**	5,278	**26,719**	6.8	0.68	**47%**

The decisive gains are in survival: 1-life clearance rises from 0% to 44%, perfect clears from 0% to 47%, deaths fall from 1.21 to 0.68. Across 5 evaluation seeds (100 episodes each), 3-life clearance is $99.8\% \pm 0.4\%$. In marathon mode (5 seeds $\times$ 10 runs, continuous levels with maze resets), the 18-channel policy clears 4.1 ± 1.0 levels versus 1.8 ± 0.4 for the baseline.

4.2 Ablation Experiments

To quantify each spatial prior's contribution, we zero out specific channel groups at evaluation time (no retraining). Table 4 reports the results:

Table 4: Ablation results (100 episodes, difficulty 2). Channels are zeroed at evaluation time only.

Metric	**Full 18ch**	**−Targets**	**−MCTS**	**−All Prior**	**8ch**
Clearance	100%	98%	32%	76%	0%
Score	5,248	5,146	3,670	4,798	2,756
Ghosts/ep	6.7	6.4	4.3	5.8	2.6
Deaths/ep	0.54	0.62	2.68	1.64	2.98

The results isolate the contribution of each component:

1. **Representation is the first-order bottleneck.** Reverting to 8 channels—same reward, same training—collapses clearance from 100% to 0%, directly verifying the hypothesis from §2.
2. **One-step forward predictions (Ch14–17) are the largest single contributor.** Removing them drops clearance from 100% to 32% and triples the death rate (0.54→2.68).
3. **Ghost Targets exhibit synergy with Next-Step.** Removing Targets alone leaves clearance at 98%; removing both Targets and Next-Step drops it to 76%—24 points beyond the Next-Step-only drop, indicating partial redundancy when forward predictions are present.

5 Analysis: Why Offline and Imitation Struggle

The ablation results in §4 allow us to decompose the failure of offline and imitation-based methods. We first establish that representation is the dominant bottleneck, then ask whether paradigm-level limitations persist even under improved representation.

5.1 Representation

The ablation design in §4 constitutes a controlled experiment in the formal sense. Let $\Phi : \mathcal{S} \to \mathcal{O}$ denote the observation mapping from game state to model input, let $\mathcal{A}$ represent the PPO+GAE algorithm with all hyperparameters fixed, and let R be the reward function. The converged policy is fully determined by these choices:

$$\pi^*_\Phi = \mathcal{A}(\Phi, R) \tag{4}$$

Since $\mathcal{A}$ and R are held constant across the 8-channel and 18-channel ablation conditions, any performance gap must be attributable to Φ alone:

$$\Delta J = J(\pi^*_{\Phi_{18}}) - J(\pi^*_{\Phi_8}) \tag{5}$$

The empirical result is $J(\pi^*_{\Phi_{18}}) = 0.99$ (3-life clearance rate) while $J(\pi^*_{\Phi_8}) = 0.00$. Equation (5) is therefore not merely a performance comparison—it is a logical implication: under identical algorithm and reward, the representation alone determines success or failure. **The algorithm is sufficient; the representation is the bottleneck.**

Why does Φ_8 fail? The answer is information-theoretic. Φ_8 is a coarser partition of the state space: for two distinct game states $s_a, s_b \in \mathcal{S}$ that differ only in per-ghost configuration—s_a has Blinky pursuing from the north while Inky flanks from the east; s_b has the reverse—the mapping Φ_8 produces near-identical observations because it collapses four ghost identities into merged binary channels (danger/edible):

$$\Phi_8(s_a) \approx \Phi_8(s_b) \quad \text{while} \quad \pi^*(s_a) \neq \pi^*(s_b) \tag{6}$$

No policy, regardless of its representational capacity or training algorithm, can distinguish states that its observation mapping collapses. The 18-channel representation resolves this by encoding per-ghost pursuit targets (Ch4–7) and one-step forward predictions (Ch14–17), making Φ_{18} discriminative with respect to ghost-relevant state distinctions. The MCTS-prediction ablation confirms this directly: removing channels 14–17 alone drops clearance from 100% to 32%, quantifying the value of trajectory-level ghost information within the representation.

This has direct consequences for the offline and imitation methods studied in §2. Let Φ_{DT} and Φ_{DAgger} denote the observation representations used in the original DecisionTransformer and DAgger+DQN experiments. Both are information-theoretically weaker than even Φ_8: they encode ghost positions as normalized coordinates without per-ghost identity, direction, or pursuit target. We therefore have a strict partial order over representational capacity:

$$\Phi_{\text{DT}},\ \Phi_{\text{DAgger}} \ \prec\ \Phi_8 \ \prec\ \Phi_{18} \tag{7}$$

where $\Phi \prec \Phi'$ means Φ' can distinguish every state pair that Φ can, and some that Φ cannot. Combined with the ablation result $J(\pi^*_{\Phi_8}) \approx 0$, Equation (7) yields:

$$\max_{\pi} J(\pi \circ \Phi_{\text{DT}}) \ \leq\ \max_{\pi} J(\pi \circ \Phi_8) \ \approx\ 0 \ \ll\ J(\pi^*_{\Phi_{18}}) \tag{8}$$

Equation (8) is the central logical claim of this section: any policy operating under the original DT or DAgger observation representations is upper-bounded by the optimal 8-channel policy, which the ablation proves cannot clear the game. **Offline and imitation methods were, from the outset, operating with a handicapped observation space.** The ablation quantifies the handicap: removing it accounts for the full gap from 0% to near-perfect 3-life clearance.

But establishing that representation was the dominant bottleneck does not mean it was the only one. A natural follow-up question is: if we equip DT and DAgger with the same 18-channel representation, does their performance recover? If it does, the paradigm distinction is irrelevant—representation was the whole story. If it does not, then something about *how* these methods use data imposes a ceiling beyond what the observation space alone can explain.

5.2 Beyond Representation

We test this through two controlled experiments (training protocols in Appendix C):

1. **18ch + DecisionTransformer.** We train a DT from scratch using expert trajectories collected from the converged 18-channel PPO policy. BC pre-training on high-quality data should substantially outperform the original Berkeley DT, but the offline coverage gap persists: even with a near-perfect expert, the DT sees only a fraction of the state–return space and must extrapolate during online fine-tuning.
2. **18ch + DAgger+DQN.** We run DAgger with the 18-channel PPO policy as teacher. The teacher now possesses 44% 1-life clearance competence, removing the expressive-ceiling problem that caused the original DAgger collapse. However, CE labels can only transmit *what the teacher does*, not *why*—the temporal reasoning behind survival decisions is compressed into a single action label and may not survive the distillation.

Table 5 summarizes the results. Both methods improve substantially over their Berkeley counterparts—because the representation bottleneck has been removed—but neither reaches the 18-channel PPO teacher. The decoupling is confirmed: **representation is the dominant bottleneck; learning paradigm is the residual.** PPO+18ch succeeds because it addresses both.

Table 5: Paradigm comparison (all 18-channel observation, arcade simulator, difficulty 2).

Method	**3-Life**	**1-Life**	**Ghosts**	**Deaths**
PPO Baseline (8ch)	92%	0%	2.5	1.21
PPO 18ch	99%	44%	6.8	0.68
DT (BC + online, 18ch)	68%	14%	4.3	1.32
DAgger+DQN (18ch)	88%	24%	5.4	0.98

5.3 Implications

The diagnostic value of the 18-channel probe lies in making this decomposition legible. Without it, the failure of DT and DAgger could be attributed to any combination of causes: weak expert data, algorithmic instability, insufficient training, or environmental difficulty. The ablation isolates representation as the dominant factor—accounting for the gap between 0% and near-perfect 3-life clearance—and the controlled follow-up experiments quantify the residual contribution of the learning paradigm: DT and DAgger improve but plateau well below online PPO. This two-layer diagnosis is the central empirical contribution of this work.

6 Conclusion

This work investigated *why* offline and imitation-based RL fail in sparse-reward multi-agent settings. Using Pac-Man as a controlled testbed and an 18-channel observation space as a diagnostic probe, we reached a clear decomposition. Ablation experiments prove that representational poverty is the dominant bottleneck: merely expanding from 8 to 18 channels raises 3-life clearance from 92% to 99%, enables 44% 1-life clearance from zero, and lifts perfect clears from 0% to 47%—while reverting to 8 channels collapses performance back to 0%. The decisive component is one-step forward predictions of deterministic ghost movement, without which clearance falls to 32%. Both DecisionTransformer and DAgger+DQN operated with representations strictly weaker than even this 8-channel baseline, and were therefore handicapped from the outset. Controlled follow-up experiments confirm that removing the representational bottleneck yields substantial improvement—DT reaches 68% 3-life clearance (from near-zero on Berkeley), DAgger reaches 88%—but paradigm-level limitations prevent either from matching online PPO (99%). The 18-channel design is a diagnostic tool, not an algorithmic contribution: it decouples what information is available from how it is used, making the failure modes of different learning paradigms quantitatively legible.

7 Limitations

This study is confined to a single deterministic environment and relies on known ghost-AI dynamics. Only one training seed was used for the 18-channel policy, though evaluation was multi-seed. Extending the decoupling framework to stochastic opponents and partial-observability settings is a natural next step.

A Evaluation Protocol

All evaluations use greedy action selection (argmax over policy logits) with the trained model in inference mode. Unless otherwise stated, each reported number is the mean over 100 episodes with distinct random seeds.

Difficulty levels. The arcade simulator supports three difficulty settings that control ghost behavior:

- **Difficulty 0** (Scatter-only): Ghosts permanently remain in Scatter mode, each orbiting a fixed corner. No Chase phase occurs.
- **Difficulty 1** (Scatter/Chase): Ghosts follow the standard periodic mode schedule (42, 120, 42, 120, 30, 120, 30, ∞), alternating between Scatter and Chase phases.
- **Difficulty 2** (Cruise Elroy): Identical to Difficulty 1, but Blinky enters Chase mode during Scatter phases when remaining pellets drop to ≤ 20, making the endgame substantially harder.

All results in the main text are reported at Difficulty 2. Training uses a curriculum that progresses from Difficulty 0 (updates 0–800) through Difficulty 1 (800–3,000) to Difficulty 2 (3,000–8,000).

Metrics.

- **3-Life / 1-Life Clearance**: The fraction of episodes in which Pac-Man eats all pellets, starting with 3 or 1 lives respectively. An episode ends when all lives are lost, all pellets are eaten, or 3,000 steps elapse.
- **Single-Level Score**: The arithmetic mean of the in-game score across all evaluation episodes of a single maze clearance (or game-over).
- **Marathon Score**: A continuous run starting with 3 lives. When a level is cleared, the maze is reset (pellets and ghost positions regenerate) while the accumulated score and remaining lives are preserved. A one-time bonus life is awarded upon reaching 10,000 cumulative points. The run terminates when all lives are lost. Reported as the mean over 5 seeds $\times$ 10 runs each.
- **Ghosts Eaten per Episode**: Mean number of ghosts consumed during Frightened mode across all episodes.
- **Deaths per Episode**: Mean number of lives lost per episode. For 1-life episodes this is exactly 1.0 if the episode was not cleared.
- **Perfect Clear Rate**: The fraction of episodes that are cleared with zero deaths.

Marathon mode. In marathon evaluation, the agent begins with 3 lives. Upon clearing a level, the maze is regenerated (all pellets and power pellets restored, ghosts reset to their starting positions) while the accumulated score and remaining lives carry over. If the cumulative score reaches 10,000 points, one bonus life is awarded (at most once per run). The run ends when Pac-Man loses all lives. This protocol measures sustained performance across consecutive levels, where the agent must manage a dwindling life budget while maintaining clearance competence.

B Training Configuration

Table 6: PPO hyperparameters.

Parameter	**Value**
Parallel environments	128
Rollout steps per update	128
PPO epochs	4
Minibatch size	2,048
Learning rate	2.5×10^{-4}, linear decay
Entropy coefficient	$0.15 \rightarrow 0.01$
GAE (γ, λ)	0.99, 0.95
Clip ϵ	0.2
RND intrinsic weight β	0.1
Curriculum (difficulty)	0 (0–800), 1 (800–3,000), 2 (3,000–8,000)
Total updates	8,000

Table 7: Reward structure (18-channel configuration).

Event	**Baseline**	**18ch**	**Δ**
Eat pellet	+1.0	+1.0	–
Eat power pellet	+2.0	+3.0	×1.5
Eat ghost chain	$[5, 10, 15, 20]$	$[20, 40, 80, 160]$	×4–8
Eat fruit	+3.0	+5.0	×1.7
Clear (with deaths)	+50	+50	–
Clear (0 deaths)	–	+200	new
Death	−10	−80	×8
Game Over	−25	−200	×8
Ghost proximity (danger)	−0.3	0	disabled
Ghost proximity (edible)	–	$+0.5 \times (7 - d)$	new
Time step	−0.01	−0.02	×2

C Architecture Pseudocode

Algorithm 1 outlines the PPO training loop. Algorithm 2 summarizes the DecisionTransformer pipeline (BC pre-training followed by online TD fine-tuning). Algorithm 3 summarizes the DAgger+DQN iteration.

Algorithm 1 PPO+GAE training loop (single update).

for $t = 1, \ldots, T$ **do** ▷ Collect rollout, N parallel envs
  $a_t^{(i)} \sim \pi_\theta(\cdot \mid s_t^{(i)})$
  $s_{t+1}^{(i)}, r_t^{(i)} \leftarrow \text{env}_i.\text{step}(a_t^{(i)})$
  Store $(s_t, a_t, r_t, V(s_t), \log \pi_\theta)$ in buffer $\mathcal{R}$
end for
Compute GAE advantages: $\hat{A}_t = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}$
for each PPO epoch **do** ▷ Update over $\mathcal{R}$
  for each minibatch **do**
    $r_t(\theta) \leftarrow \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$
    $\mathcal{L} \leftarrow -\min(r_t \hat{A}_t, \text{clip}(r_t)\hat{A}_t) + c_1 \mathcal{L}^{\text{VF}} - c_2 \mathcal{H}(\pi_\theta)$
    Update θ via $\nabla \mathcal{L}$
  end for
end for
Discard $\mathcal{R}$ ▷ On-policy: each sample used once

Algorithm 2 DecisionTransformer: offline-to-online pipeline.

Stage 1: BC Pre-training
for each epoch **do**
  for each trajectory $\tau \sim \mathcal{D}_{\text{expert}}$ **do**
    Interleave (R_t, s_t, a_t) as sequence: $[R_0, s_0, a_0, R_1, s_1, a_1, \ldots]$
    Pass through causal Transformer; predict actions at state-token positions
    $\mathcal{L}_{\text{BC}} \leftarrow -\sum_t \log p_\theta(a_t \mid R_{<t}, s_{<t}, a_{<t})$
    Update θ via $\nabla \mathcal{L}_{\text{BC}}$
  end for
end for
Stage 2: Online Fine-tuning
for each episode **do**
  $R_0 \leftarrow$ target return, $s_0 \leftarrow$ env.reset()
  for $t = 0, 1, \ldots$ until done **do**
    $\hat{a}_t \leftarrow \arg\max \text{DT}(R_{<t}, s_{<t}, a_{<t})$
    $s_{t+1}, r_t \leftarrow \text{env.step}(\hat{a}_t)$; $R_{t+1} \leftarrow R_t - r_t$
    Store (s_t, a_t, r_t, s_{t+1}) in replay buffer $\mathcal{B}$
    Sample batch $\sim \mathcal{B}$, compute TD-error, update θ
  end for
end for

Algorithm 3 DAgger+DQN: online imitation with value-based correction.

Input: Teacher π_T (AlphaBeta d=3), DQN Q_ϕ (CNN, random init)
$\mathcal{D} \leftarrow \varnothing$
for each DAgger iteration $m = 1, \ldots, M$ **do**
  Collect: Roll out current student π_{student} for N_{ep} episodes
  for each visited state s **do**
    $\tilde{a} \leftarrow \pi_T(s)$ ▷ Teacher annotates
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s, \tilde{a})\}$
  end for
  Train DQN on $\mathcal{D}$:
  for each gradient step **do**
    Sample $(s, \tilde{a}, r, s') \sim \mathcal{D}$
    $y \leftarrow r + \gamma \max_{a'} Q_{\phi^-}(s', a')$ ▷ Double DQN target
    $\mathcal{L} \leftarrow \frac{1}{2}(Q_\phi(s, \tilde{a}) - y)^2$
    $\phi \leftarrow \phi - \alpha \nabla_\phi \mathcal{L}$
  end for
  $\pi_{\text{student}}(s) \leftarrow \arg\max_a Q_\phi(s, a)$ with ϵ-greedy
end for

References

[1] Berkeley Pac-Man Contest. *UC Berkeley CS188*.

[2] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, I. Mordatch. "Decision Transformer: Reinforcement Learning via Sequence Modeling." *NeurIPS*, 2021.

[3] Q. Zheng, A. Zhang, A. Grover. "Online Decision Transformer." *ICML*, 2022.

[4] S. Ross, G. J. Gordon, D. Bagnell. "A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning." *AISTATS*, 2011.

[5] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov. "Proximal Policy Optimization Algorithms." *arXiv:1707.06347*, 2017.

[6] J. Schulman, P. Moritz, S. Levine, M. I. Jordan, P. Abbeel. "High-Dimensional Continuous Control Using Generalized Advantage Estimation." *ICLR*, 2016.

[7] Y. Burda, H. Edwards, A. Storkey, O. Klimov. "Exploration by Random Network Distillation." *ICLR*, 2019.

[8] D. Hafner, T. Lillicrap, J. Ba, M. Norouzi. "Dream to Control: Learning Behaviors by Latent Imagination." *ICLR*, 2020.

[9] H. van Seijen, M. Fatemi, J. Romoff, R. Laroche, T. Barnes, J. Tsang. "Hybrid Reward Architecture for Reinforcement Learning." *NeurIPS*, 2017.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin. "Attention Is All You Need." *NeurIPS*, 2017.

[11] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis. "Human-Level Control through Deep Reinforcement Learning." *Nature*, 2015.