

No Back-Propagation Through Time via Future Action Prediction

Kasra Fouladi

Iran University of Science and Technology (IUST)

k4sr405@gmail.com

Github: <https://github.com/kasrafouladi/StrikeForce-No-BPTT-via-FAP>

July 2026

Domain: Imitation Learning, Game AI, POMDPs

Keywords: behavioral cloning, future prediction, stop-gradient, representation learning, partial observability

1 Abstract

We introduce a decoupled architecture for imitation learning in partially observable games that removes Back-Propagation Through Time (BPTT) from the perception backbone entirely. A **Future-Action Predictor (FAP)** backbone compresses each observation into a latent vector trained solely to forecast the expert’s future actions over a bounded horizon H . A separate **Transformer aggregator** then combines a sliding window of these detached representations, together with local raw observations and the agent’s own action history, to produce the policy output. Because gradients from the policy objective are stopped before the backbone, the two components specialise independently, giving memory complexity $\mathcal{O}(H^2)$ with respect to episode length T . We provide a theoretical analysis showing that, under a *pseudo-Markovian* assumption, the FAP objective avoids the temporal Jacobian products responsible for BPTT’s vanishing/exploding-gradient mechanism, and that a no-regret aggregation mechanism exists for the overlapping predictions. Our bias-variance analysis does not assume identical per-predictor statistics or monotonicity of bias/variance with horizon; under the stated conditions, non-perfectly correlated frame-wise predictions can yield variance reduction, while perfect correlation recovers the corresponding single-predictor behavior. We evaluate our approach on the StrikeForce game environment and compare the resulting architecture against alternative aggregation strategies.

2 Introduction

2.1 Problem

Modern game AI frequently deals with **partially observable Markov decision processes (POMDPs)**: the agent sees only part of the true state, and good decisions depend on the history of observations, not just the current frame.

2.2 Limitations of Recurrence and BPTT

The standard remedy is a recurrent network unrolled over the whole episode via back-propagation through time (BPTT). This remedy, however, is memory-hungry, prone to unstable gradients, and couples representation learning to long-horizon credit assignment.

2.3 A Guiding Intuition

Given one freeze-frame, a human observer can make a reasonable guess about the next few moments, with the guess degrading further out. This is the role we ask of our backbone—not to recover the full state, but to produce a representation shaped by “what is likely to happen next.” The actual work of combining information across multiple frames is delegated to a separate, short-window aggregator.

2.4 Our Decoupled Solution

We instantiate this as two independently trained components (Section 3). A Future-Action Predictor (FAP) backbone compresses each observation into a latent vector, trained only via a multi-step future-action-prediction loss, with no gradient from the policy objective reaching it. A Transformer aggregator consumes a sliding window of these detached latents, local raw observations, and the agent’s own executed-action history, to produce the immediate policy output. The prediction horizon of the backbone and the window size of the aggregator are tied to the same single hyperparameter H .

2.5 Summary of Contributions

Architecture:

- A stop-gradient boundary that decouples backbone training from aggregator training, removing BPTT from the backbone while preserving causality at inference.
- A bounded-window Transformer aggregator that fuses detached future-shaped latents, raw observations, and action history without recurrence.

Theory:

- A derivation (Section 4) showing that, under a pseudo-Markovian assumption, the FAP training path contains no temporal Jacobian product analogous to that of BPTT.
- A no-regret Hedge guarantee for the aggregation of overlapping predictions, providing a theoretical reference for the aggregation problem.
- A generalised bias-variance analysis that does not assume identical per-predictor statistics, monotonicity of bias/variance, or any specific correlation structure beyond boundedness. Under the stated covariance conditions, aggregation can reduce variance; when the predictors are perfectly correlated, the aggregate reduces to the corresponding single-predictor behavior.
- Memory complexity $\mathcal{O}(H^2)$ for the bounded-window Transformer, independent of T .

3 Method

3.1 Backbone: Future Action Predictor (FAP)

Given an observation o_t (a 31×31 grid with 32 channels), we use a two-stage axial cross-attention backbone to produce a compact latent $\mathbf{z}_t \in \mathbb{R}^{256}$. H linear heads map $\mathbf{z}_t$ to logits for each future action:

$$\hat{a}_t^{(0)}, \hat{a}_t^{(1)}, \dots, \hat{a}_t^{(H-1)} = \text{FAP}(o_t)$$

where $\hat{a}_t^{(k)}$ predicts the expert action k steps ahead. **This is a sequence of H distinct future predictions, not H guesses for the same instant.**

Backbone loss: Cross-entropy loss over the future-action sequence:

$$\mathcal{L}_t^{\text{BB}} = -\frac{1}{\sum_{k=0}^{H-1} \gamma^k} \sum_{k=0}^{H-1} \gamma^k \log p_\theta(a_{t+k} \mid o_t, k), \quad \gamma = 0.9,$$

where $p_\theta(a_{t+k} \mid o_t, k)$ denotes the softmax probability assigned by the k -th future-action head to the expert action at time $t+k$.

3.2 Policy Aggregator with Detached Features

At time t , the policy network receives a token sequence $[z_{t-H+1}, \dots, z_t]$. Each token z_i encapsulates:

1. $\text{sg}(\mathbf{z}_i)$ (detached backbone representation),
2. $\mathbf{c}_i$: the raw feature vector sampled at the agent’s fixed grid cell (player status – e.g., health and other per-frame attributes), read directly from the observation channels, not a spatial window,
3. previous executed action a_{i-1} (one-hot).

Note that $\mathbf{z}_i$ carries no less information than the full prediction vector $[P_i^{(0)}, \dots, P_i^{(H-1)}]$ output by the FAP heads, since the latter is itself a deterministic (learned) function of $\mathbf{z}_i$. This motivates tokenizing on the compact latent $\mathbf{z}_i$ rather than on the H -way prediction outputs directly: no information available through the prediction heads is discarded, while the token dimensionality stays fixed regardless of H .

Gradients from the policy loss flow inside the aggregation module but **never** enter the backbone.

Memory complexity: The Transformer self-attention over H tokens costs $\mathcal{O}(H^2)$ per forward pass, independent of T . Full-episode BPTT requires $\mathcal{O}(T)$ memory.

3.3 Training Procedure and Causality

$$\mathcal{L}^{\text{total}} = \mathcal{L}^{\text{R}} + \mathcal{L}^{\text{BB}}, \quad \nabla_{\theta_{\text{backbone}}} \mathcal{L}^{\text{R}} = 0, \quad \nabla_{\theta_{\text{aggregator}}} \mathcal{L}^{\text{BB}} = 0$$

The backbone is trained with the time-discounted cross-entropy loss $\mathcal{L}^{\text{BB}}$, while the policy aggregator is trained with the focal loss $\mathcal{L}^{\text{R}}$ using $\gamma_{\text{focal}} = 2$. Training uses AdamW with no backbone BPTT. The backbone processes each frame independently; the aggregator’s effective history is bounded by H .

4 Theoretical Analysis

In this section we provide a mathematical analysis of replacing BPTT with FAP in pseudo-Markovian environments. We first formalise the environment assumption, then analyze gradient stability, and finally establish a theoretical guarantee for the aggregation step without assuming identical per-predictor statistics or monotonicity of bias/variance with horizon.

4.1 Setting and Pseudo-Markov Property

We consider an imitation-learning agent acting in a discrete-time, partially observable, open-world environment. The agent receives observations o_t at each frame. Because the environment is open-world, agents or objects outside the current field of view can enter later. However, in physical or simulated environments with finite movement speeds, the influence of far-away entities can decay over time. This motivates an ε -Markov (or pseudo-Markov) model with an effective horizon H .

Definition 1 (ε -Markov Property). *An environment satisfies the ε -Markov property with horizon H if for the optimal policy's action distribution a_t^* , the following holds:*

$$TV(P(a_t^* \mid o_{t-H}, \dots, o_t), P(a_t^* \mid o_1, \dots, o_t)) \leq \varepsilon,$$

where TV denotes the Total Variation Distance. In words, conditioning on more than the last H frames changes the optimal action distribution by at most ε .

Definition 2 (Decaying Memory / Mixing Time). *Equivalently, there exist constants C and λ such that for any value function V ,*

$$|\mathbb{E}[R_t \mid o_1, \dots, o_t] - \mathbb{E}[R_t \mid o_{t-W}, \dots, o_t]| \leq C e^{-\lambda W}.$$

Implication for the analysis: Under the ε -Markov assumption, restricting the policy to a finite context window changes the optimal action distribution by at most the specified ε in total variation. This provides the finite-horizon context used by the FAP formulation.

4.2 Part A — BPTT vs. FAP Gradient Stability

Let the agent interact with the environment for T discrete frames, $t = 1, \dots, T$. Let o_t denote the observation at frame t (optionally augmented with a short local context window $o_{t-W}, \dots, o_t$) and a_t the true (demonstrated) discrete action.

Fix a prediction horizon H . At frame t , define the target vector

$$z_t = (a_{t+1}, a_{t+2}, \dots, a_{t+H}), \tag{1}$$

i.e., the vector of the next H ground-truth actions. This vector is determined purely by the demonstration data.

A backbone network f_θ (e.g., a CNN or MLP) consumes the observation (and the short local context) and produces a set of predictions:

$$\hat{P}_t^h[a_t] \quad : \text{predicted probability of action } a_t \text{ given observation } o_{t-h}, \quad h = 1, \dots, H.$$

These predictions are grouped into a vector:

$$\hat{z}_t = (\hat{P}_{t+1}^1[a_{t+1}], \hat{P}_{t+2}^2[a_{t+2}], \dots, \hat{P}_{t+H}^H[a_{t+H}]). \tag{2}$$

The backbone is trained with a direct supervised loss for every t independently:

$$L_t(\theta) = \ell(f_\theta(o_t), z_t), \quad (3)$$

where ℓ is any per-frame loss (e.g., focal loss or cross-entropy for discrete actions). The gradient $\partial L_t / \partial \theta$ does **not** involve a chain of hidden-state transitions across frames.

For comparison, consider a recurrent policy with hidden state recurrence $h_{i+1} = \varphi(h_i, o_{i+1}; \theta)$ and a loss L_s evaluated at frame s using h_s :

$$\frac{\partial L_s}{\partial \theta_t} = \frac{\partial L_s}{\partial h_s} \cdot \left(\prod_{i=t}^{s-1} \frac{\partial h_{i+1}}{\partial h_i} \right) \cdot \frac{\partial h_t}{\partial \theta}. \quad (4)$$

Proposition 1 (Norm bound on the BPTT signal). *If $\left\| \frac{\partial h_{i+1}}{\partial h_i} \right\| \leq \rho$ for some constant ρ , then*

$$\left\| \prod_{i=t}^{s-1} \frac{\partial h_{i+1}}{\partial h_i} \right\| \leq \rho^{s-t}. \quad (5)$$

Proof. Submultiplicativity of the operator norm: $\|AB\| \leq \|A\| \cdot \|B\|$. Applying this recursively yields the bound. $\square$

If $\rho < 1$, the bound decays exponentially with temporal distance, while $\rho > 1$ allows exponential growth. This is the mechanism underlying vanishing and exploding gradients in the recurrent setting.

Proposition 2 (FAP’s credit-assignment path has length 1). *For the FAP loss (3),*

$$\frac{\partial L_t(\theta)}{\partial \theta} = \frac{\partial}{\partial \theta} \ell(f_\theta(o_t), z_t). \quad (6)$$

Proof. By construction, z_t is fixed, detached data, so $\partial z_t / \partial \theta = 0$. The only θ -dependent term is $f_\theta(o_t)$, evaluated through a single feed-forward pass. No product of temporal state-transition Jacobians appears. $\square$

Corollary. The temporal Jacobian-product mechanism responsible for the bound in Proposition 1 does not arise in the FAP gradient.

Thus, under the stated ε -Markov assumption, FAP removes the long temporal Jacobian chain from the backbone’s credit-assignment path.

4.3 Part B — Theoretical Guarantee for Aggregation

Part A addresses the temporal credit-assignment path. We now analyze the aggregation of the H predictions and give a no-regret guarantee for the corresponding expert-combination problem.

4.3.1 Theoretical Setting: Hedge / Exponential Weights

Define the per-frame loss of predictor h at frame t as:

$$\mathcal{L}_t^h = 1 - \hat{P}_t^h[a_t], \quad (7)$$

where $\hat{P}_t^h[a_t]$ is the predicted probability of the true action a_t given o_{t-h} . This loss is bounded in $[0, 1]$.

Let w_t^h be the weight assigned to predictor h at frame t , with $\sum_h w_t^h = 1$. The aggregated loss is:

$$\mathcal{L}_t^{agg} = \sum_h w_t^h \mathcal{L}_t^h = 1 - \sum_h w_t^h \hat{P}_t^h[a_t]. \quad (8)$$

The best predictor in hindsight is defined as:

$$best = \arg \min_h \mathbb{E}_{t \sim \text{episode}} [\mathcal{L}_t^h], \quad (9)$$

where $\mathbb{E}_{t \sim \text{episode}}$ denotes the empirical average over all frames in the episode.

The classical Hedge algorithm maintains weights via:

$$w_{t+1}^h = \frac{w_t^h \exp(-\eta \mathcal{L}_t^h)}{\sum_j w_t^j \exp(-\eta \mathcal{L}_t^j)}. \quad (10)$$

Theorem 1 (Hedge Regret Bound). *Let T be the number of frames in the episode. With learning rate $\eta = \sqrt{8 \ln H / T}$, the cumulative loss of the aggregator satisfies:*

$$\sum_{t=1}^T \mathcal{L}_t^{agg} \leq \sum_{t=1}^T \mathcal{L}_t^{best} + \sqrt{\frac{T \ln H}{2}}, \quad (11)$$

where $\mathcal{L}_t^{best}$ denotes the loss of the best predictor as defined in (9).

Dividing both sides by T yields the bound on the expected per-frame loss:

$$\mathbb{E}_{t \sim \text{episode}} [\mathcal{L}_t^{agg}] \leq \mathbb{E}_{t \sim \text{episode}} [\mathcal{L}_t^{best}] + \sqrt{\frac{\ln H}{2T}}. \quad (12)$$

Proof. The result follows from the standard potential-function analysis of Hedge using Hoeffding's lemma. $\square$

4.3.2 Practical Implementation: Offline Transformer + Argmax

Deployment consideration: The Hedge algorithm (10) requires observing the true action a_t at test time to update the weights, which is unavailable during autonomous inference.

In our practical architecture:

1. The **Hedge bound** (12) provides a theoretical reference for the corresponding aggregation problem. Hedge itself is a linear, online combiner, whereas the Transformer is trained offline and can represent nonlinear aggregation functions.
2. In practice, we use an **offline-trained Transformer** (or a small MLP) to approximate the aggregation function on logged demonstration data.
3. At inference time, because the action space is discrete, we select the final action using the **argmax** over the aggregated logits or probabilities produced by the Transformer.

This separates the theoretical online-learning reference from the practical offline aggregation mechanism used for deployment.

4.4 Part C — Variance Analysis of the Hedge Weighted Aggregation

We now analyze the variance of the aggregation strategy defined in Part B using the same notation. The weights w_t^h are those produced by the Hedge update rule (Equation 10). At any round t , these weights define the aggregated prediction. Let the **best single predictor** be the one that achieves the minimum cumulative loss $\sum_{t=1}^T \mathcal{L}_t^{best}$ in the Hedge analysis. Denote its index by $best$.

We define the variance of the best predictor's loss over the episode as:

$$\text{Var}(\mathcal{L}^{best}) = \text{Var}_{t \sim \text{episode}} \left(\hat{P}_t^{best}[a_t] \right), \quad (13)$$

where $\hat{P}_t^{best}[a_t]$ is the probability assigned to the true action by the best predictor.

Similarly, the variance of the aggregated loss is:

$$\text{Var}(\mathcal{L}^{agg}) = \text{Var}_{t \sim \text{episode}} \left(\sum_h w_t^h \hat{P}_t^h[a_t] \right). \quad (14)$$

Using the definition of variance over the episode, we have:

$$\text{Var}(\mathcal{L}^{agg}) = \sum_h \sum_k \text{Cov}_{t \sim \text{episode}} \left(w_t^h \hat{P}_t^h[a_t], w_t^k \hat{P}_t^k[a_t] \right). \quad (15)$$

If the weights are fixed ($w_t^h = w^h$), this simplifies to:

$$\text{Var}(\mathcal{L}^{agg}) = \sum_h \sum_k w^h w^k \text{Cov}_{t \sim \text{episode}} \left(\hat{P}_t^h[a_t], \hat{P}_t^k[a_t] \right). \quad (16)$$

From (13) and (16), we can compare $\text{Var}(\mathcal{L}^{agg})$ and $\text{Var}(\mathcal{L}^{best})$. The aggregated variance (16) is a weighted sum of covariances between predictions.

A sufficient (but not necessary) condition for variance reduction is that the covariances between different predictors ($h \neq k$) are sufficiently small relative to their variances, together with weights that are not concentrated on a single predictor. In particular, if:

1. $\text{Cov}(\hat{P}_t^h, \hat{P}_t^k) \approx 0$ for $h \neq k$, and
2. $\sum_h (w^h)^2 < 1$,

then the cross-covariance contribution is limited and the aggregate variance can be smaller than that of an individual predictor. The precise comparison depends on the individual predictor variances and the covariance structure.

However, this is not the only way variance can be reduced. Even when covariances are non-zero, an aggregation can achieve lower variance than an individual predictor when the weights appropriately balance variance and covariance terms.

The correlation structure can be measured on the demonstration dataset. In practice, we compute:

$$\bar{\rho} = \frac{1}{H(H-1)} \sum_{h \neq k} \text{Corr}_{t \sim \text{episode}} \left(\hat{P}_t^h[a_t], \hat{P}_t^k[a_t] \right), \quad (18)$$

where Corr is the Pearson correlation coefficient. Values below 1 indicate that the predictors are not perfectly correlated, although $\bar{\rho} < 1$ alone is not sufficient to establish variance reduction without considering the predictor variances and the weighting scheme.

4.5 Summary of Theoretical Results

Under the stated ε -Markov (pseudo-Markovian) assumption for discrete, open-world environments:

1. **Stability (Part A):** BPTT involves a temporal product of recurrent Jacobians whose norm admits the geometric bound ρ^{s-t} . In contrast, the FAP per-frame loss has no temporal Jacobian product, so its credit-assignment path does not extend across the sequence through a recurrent state.
2. **Theoretical aggregation reference (Part B):** The aggregation of H overlapping predictions admits the Hedge regret bound in Equation (12). In practice, the Transformer is trained offline to approximate a suitable aggregation function and uses $\arg \max$ for discrete action selection.
3. **Variance analysis (Part C):** The bias-variance analysis does not require identical per-predictor statistics or monotonicity with horizon. Under suitable covariance and weighting conditions, aggregation can reduce variance; when the predictors are perfectly correlated, aggregation provides no diversification benefit.

Together, these results provide a theoretical basis for separating future-action prediction from downstream temporal aggregation in pseudo-Markovian interactive environments, while the empirical section evaluates the resulting architecture under the StrikeForce benchmark.

5 Environment: StrikeForce

StrikeForce is a 2D battle-arena environment featuring partial observability, fog of war, and stochastic enemy policies [14]. The agent observes a local 31×31 grid with 32 feature channels. The action space has 9 discrete actions (in this research models will only have 7 actions). Demonstrations are collected from a human player. The stochastic adversary design provides a setting in which temporal information can decay over time, consistent with the pseudo-Markovian modeling assumption used in the theoretical analysis.

6 Experimental Design

6.1 Models Under Comparison

Because of the stop-gradient boundary described in Section 2.3, the FAP backbone is trained exactly once via the backbone loss $\mathcal{L}^{\text{BB}}$, independently of the aggregation strategy applied on top of it. Two of the three comparisons below therefore require *no additional training run*: they reuse a single fixed, already-trained backbone checkpoint and differ only in which aggregation module consumes its per-frame outputs at inference and evaluation time. We consider three models to isolate the contribution of the aggregation stage while holding the backbone fixed:

- **M1 — No Aggregator (Head-0 Direct):** the policy action at frame t is taken directly as $\arg \max_k \hat{P}_t^0[k]$, i.e., the $\arg \max$ of prediction head $h = 0$ under the 0-indexed convention $h \in \{0, \dots, H - 1\}$. No aggregator or learned temporal fusion is used. This isolates the contribution of the aggregation stage by using a single per-frame FAP head.
- **M2 — Transformer Aggregator:** the full architecture of Section 2.2, with `tf_policy` attending over the sliding window of detached backbone tokens, raw per-frame status features, and action history.

- **M3 — Linear Aggregator:** a learned linear aggregation of the same token sequence consumed by M2. At frame t , the aggregator concatenates the window of tokens $[z_{t-H+1}, \dots, z_t] \in \mathbb{R}^{H \times d}$ into a single vector and maps it linearly to action logits:

$$\text{logits}_t = W \cdot \text{vec}([z_{t-H+1}, \dots, z_t]) + b,$$

with $W \in \mathbb{R}^{K \times (Hd)}$ and $b \in \mathbb{R}^K$, where $d = 256$ is the per-token dimensionality and $K = 7$ is the number of actions. M3 is trained with the same focal loss ($\gamma_{\text{focal}} = 2$) and the same logged demonstrations as M2, and uses $\arg \max$ at inference. This is the strongest linear aggregation of the same token inputs and isolates the contribution of nonlinear, input-dependent attention in M2.

M1 is an evaluation-time construction requiring no additional training; M3 is a learned aggregator trained on the same data as M2; M2 is the full architecture. The stop-gradient boundary allows all three to share the same fixed backbone checkpoint without retraining it. M2 and M3 are trained on the same logged demonstration set, consisting of 16 episodes of $T = 1024$ frames each, while M1 is an evaluation-time construction that reuses the same fixed backbone checkpoint without additional training.

6.2 Excluded Comparisons: Full BPTT and Truncated BPTT

We do *not* include a no-stop-gradient (full BPTT) variant in our empirical comparisons, nor do we attempt to train one. This exclusion follows from the computational and architectural consequences of removing the stop-gradient boundary:

1. **The sliding window does not bound the backward graph.** WINDOW_SIZE=31 bounds the number of tokens visible to the aggregator at a given step, but it does not bound the autograd graph if `A_vec.detach()` is removed. Each token in the window is itself produced by a full backbone forward pass on its corresponding frame; removing an older token from the sliding-window deque therefore does not detach its computational graph. Because `tf_policy` mixes the tokens through attention, and the backbone does not introduce an independent episode-boundary reset, the backward graph remains transitively connected to earlier frames. The resulting computation is therefore effectively unbounded through the episode, rather than being limited to a 31-frame backward horizon. For an episode of $T = 1024$ frames, this approaches full-episode BPTT in both graph depth and memory requirements.
2. **The resulting memory requirement is far beyond our hardware budget.** In our measurements, a single forward pass of the backbone requires approximately 200MB of memory. Retaining the corresponding computation graphs across $T = 1024$ frames would therefore place the naive full-BPTT execution in the order of $1024 \times 200\text{MB} \approx 200\text{GB}$, which is far beyond the memory available on our hardware. The resulting regime is therefore infeasible for the experimental system.
3. **A partitioned-backward alternative remains impractical and changes the empirical setting.** We also considered a more memory-efficient scheduling strategy in which `backward()` is called only for frames belonging to one residue class modulo 31, gradients are accumulated, and the procedure is repeated for all 31 residue classes. This reduces the estimated peak memory requirement to roughly 12GB. Although this is substantially smaller than the approximately 200GB naive full-BPTT regime, it remains much larger than the approximately 200MB memory footprint of a single forward pass and requires substantially

more memory and computational resources than are available in our experimental setup. The resulting $\sim 12\text{GB}$ peak requirement is therefore also outside the feasible training regime for the hardware used in this study.

Moreover, partitioning the backward passes does not convert the computation into the length-1 credit-assignment path analyzed for FAP. Within each partition, frames connected through the attention window still introduce genuine cross-frame Jacobian products. The resulting gradient therefore remains within the BPTT family but corresponds to a different computational schedule and would require a separate theoretical treatment.

Consequently, we analyze the vanishing/exploding-gradient mechanism in Part A mathematically rather than through an empirical full-BPTT training run. Under the hardware constraints of this study, both the naive full-BPTT implementation and the approximately 12GB partitioned alternative are outside the feasible training regime.

We also do not report a truncated BPTT (TBPTT) baseline with a matched window of $k = 31$ frames. With $k = 31$, a single backward graph retains 31 frame activations plus the aggregator’s attention graph. On the available hardware (a single consumer GPU), the measured peak memory requirement for this configuration exceeds the available VRAM even with gradient checkpointing and gradient accumulation applied, and the training time is correspondingly infeasible for a sole-author study without dedicated compute. Smaller windows such as $k = 8$ or $k = 16$ do not preserve the matched-horizon comparison that the FAP formulation requires, and a mismatched comparison would confound the aggregation mechanism with the effective memory length. We therefore rely on the theoretical analysis of Part A to characterize the BPTT mechanism, and leave an empirical matched-window comparison to future work with sufficient compute.

6.3 Evaluation Protocol

Evaluation uses a fixed set of $n = 64$ paired episodes, with the heuristic inference-time action mask (`bc_inference`) applied uniformly to all three models. The same random seed, map, and initial condition are used within each paired episode, so that differences between models within a pair are attributable to the aggregation strategy rather than to environment stochasticity. All results are reported as paired comparisons with $n = 64$ and are corrected for multiple comparisons using Benjamini–Hochberg FDR control over all $(3 \text{ pairs}) \times (14 \text{ metrics})$ tests jointly. The analysis is implemented in the accompanying `compare_models.py` script, which also generates all LaTeX tables and figures referenced below.

- **Mask policy: uniform, always on.** The heuristic inference-time action mask (`bc_inference`) is applied to all three models; no mask-off condition is evaluated. This keeps the comparison at the level of the deployed system and holds inference-time post-processing fixed across models.
- **Episode count.** Each model is evaluated over $N_{\text{episodes}} = 64$ episodes.

Why a paired design and not aggregate averages. The no-aggregator baseline (M1) exhibits high per-episode variance in survival time (sample standard deviation of 1.08×10^4 against a mean of 2.01×10^4 , i.e. a coefficient of variation above 0.5). Unpaired comparisons at this variance level cannot reliably distinguish policy differences from episode-to-episode environment randomness. Pairing on identical initial conditions removes this shared variance and allows the paired t -test or Wilcoxon signed-rank test to detect policy-level effects that would otherwise be masked at this sample size.

6.4 Metrics

We use three groups of metrics. All metrics are oriented so that larger values indicate a better policy; none of the metrics is an accumulator that grows with episode length in a way that would trivially reward longer survival.

Base metrics. The five quantities extracted from each episode log are ¹

$$N_{\text{kills}} \text{ (kills), } T_{\text{alive}} \text{ (frames alive), } A_{\text{HP}} \text{ (area under HP curve),}$$

$$A_{\text{stamina}} \text{ (area under stamina curve), } A_{\text{damage}} \text{ (area under damage-per-bullet curve).}$$

These are the raw quantities on which the derived metrics below are built. They are reported for completeness but are not used as primary evidence on their own: A_{HP} , A_{stamina} , and A_{damage} are accumulators and therefore correlate with T_{alive} ; N_{kills} and T_{alive} are the volume measures on which the aggregation comparison is defined.

Derived metrics. Six normalized quantities are computed from the base metrics:

$$S_1 = \frac{N_{\text{kills}}}{T_{\text{alive}}}, \quad S_2 = \frac{N_{\text{kills}}}{A_{\text{stamina}}}, \quad S_3 = \frac{N_{\text{kills}}^2}{T_{\text{alive}} A_{\text{damage}}},$$

$$r_{\text{HP}} = \frac{A_{\text{HP}}}{T_{\text{alive}}}, \quad r_{\text{stam}} = \frac{A_{\text{stamina}}}{T_{\text{alive}}}, \quad r_{\text{dmg}} = \frac{A_{\text{damage}}}{T_{\text{alive}}}.$$

S_1 is the kill rate, S_2 is kills per unit of accumulated stamina (an engagement-efficiency measure), and S_3 is a damage-normalized kill rate. r_{HP} , r_{stam} , and r_{dmg} are the per-frame HP, stamina, and damage rates; larger values indicate a policy that maintains a higher combat capacity per unit time. All six are higher-is-better.

Composite metrics. Three kill-weighted resource metrics are defined as

$$M_{\text{HP}} = \frac{N_{\text{kills}} A_{\text{HP}}}{T_{\text{alive}}^2}, \quad M_{\text{stam}} = \frac{N_{\text{kills}} A_{\text{stamina}}}{T_{\text{alive}}^2}, \quad M_{\text{dmg}} = \frac{N_{\text{kills}} A_{\text{damage}}}{T_{\text{alive}}^2}.$$

Each metric combines the kill count with a resource accumulator and normalizes by squared episode duration, which penalizes duration more strongly than a single-time normalization: when a resource accumulator grows approximately linearly with time, the metric behaves roughly like kills per unit time multiplied by the average resource level per unit time. It therefore measures a duration-normalized engagement-resource trade-off, not total accumulated combat output. A policy that survives for a long time while producing few kills is penalized, while a policy that produces kills and maintains resource capacity can score well. This interpretation is useful for comparing policies under the same episode protocol, but the metric is a designed proxy rather than a universally valid measure of policy quality. We consequently report it alongside the base and derived metrics rather than treating it as standalone proof of aggregation quality.

¹<https://github.com/bistoyek21-ric/StrikeForce/blob/main/StrikeForce-client/StrikeForce-Lab-Protocol.md>

6.5 Results

The evaluation contains 64 valid paired episodes for each model (M1, M2, and M3), with no malformed data blocks. Composite metrics are computed per episode using a squared-time denominator (T_{alive}^2).

Metric	M1 (No Aggregator)	M2 (Transformer)	M3 (Linear)
T_{alive}	20059.7 ± 10820.0	11908.0 ± 6714.0	13608.8 ± 8498.0
N_{kills}	236.6 ± 120.4	184.1 ± 100.0	170.0 ± 105.5
A_{HP}	$(2.342 \pm 2.177) \times 10^8$	$(1.032 \pm 0.680) \times 10^8$	$(1.413 \pm 1.552) \times 10^8$
A_{stamina}	$(1.853 \pm 0.684) \times 10^8$	$(1.214 \pm 0.456) \times 10^8$	$(1.383 \pm 0.594) \times 10^8$
A_{damage}	$(4.436 \pm 3.964) \times 10^7$	$(2.277 \pm 1.739) \times 10^7$	$(2.579 \pm 2.480) \times 10^7$

Table 1: Base metrics, mean $\pm$ standard deviation across 64 episodes.

Metric	M1	M2	M3
$S_1 = N_{\text{kills}}/T_{\text{alive}}$	0.0121077	0.0156579	0.0125231
$S_2 = N_{\text{kills}}/A_{\text{stamina}}$	1.23764×10^{-6}	1.45085×10^{-6}	1.16798×10^{-6}
$S_3 = N_{\text{kills}}^2/(T_{\text{alive}}A_{\text{damage}})$	8.42256×10^{-8}	1.46705×10^{-7}	9.88703×10^{-8}
$r_{\text{HP}} = A_{\text{HP}}/T_{\text{alive}}$	10459.3	8540.28	9275.09
$r_{\text{stam}} = A_{\text{stamina}}/T_{\text{alive}}$	10059.6	11021.9	10911.5
$r_{\text{dmg}} = A_{\text{damage}}/T_{\text{alive}}$	1976.62	1800.78	1703.08

Table 2: Derived metrics. Entries are means of the episode-level metrics; bold marks the largest mean in each row.

Metric	M1	M2	M3
$M_{\text{HP}} = N_{\text{kills}}A_{\text{HP}}/T_{\text{alive}}^2$	124.875	133.708	115.443
$M_{\text{stam}} = N_{\text{kills}}A_{\text{stamina}}/T_{\text{alive}}^2$	122.446	173.693	137.149
$M_{\text{dmg}} = N_{\text{kills}}A_{\text{damage}}/T_{\text{alive}}^2$	23.9833	28.7439	21.6934
Mean relative composite score	0.8244	1.0000	0.8026

Table 3: Composite metrics, computed with T_{alive}^2 in the denominator. The relative score is the mean of each model’s three composite metrics normalized by the across-model maximum for each metric.

6.6 Paired Statistical Analysis

All pairwise tests use the 64 matched episodes. The analysis reports paired t -tests or Wilcoxon signed-rank tests as appropriate, and Benjamini–Hochberg correction across the 42 tests (three pairs by 14 metrics). The table below highlights the principal corrected results; the complete test summaries are in Appendix Tables 5–7.

Pair	Metric (difference is second minus first)	Δ	q_{FDR}	d_z
M1 vs M2	S_1 (M2–M1)	+0.00355	< 0.001	+1.27
M1 vs M2	M_{stam} (M2–M1)	+51.25	< 0.001	+1.10
M1 vs M2	S_3 (M2–M1)	$+6.25 \times 10^{-8}$	< 0.001	+1.09
M1 vs M2	A_{stamina} (M2–M1)	-6.39×10^7	< 0.001	-0.78
M1 vs M2	T_{alive} (M2–M1)	-8151.72	< 0.001	-0.63
M1 vs M2	N_{kills} (M2–M1)	-52.48	0.0198	-0.32
M1 vs M3	A_{stamina} (M3–M1)	-4.71×10^7	< 0.001	-0.51
M1 vs M3	T_{alive} (M3–M1)	-6450.91	< 0.001	-0.48
M1 vs M3	N_{kills} (M3–M1)	-66.61	0.0026	-0.42
M1 vs M3	A_{damage} (M3–M1)	-1.86×10^7	0.0011	-0.42
M1 vs M3	r_{stam} (M3–M1)	+851.89	0.0026	+0.39
M2 vs M3	S_1 (M3–M2)	-0.00313	< 0.001	-1.22
M2 vs M3	S_3 (M3–M2)	-4.78×10^{-8}	< 0.001	-0.99
M2 vs M3	S_2 (M3–M2)	-2.83×10^{-7}	< 0.001	-0.80
M2 vs M3	M_{stam} (M3–M2)	-36.54	< 0.001	-0.72
M2 vs M3	M_{dmg} (M3–M2)	-7.05	< 0.001	-0.53

Table 4: Selected paired comparisons. Positive Δ means the second model has a larger value; the interpretation as an improvement depends on the metric.

Figure 1 visualizes the same corrected results across all fourteen metrics and all three pairwise comparisons, with 95% confidence intervals on the paired mean difference and a filled/open marker convention for corrected significance. The summary rows of Table 4 highlight the principal effects; the figure shows the distribution of effect sizes and the location of the Benjamini–Hochberg threshold across the full set of 42 tests.

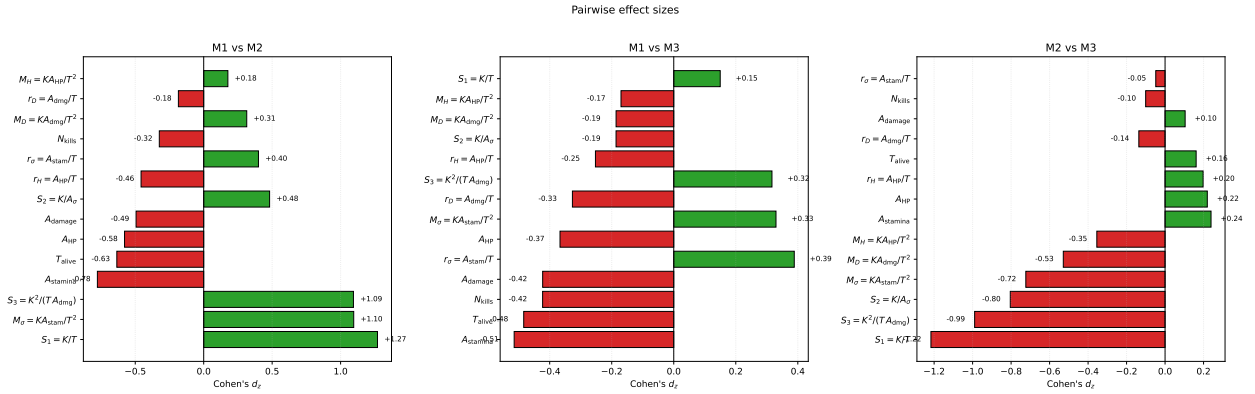


Figure 1: Paired effect sizes with 95% confidence intervals on the paired mean difference $\Delta = M_2 - M_1$, for all fourteen metrics across the three pairwise comparisons (M1 vs M2, M1 vs M3, M2 vs M3). Filled markers indicate $q_{\text{FDR}} < 0.05$ after Benjamini–Hochberg correction across the 42 pair-by-metric tests of Section 6.6; open markers indicate comparisons that are not significant after correction.

6.7 Interpretation of Results

The Transformer aggregator improves efficiency-oriented metrics, while the no-aggregator baseline retains higher raw volume. Reading Tables 2 and 3 jointly, M2 has the highest mean value on all six derived rate metrics that measure engagement per unit of time or resource, and on all three composite metrics. Relative to M1, M2 increases the kill rate S_1 from 0.01211 to 0.01566 ($q < 0.001$, $d_z = 1.27$), the kills-per-stamina ratio S_2 from 1.2376×10^{-6} to 1.4509×10^{-6} ($q < 0.001$, $d_z = 0.48$), the damage-normalized kill output S_3 from 8.42×10^{-8} to 1.47×10^{-7} ($q < 0.001$, $d_z = 1.09$), and the kill-weighted stamina composite M_{stam} from 122.45 to 173.69 ($q < 0.001$, $d_z = 1.10$). The gains are +29.3% for S_1 , +17.2% for S_2 , +74.2% for S_3 , and +41.8% for M_{stam} . By contrast, M1 retains higher raw volume on the base metrics: mean kills (236.6 vs. 184.1, $q = 0.0198$, $d_z = -0.32$), survival time (20059.7 vs. 11908.0 frames, $q < 0.001$, $d_z = -0.63$), and accumulated stamina (1.853×10^8 vs. 1.214×10^8 , $q < 0.001$, $d_z = -0.78$). M1 also retains a higher per-frame HP rate r_{HP} (10459.3 vs. 8540.3, $q = 0.0010$, $d_z = -0.46$).

The pattern is therefore not that the Transformer aggregator strictly dominates the direct head-0 baseline; it is that aggregation shifts the policy from a longer-survival, lower-efficiency profile to a shorter-survival, higher-efficiency one. Under the episode protocol used here, both profiles are Pareto-optimal on the kills-vs-composite-damage plane (Section 6.7, Pareto paragraph), and the choice between them is a choice of operating point rather than a choice between a better and a worse policy.

Figure 2 summarizes the resulting profile across the six engagement-oriented metrics that show corrected significant differences in at least two of the three pairwise comparisons (S_1 , S_2 , S_3 , r_{stam} , M_{stam} , M_{dmg}). Because each axis is min-max normalized across the three models, the figure is a descriptive profile rather than an inferential test; the corrected q -values reported above and in the appendix remain the basis for the statistical claims.

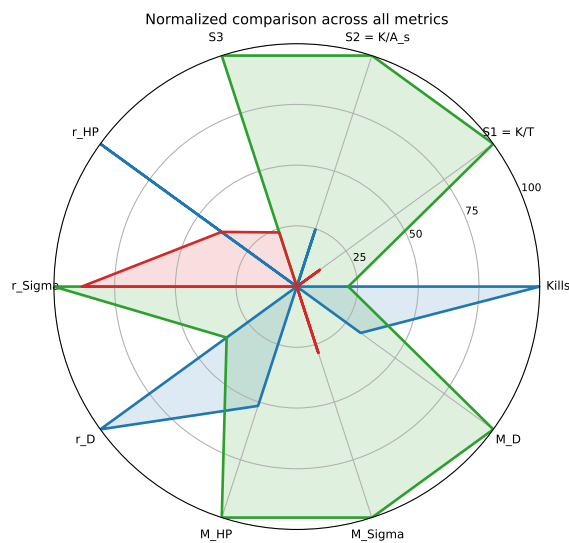


Figure 2: Normalized profile of the three models over the six engagement-oriented metrics that show corrected significant differences in at least two of the three pairwise comparisons (S_1 , S_2 , S_3 , r_{stam} , M_{stam} , M_{dmg} ; Section 6.6). Each axis is min-max normalized across the three models, so that the best model on a given axis scores 1 and the worst scores 0. The figure is descriptive and is not used as an inferential test.

The linear aggregator does not capture the benefit of nonlinear attention and is dominated by both M1 and M2. M3 has lower mean kills, survival time, accumulated HP, accumulated stamina, and accumulated damage than M1, and most of these differences are corrected-significant: T_{alive} ($q < 0.001$, $d_z = -0.48$), A_{stamina} ($q < 0.001$, $d_z = -0.51$), N_{kills} ($q = 0.0026$, $d_z = -0.42$), A_{damage} ($q = 0.0011$, $d_z = -0.42$), and A_{HP} ($q < 0.001$, $d_z = -0.37$). M3 does show corrected-significant gains over M1 on r_{stam} ($q = 0.0026$, $d_z = +0.39$), M_{stam} ($q = 0.0192$, $d_z = +0.33$), and S_3 ($q = 0.0223$, $d_z = +0.32$), but these gains are smaller in effect size than the M2 efficiency gains and do not extend to the kill rate S_1 ($q = 0.2743$) or to the composite HP metric M_{HP} ($q = 0.2142$).

Compared with M2, the linear aggregator is markedly worse on every efficiency-oriented metric: S_1 ($q < 0.001$, $d_z = -1.22$), S_3 ($q < 0.001$, $d_z = -0.99$), S_2 ($q < 0.001$, $d_z = -0.80$), M_{stam} ($q < 0.001$, $d_z = -0.72$), M_{dmg} ($q < 0.001$, $d_z = -0.53$), and M_{HP} ($q = 0.0039$, $d_z = -0.35$). The raw-volume comparisons between M2 and M3 are not corrected-significant (N_{kills} : $q = 0.1903$; T_{alive} : $q = 0.5155$; A_{HP} : $q = 0.4758$; A_{stamina} : $q = 0.0877$; A_{damage} : $q = 0.8462$), so M2 and M3 have comparable raw volume, but M2 achieves it with substantially better efficiency. Under the present evaluation, the learned linear aggregation used by M3 does not improve performance over the direct single-head baseline, and it is Pareto-dominated by both M1 and M2.

Pareto comparison and overall summary. Table 3 places the three models in the two-dimensional plane of mean raw kills and composite damage M_{dmg} . Reading the two columns jointly, M3 is Pareto-dominated: it is lower than M1 on mean kills (170.0 vs. 236.6) and lower than M2 on M_{dmg} (21.69 vs. 28.74), so no reader can prefer M3 on either axis. Both M1 and M2 are Pareto-optimal: M1 has more mean kills (236.6 vs. 184.1), whereas M2 has a larger M_{dmg} (28.74 vs. 23.98). The normalized mean composite scores are 0.8244 for M1, 1.0000 for M2, and 0.8026 for M3. These scores are descriptive summaries, not inferential tests; they aggregate the three composite metrics by min-max normalization and do not supersede the paired comparisons above. Figure 3 visualizes this comparison.

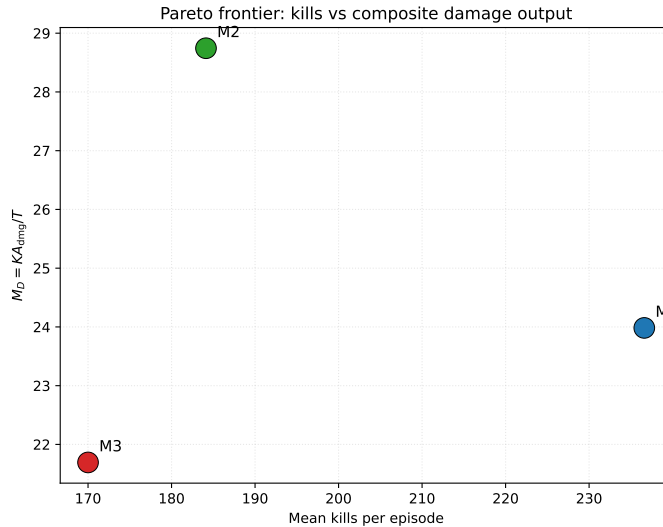


Figure 3: Pareto frontier in the plane of mean kills per episode and composite damage $M_{\text{dmg}} = N_{\text{kills}}A_{\text{damage}}/T_{\text{alive}}^2$. M3 is dominated by both M1 and M2; both M1 and M2 are Pareto-optimal, with M1 producing more mean kills and M2 a larger M_{dmg} . This figure summarizes the two-dimensional comparison described in the surrounding text and is descriptive rather than inferential.

Limitations. The experiment evaluates three models on 64 paired episodes in one environment and uses episode-level proxy metrics rather than a direct behavioral-quality measure. The findings support claims about these measured metrics under this evaluation protocol; they do not establish universal ordering of aggregation methods or rule out regressions on unmeasured behaviors. The relatively large per-episode variance on the base metrics (e.g. A_{HP} for M1 has standard deviation comparable to its mean) means that the raw-volume differences, while corrected-significant for several metrics, are estimated with substantial uncertainty; the effect sizes reported in the paired analyses should be preferred over the raw means as a summary of magnitude. The figures in this section visualize the same measurements that the tables and paired test summaries report; they are not independent measurement runs and do not carry additional statistical information beyond what the tables and their corrected q -values already provide.

7 Related Work

The theoretical and architectural foundations of our work span several active research areas. The Transformer architecture [1] provides the backbone for our bounded-window aggregator. Our decoupled training strategy, where the perception backbone is trained with a stop-gradient from the policy objective, draws inspiration from self-supervised and contrastive representation learning methods [9, 10, 11]. The use of auxiliary future prediction as a learning signal is central to world models [5, 6, 7] and contrastive predictive coding [8].

From a sequential modeling perspective, our work relates to efficient long-sequence architectures [12, 13], while explicitly avoiding recurrent BPTT in the backbone. The theoretical analysis of our aggregator leverages no-regret online learning [15, 16]. Finally, our problem setting—imitation learning in partially observable domains—builds upon classical results in behavioral cloning [17] and POMDP theory [18].

8 Scope Statement

We trade backbone BPTT’s memory cost and temporal gradient coupling for a fixed-size attention window, sufficient when effective history length is bounded by H under the ε -Markov modeling assumption.

Explicitly not claimed:

- That the backbone recovers the full hidden state from a single frame.
- That gradients from the policy objective cross time into the backbone.
- That the model extrapolates beyond the trained horizon H .
- That the ε -Markov assumption holds exactly in every environment; it is a modeling idealisation and is not established here as an exact property of StrikeForce.
- That H can be derived purely from theory; H is selected via validation on demonstration data.
- **That the frame-wise predictors have identical bias, variance, or any monotonic relationship.** The theoretical analysis does not rely on these assumptions. Variance reduction depends on the covariance structure and the aggregation weights.

- **That the Transformer’s performance is exactly captured by the linear analysis.** The linear analysis provides a theoretical reference for the aggregation problem; the Transformer is evaluated empirically as a nonlinear alternative.
- **That the vanishing/exploding-gradient analysis of Part A is empirically re-evaluated with a trained no-stop-gradient full-BPTT model on StrikeForce.** Such a run is outside the feasible training regime under the hardware constraints described in Section 6.2.
- That the base, derived, and composite metrics are computed from independent measurement runs. All three metric groups are aggregated from the same $n = 64$ paired evaluation episodes described in Section 6.3; the derived and composite groups are deterministic transformations of the base group and are not independent measurements.
- That the figures in Section 6 establish statistical significance on their own. They are descriptive visualizations of the tables and the paired test summaries; inferential claims are supported only by the tables and their associated corrected q -values.

A Full Paired Statistical Results

For each comparison, Δ is the second model minus the first. q_{FDR} denotes the Benjamini–Hochberg adjusted p -value across the 42 pair-by-metric tests. The wins column counts episode pairs favoring the first and second model, respectively. Composite metrics in these tables use T_{alive}^2 .

Metric ($\Delta = M_2 - M_1$)	Δ mean	q_{FDR}	d_z	Wins M1:M2
N_{kills}	−52.4844	0.0198	−0.32	43:21
T_{alive}	−8151.72	< 0.001	−0.63	48:16
A_{HP}	-1.310×10^8	< 0.001	−0.58	50:14
A_{stamina}	-6.394×10^7	< 0.001	−0.78	49:15
A_{damage}	-2.160×10^7	< 0.001	−0.49	48:16
S_1	+0.00355024	< 0.001	1.27	6:58
S_2	$+2.13206 \times 10^{-7}$	< 0.001	0.48	20:44
S_3	$+6.24797 \times 10^{-8}$	< 0.001	1.09	7:57
r_{HP}	−1919.06	0.0010	−0.46	43:21
r_{stam}	+962.318	0.0041	0.40	22:42
r_{dmg}	−175.844	0.1892	−0.18	35:29
M_{HP}	+8.83302	0.2008	0.18	25:39
M_{stam}	+51.2473	< 0.001	1.10	7:57
M_{dmg}	+4.76069	0.0225	0.31	23:41

Table 5: Full M1–M2 paired comparison.

Metric ($\Delta = M_3 - M_1$)	Δ mean	q_{FDR}	d_z	Wins M1:M3
N_{kills}	−66.6094	0.0026	−0.42	45:19
T_{alive}	−6450.91	< 0.001	−0.48	44:20
A_{HP}	-9.286×10^7	< 0.001	−0.37	44:20
A_{stamina}	-4.705×10^7	< 0.001	−0.51	43:21
A_{damage}	-1.857×10^7	0.0011	−0.42	44:20
S_1	+0.000415406	0.2743	0.15	32:32
S_2	-6.96663×10^{-8}	0.1892	−0.19	40:24
S_3	$+1.46447 \times 10^{-8}$	0.0223	0.32	25:39
r_{HP}	−1184.25	0.0715	−0.25	40:24
r_{stam}	+851.892	0.0026	0.39	21:43
r_{dmg}	−273.545	0.0196	−0.33	40:24
M_{HP}	−9.43189	0.2142	−0.17	41:23
M_{stam}	+14.7035	0.0192	0.33	28:36
M_{dmg}	−2.28983	0.1519	−0.19	40:24

Table 6: Full M1–M3 paired comparison.

Metric ($\Delta = M_3 - M_2$)	Δ mean	q_{FDR}	d_z	Wins M2:M3
N_{kills}	−14.125	0.1903	−0.10	38:25
T_{alive}	+1700.81	0.5155	0.16	28:36
A_{HP}	$+3.815 \times 10^7$	0.4758	0.22	29:35
A_{stamina}	$+1.689 \times 10^7$	0.0877	0.24	30:34
A_{damage}	$+3.023 \times 10^6$	0.8462	0.10	30:34
S_1	−0.00313483	< 0.001	−1.22	55:9
S_2	-2.82872×10^{-7}	< 0.001	−0.80	53:11
S_3	-4.7835×10^{-8}	< 0.001	−0.99	58:6
r_{HP}	+734.804	0.3247	0.20	32:32
r_{stam}	−110.425	0.7174	−0.05	30:34
r_{dmg}	−97.7012	0.3175	−0.14	38:26
M_{HP}	−18.2649	0.0039	−0.35	44:20
M_{stam}	−36.5438	< 0.001	−0.72	48:16
M_{dmg}	−7.05052	< 0.001	−0.53	46:18

Table 7: Full M2–M3 paired comparison.

References

- [1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. *Attention is All you Need*. In NeurIPS, 2017.
- [2] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. *Focal Loss for Dense Object Detection*. In ICCV, 2017.
- [3] I. Loshchilov and F. Hutter. *Decoupled Weight Decay Regularization*. In ICLR, 2019.
- [4] M. Jaderberg, V. Mnih, W. M. Czarnecki, et al. *Reinforcement Learning with Unsupervised Auxiliary Tasks*. In ICLR, 2017.
- [5] D. Ha and J. Schmidhuber. *World Models*. arXiv:1803.10122, 2018.
- [6] D. Hafner, T. Lillicrap, I. Fischer, et al. *Learning Latent Dynamics for Planning from Pixels*. In ICML, 2019.
- [7] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi. *Dream to Control: Learning Behaviors by Latent Imagination*. In ICLR, 2020.
- [8] A. van den Oord, Y. Li, and O. Vinyals. *Representation Learning with Contrastive Predictive Coding*. arXiv:1807.03748, 2018.
- [9] J.-B. Grill, F. Strub, F. Altché, et al. *Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning*. In NeurIPS, 2020.
- [10] X. Chen and K. He. *Exploring Simple Siamese Representation Learning*. In CVPR, 2021.
- [11] A. Stooke, K. Lee, P. Abbeel, and M. Laskin. *Decoupling Representation Learning from Reinforcement Learning*. In ICML, 2021.
- [12] A. Gu, K. Goel, and C. Ré. *Efficiently Modeling Long Sequences with Structured State Spaces*. In ICLR, 2022.
- [13] A. Gu and T. Dao. *Mamba: Linear-Time Sequence Modeling with Selective State Spaces*. arXiv:2312.00752, 2023.
- [14] K. Fouladi. *StrikeForce: A 2D Partially Observable Battle-Arena Testbed for Imitation Learning and Multi-Agent Research*. ResearchGate, 2026. doi:10.13140/RG.2.2.24446.96326. <https://doi.org/10.13140/RG.2.2.24446.96326>
- [15] S. Ross, G. Gordon, and D. Bagnell. *A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning*. In AISTATS, 2011.
- [16] Y. Freund and R. E. Schapire. *A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting*. Journal of Computer and System Sciences, 1997.
- [17] D. A. Pomerleau. *Efficient Training of Artificial Neural Networks for Autonomous Navigation*. Neural Computation, 1991.
- [18] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. *Planning and Acting in Partially Observable Stochastic Domains*. Artificial Intelligence, 1998.