Feint Behaviors and Strategies: Formalization, Implementation, and Evaluation
(Preprint Version)

Junyu Liu Xiangjun Peng[‡]

Abstract

Feint behaviors refer to a set of nuanced deceptive behaviors, which enable players temporal and spatial advantages over opponents in competitive games. Such behaviors are crucial tactics in most competitive Multi-Player games (e.g., boxing, fencing, basketball, motor racing, etc.). However, existing literature do not provide comprehensive or concrete formalization for Feint behaviors, and their implications on game strategies. In this paper, we introduce the first comprehensive formalization of Feint behaviors at action-level and strategy-level, and provide concrete implementation and quantitative evaluation in Multi-Player games. The key idea of our work is to (1) allow automatic generation of Feint behaviors via Palindrome-directed templates, and combine them with intended high-reward actions in a Dual-Behavior Model; (2) address Feint implications on game strategies in terms of the temporal, spatial and their collective impacts; and (3) provide a unified implementation scheme of Feint behaviors in existing MARL frameworks. The experimental results show that our design of Feint behaviors can (1) greatly improve the game reward gains; (2) significantly improve the diversity of Multi-Player Games; and (3) only incur negligible overheads in terms of time consumption.

*Note that [‡]refers the affiliation to *The Chinese University of Hong Kong.*

Contents

1 Introduction

In most real-world Multi-Player Games (e.g., boxing, basketball, motor racing, etc.), players have complex behaviors and complicated interactions. Simulating these games usually requires to model the players' behaviors into action spaces at action-level and explore strategies based on them [34, 35]. Among commonly seen behaviors in real-world games, Feint behaviors is a class of tactic behaviors which are used to mislead opponents to gain future strategic advantages. Such behaviors are generally nuanced in terms of movements (e.g., fake overhead punch in boxing, crossover in basketball, early-braking and fake running wide in motor racing, etc.), but could gain huge strategic advantages and increase the games' diversity ([19, 26]). In game simulations, however, current literature general lack a comprehensive or concrete modeling of Feint behaviors in both action-level and strategy-level formalization. [34] mentioned Feint behaviors as a proof-of-concept to construct animations for nuanced game strategies with enhanced unpredictability. More recently, [35] provides a set of pre-defined Feint behaviors for model animation, to optimize game strategies through training and generation via Reinforcement Learning. However, no work provides detailed formalization to address the action-level characteristic of Feint and provide Feint behavior generation guidelines. On the strategy-level, existing learning-based works either neglect Feint behaviors or implicitly assume that they are the same as other behaviors which could have same impacts on strategies through learning. We show that the existing learning-based approaches cannot effectively model Feint behaviors in strategy-level, since Feint behaviors require intricate planning which is an active process.

Our work provides the first comprehensive and concrete formalization of Feint behaviors in action-level and strategy-level. We first present an automatic way to generate Feint behaviors using **Palindrome-directed Templates** based on our observation on Feint characteristics, and provide **Dual-Behavior Model** to showcase the design consideration for combing Feint behaviors and follow-up actions. Based on the action-level formalization, we model the Feint behavior impacts on strategy-level in terms of the temporal, spatial, and their collective impacts under a learnable scheme. Then, we provide a concrete and unified implementation to incorporate the action-level and strategy-level formalizations in common Multi-Player Reinforcement Learning (MARL) frameworks to showcase the effectiveness of our formalization.

To provide a unified definition of Feint behavior in both continuous and discrete action space, we highlight the difference between the terms **action** and **behavior** used in our formalization. We use **action** as the minimal unit movement in a unit time step, such as a unit step movement along the X and Y axis in a 2D board game, raising arms for a certain distance in a boxing game, turning steering wheels while applying brakes for a certain degree in a racing game, etc. This definition of action coincides with the commonly used definition of action in general MARL environments, which is intuitive to understand, simulate, and build our formalization of Feint upon it. One may argue that in some game simulations, combat movements like a cross punch are simply considered as one action, but one could always divide those movements into several unified unit-time-step actions to create a unified alignment in terms of time step in games. In terms of **behavior**, we refer to it as a combination of several actions in a sequence (e.g., a cross punch in boxing games). Thus, Feint could be naturally defined as a behavior that uses a sequence of actions to deceive opponents and lead to large reward actions in the near future. We first describe our observation of Feint behaviors' characteristics and introduce our formalization at the action level.

To properly examine the effectiveness of our formalization, we extensively construct a complex and physics-based boxing game as abstraction of some animation-related works [34, 35]. We use a two-player and a 6-player scenario with 4 commonly used MARL models (MADDPG [20], MASAC [11, 13], MATD3 [2], and MAD3PG [4, 6]) to extensively evaluate our formalization. We

also evaluate our formalization of Feint in a stratigic real-game, Alpha Star, to evaluate the game diversity gain introduced by our formalization. The results show that our formalization of Feint could significantly increase the gaming rewards in all scenarios with all 4 MARL models. For the Diversity Gain, our method can increase the exploitation of the search space by 1.98X, measured by the Exploitability metrics. Our implementation scheme only incur less than 5% overheads in terms of per game episode time consumption. We conclude that our formalization of Feint behaviors is effective and practical, significantly increasing players' game rewards and making Multi-Player Games more interesting.

2 Background

2.1 Feint Behaviors in the Real-World and Simulated Games

Feint behaviors are common for human players, as a set of active actions to obtain strategic advantages in real-world games. Examples can include sports games such as boxing, basketball, and motor racing [10, 9, 12], and electronic games such as King of Fighters and Starcraft [33, 5]. Feint behaviors are not simple deceptive behaviors as their goal is to not to gain rewards for themselves but to create temporal and spatial advantages for some short-term follow-up actions. In addition, Feint behaviors have nuanced action formalizations. Though Feint is undoubtedly important in many real-world games, there still lacks a comprehensive formalization of Feint in Multi-Player Game simulations using Non-Player Characters (NPCs). There are only a limited amount of works to tackle this issue. [34] is an early example of incorporating Feint as a proof-of-concept, which focuses on constructing animations for nuanced game strategies for more unpredictability from NPCs. More recently, [36] uses a set of pre-defined Feint action sequences for the animation, which further serves under an optimized version of control strategies based on Online Reinforcement Learning (i.e. in animating combat scenes). However, these prior works (1) lack concrete formalizations of Feint behavior characteristics, which cannot fully unveil the variety of Feint behaviors in action-level; and (2) lack comprehensive explorations of Feint behaviors implications on game strategies, which neglects the potential impacts of fusing effective Feint behaviors into strategies; and (3) solely focus on Two-Player Games, which can not be effectively generalized to multi-player scenarios

2.2 Modeling Behaviors at Action-Level in Game Animation and Simulation

Modeling characters' behaviors (series of actions) in games could be divided into two categories based on the main purpose: animation-driven modeling or simulation-driven modeling, though animation and simulation are inherently closely correlated. Animation-driven methods mainly focus on modeling the behaviors themselves, with goals of producing a variety of nuanced and coherent action sequences. The interactions with the environment (whether physics-based or not) are generally considered after the modeling of the behaviors and are generally simplified to showcase the behaviors themselves. **Patch-based generation** is a direct way for such methods, which directly compose behaviors by combining pre-defined action sequences [35]. This approach is widely adopted in the industry due to its high production efficiency, supported by an extensive amount of animation libraries (e.g. Mixamo [32]) [16, 30, 37]. However, in recent years, **Learning-based generation** dominates the field as they could automatically produce animated behaviors to mimic the styles of learned actions from the training inputs [17, 27]. On the other hand, simulation-driven modeling usually considers the full interactions with the environment in the first hand. These methods generally formalize the behavior modeling process using Reinforcement Learning (RL) based frameworks to fully explore the complicated space of physics-based action modeling [**?**]. In our work, we use a animation-driven modeling with strong physical constraints to describe our observations of Feint behavior characteristics and use the general simulation-driven modeling in MARL schemes for learnable formalization of Feint in action and strategy levels.

2.3 MARL Models at Strategy-Level in Multi-Player Game Simulations

Multi-Agent Reinforcement Learning (MARL) aims to learn optimal policies for agents in a multi-agent environment, which consists of various agent-agent and agent-environment interactions. Many single-agent Reinforcement Learning methods (e.g. DDPG [18], SAC [11], PPO [29] and TD3 [8], D4PG [4]) can not be directly used in multi-agent scenarios, since the rapidly-changing multi-agent

environment can cause highly unstable learning results (evidenced by [20]). Thus, recent efforts on MARL model designs aim to address such an issue. [7] proposes Counterfactual Multi-Agent (COMA) policy gradients, which uses centralised critic to estimate the Q-function and decentralised actors to optimize agents' policies. [20] proposes Multi-Agent Deep Deterministic Policy Gradient (MADDPG), which decreases the variance in policy gradient and instability of Q-function of DDPG in multi-agent scenarios. [13] proposes Multi-Agent Actor-attention Critic (MAAC), which applies attention entropy mechanism to enable effective and scalable policy learning. These models can have varied impacts within a diverse set of scenarios. [6] introduces Multi-agent Distributed Deep Deterministic Policy Gradient (MAD3PG), which extends the D4PG to multi-agent scenarios with distributed critics to enable distributed tracking. [2] proposes Multi-Agent Twin Delayed Deep Deterministic Policy Gradient (MATD3), which integrates twin delayed Q-learning and addressing the overestimation bias in Q-values in a multi-agent setting. Though different MARL models have different design details, they all share the same high-level learning structure. Thus, our goal is to provide a unified scheme to fuse our formalization of Feint behaviors into game simulations that could be learned using common MARL models, enabling effective Feint behaviors impacts regardless of specific design choices of MARL models.

3 Formalizing Feint behavior

We introduce our formalization of Feint behaviors in action level regarding (1) how to automatically generate Feint behavior with templates from common offensive behaviors; and (2) how can the generated Feint behaviors be synergistically combined with follow-up high-reward actions. We first introduce our methodology to automatically generate Feint behaviors, by exploiting our newly-revealed insight called **Palindrome-directed Generation of Feint Templates**. Next, we illustrate key design choices on how to combine the generated Feint behaviors with follow-up actions in a **Double-Behavior Model**, which forms the foundation for the designs of Feint -accounted strategy designs in Section 4. We choose boxing game as an example to concretely explain our insights for Feint behaviors in this section but our formalization is a unified abstraction of common games and could be easily adapted to other games including basketball, fencing, motor racing, etc.

3.1 Feint behavior characteristics and templates

Common behaviors in boxing games include hook, punch, and block, which generally require different time steps to accomplish. To provide a unified, aligned, and fine-grained formalization, we divide these behaviors into unit actions in unit time steps (e.g., a punch behavior is composed of 13 full body actions of unit time steps in Figure 1). Thus, the action space is the physically-realistic body movements of a humanoid character.

By definition, Feint behaviors aim to provide deceptive attacks, thus they are naturally expected to be derived from a subset of existing offensive behaviors. Based on our exploration, we derive two key findings from an extensive amount of offensive behaviors. First, most offensive behaviors can be decomposed into three action sequences, which are Stretch-out Sequence (Sequence 1), Reward Sequence (Sequence 2), and Retract Sequence (Sequence 3) (an example shown in the first row in Figure 1). We elaborate on each action sequence in detail. Sequence 1 delineates all the actions, by leading the agent movements to the Reward Sequence (in boxing, approaching the opponents before actually punching them); Sequence 2 contains actions that gain game rewards (in boxing, physical contact with the opponents); and Sequence 3 retracts an agent's movements to a relative rest position (in boxing, retracting back to a preparation position for next behaviors). Second, body movements in Sequence 1 and Sequence 3 usually have semi-symmetric yet reverse-order action patterns in the timeline. A behavior usually starts and ends in a similar physical state due to physical restrictions (e.g., bones and muscles stretching restrictions for a humanoid).

Under the above three-stage decomposition of offensive behaviors, there are abundant possibilities to composing Feint behaviors from the three action sequences. However, to ensure physically realistic generation, we summarize two requirements that Feint behaviors must follow: (1) Feint behaviors should follow semi-symmetrical patterns to effectively deceive opponents and return to a rest position for follow-up moves. In boxing, a human player must retract the stretched-out limbs to the relatively rest position, before stretching out to perform an actual attack action. This is because the retraction requires recharging the force to contracted muscles; and (2) transitions between adjacent actions in different behaviors are expected to be smooth, as humanoid body movements must provide continuous movements.

To satisfy the above two requirements, we propose a Feint behavior template generator called **Palindrome-directed Generation of Feint Templates**, by extracting subsets of semi-symmetrical actions from an offensive behavior and synthesizing them as a Feint behavior. The general method to generate these templates are (1) by extracting subsets of unit actions from an attack behavior, a Feint behavior can be considered as a semi-finished real attack behavior. This ensures the high similarity of a generated Feint behavior with an attack behavior, thus opponents could be

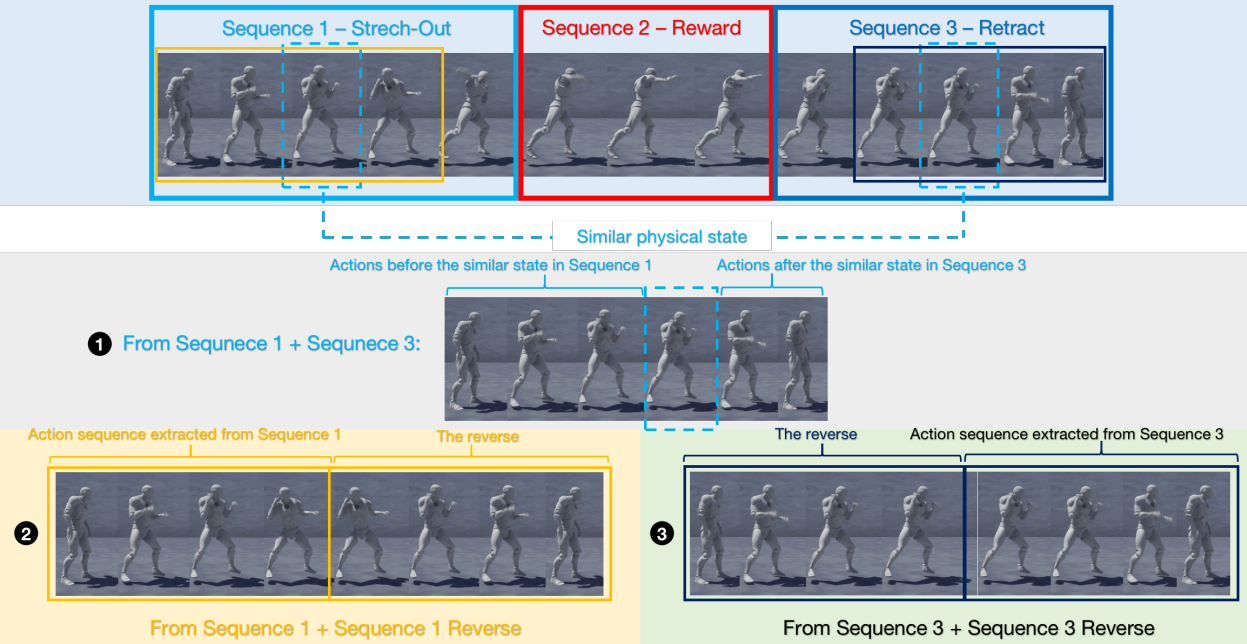


Figure 1: An example of **Palindrome-directed Generation Templates of Feint behaviors**. The first row shows an action sequence of a cross-punch behavior. 3 examples of templates are shown as ❶, ❷, and ❸ to demonstrate physically realistic generation of Feint behaviors.

deceived; and (2) by synthesizing semi-symmetric action sections, the overall movements can be connected smoothly and the naturalness of humanoid actions can be guaranteed. Within our proposed template generator **Palindrome-directed Generation of Feint Templates**, there are two key adjustable parameters in practice: (1) sequence composition positions for Feint templates; and (2) sequence length for Feint templates. We provide the rationales for these two key design choices.

(1) **Sequence composition positions for Feint templates:** Determining which position to extract the subsets of action sequences needs to ensure that the extracted actions are semi-symmetrical and allow physically realistic connections. To this end, we could have three templates with different restrictions to exploit the composing patterns: (A) For template ❶, if there are similar physical states, which refer to the positions of all joints and stretching angles are similar (as shown in ❶ of Figure 1), actions before the first similar state and after the second similar state can be extracted and directly synthesized as a Feint behavior (shown in ❶ of Figure 1); (B) For template ❷, by cutting once at any time point in Sequence 1, action sequences before the selected point and the corresponding reversion can be synthesized as a Feint behavior (shown in ❷ of Figure 1); and (C) For template ❸, similar to the second situation, by cutting once at any time point in sequence 3, action sequences after the selected point and the corresponding reversion can be synthesized as a Feint behavior (as shown in ❸ of Figure 1). With these considerations, the Feint behavior generation templates guarantee the naturalness of continuous movements via semi-symmetrical patterns.

(2) **Sequence length for Feint templates:** The choices for the length of extracted action sequences in each template can vary greatly, since multiple actions in an offensive behavior can be extracted based on different time ranges. The available choices could be any time length that results in action sequences that satisfy the physical requirements discussed above (e.g. morphologically reasonable Template ❷ or Template ❸ in Figure 1). Note that it's also possible to construct nested Feint behaviors, given a large number of feasible extraction positions. We formalize this choice as

a learnable parameter that needs to combine Feint behaviors with their intended follow-up actions (Section 3.2), and the learning adjustment is described in Section 5.

3.2 Feint behavior in consecutive game steps

Standalone Feint behaviors are meaningless in competitive games since the Feint behaviors themselves do not gain rewards for agents. Only by effectively combining Feint behaviors with intended follow-up actions could showcase their effectiveness. Thus, we define an effective Feint cycle as a **Dual-Behavior Model**, which jointly considers a Feint behavior and its intended follow-up behavior (could be a single action or an action sequence). Our formalization for standalone Feint behaviors (Section 3.1) already provides a large number of possible Feint behaviors. However, not all these morphologically reasonable Feint actions can be directly combined with all high-reward follow-up actions in combating scenarios. Therefore, certain constraints are demanded to construct effective combinations of Feint behaviors and follow-up actions. Hereby, we introduce two major considerations and then propose relevant restrictions, to enable naturalistic and suitable combinations of Feint behaviors and follow-up actions.

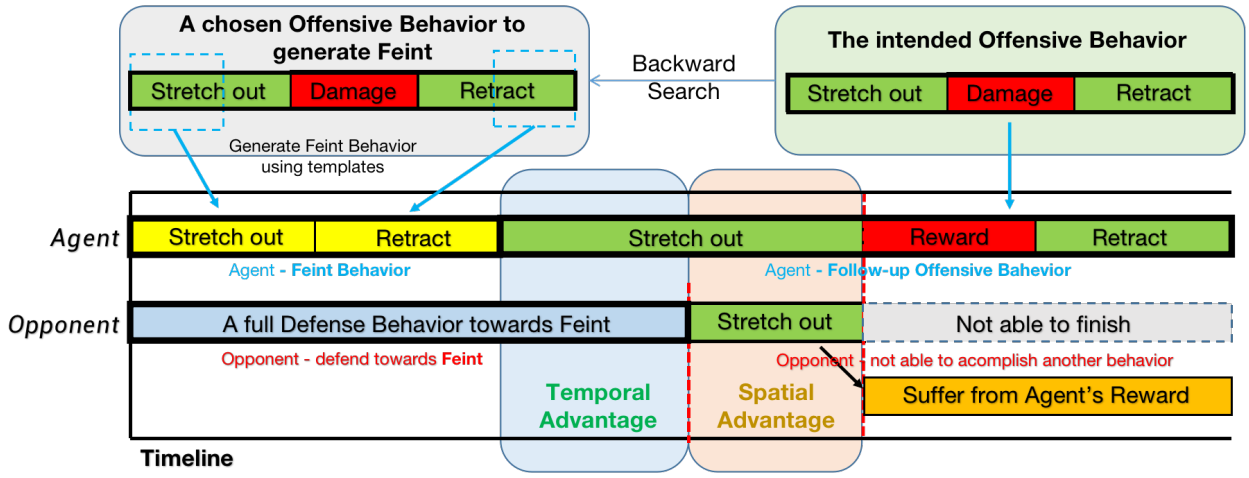


Figure 2: Dual-action Model - high-level abstraction and demonstration of internal stage transitions

(1) **Physical Constraints:** Physical constraints need to be accounted for when synthesizing Feint behaviors and follow-up actions. The ending physical state for a Feint behavior must be a state that is physically possible for an agent to perform the follow-up high-reward actions. For example, if a virtual character finishes Feint actions with the left foot forward, but the following attack action starts with the right foot forwarded, the synthesis of these two actions is inappropriate since this combination is physically unrealistic.

To ensure that the combinations of Feint behaviors and follow-up actions obey the physical constraints, we use a Reverse Search Principle which decides the intended follow-up actions (behavior) first and then use the starting physical state of this behavior to search and compose proper Feint behaviors (a more detailed description combined with strategy is described in Section 5). By first selecting an intended follow-up high-reward behavior, the end physical state of the Feint behavior is constrained to be close to the starting physical state of the follow-up behavior. Thus the composition of possible Feint behaviors using the Palindrome-directed templates should aim to start and end at a physical state that is close to the follow-up behavior. Figure 8 demonstrates a physically-realistic combination of a Feint action and an attack action.

(2) **Effectiveness:** The effectiveness of the incorporation of Feint behaviors is evaluated by whether the following attack actions can successfully hit the opponent. A successful Feint behavior would usually enable an agent to gain temporal and spatial advantages when performing the follow-up behaviors. Thus, the two design parameters introduced in Section 3 play crucial roles in combining Feint with follow-up behaviors. The abstraction of an ideal Dual-Behavior model that could enable an agent with temporal and spatial advantage is illustrated in Figure 2 and a corresponding example is provided in Figure 8. An effective Feint behavior creates temporal advantages that make the oppenoents to defend in a wrong direction and enable temporal advantages to allow the follow-up high-reward behavior to successfully gain rewards on the opponents. We also provide a detailed demonstration for successful and unsuccessful Feint cases in Figure **??** in Appendix A. This is a strategy-level decision problem and all these parameters could be naturally formalized in a Reinforcement Learning scheme, thus we leave the detailed discussion in Section 4 and implementation details in Section 5.

4 Formalizing Feint behaviors in strategy

To effectively fuse the Feint beahviors using Dual-Behavior Model into game interaction, we provide the strategy-level formalization of Feint behaviors. We use Multi-Agent Reinforcement Learning (MARL) schemes to discuss our formalization of Feint behaviors in the strategy level, as MARL provides flexibility in exposing multiple adjustable parameters in learnable policy models. As discussed in generating Feint behaviors (Section 3.1) and composing them in the Dual-Behavior Models (Section 3.2), the key considerations for effective Feint cycle is to enable temporal and spatial advantages for an agent. Thus, our strategy-level formalization centers on how to address the temporal, spatial, and their collective impacts of Feint behaviors with a Dual-Behavior Models. A more concrete introduction for fusing of Feint into the MARL frameworks is presented in Section 5.

4.1 The Basic Formalization: Derivation and Limitations

We first summarize two major limitations of existing works to justify that they cannot deliver a sufficient formalization of Feint in Multi-Player Games. Since there are no prior formalization, we discuss relevant works and derive the key features to discuss them in detail.

❶ The basic formalization on temporal impacts is insufficient for Multi-Player Games. Multi-Player Games require agents to account for future planning for decision-making, which is critical for deceptive actions like Feint [22, 24, 26]. Several works simplify the temporal impacts of deceptive game strategies in different gaming scenarios. [22] uses a discount factor γ to calculate the reward for following actions as $\sum_{t=0}^{\infty} \gamma^t R^i(s_t, a_t^i, a_t^{-i})$ for agent i. However, such a method suffers from the "short-sight" issue [24], since the weights for future actions' rewards shrink exponentially with time, which are not suitable for all gaming situations (discussed in [26]). More recently, [14] applies a long-term average reward, to equalize the rewards of all future actions as $\frac{1}{T}\sum_{t=0}^{T} R^i(s_t, a_t^i, a_t^{-i})$ (i.e. for agent i). However, such a method is restricted by the "far-sight" issue, since there are no differentiation between near-future and far-future planning. The mismatch between abstraction granularity heavily saddles with the design of Feint , because they use relatively static representations (e.g. static γ and T). Therefore, they cannot be aware of any potential changes of strategies in different phases of a game. Hence, the temporal dimension is simplified for the basic Feint formalization.

❷ The basic formalizations of spatial impacts are generally in simplified 2-player scenarios only, which cannot be effectively generalized to Multi-Player Game scenarios. Prior works, which attempt to fuse Feint into complete game scenarios, only consider two-player scenarios [36, 31]. However, in Multi-Player (more then two player) Games, gaming strategies (especially deceptive strategies) yield spatial impacts on other agents. Such impacts have been overlooked by all prior works. This is because an agent, who launches the Feint actions, can impact not only the target agent but also other agents in the scenario. Therefore, the influences of such an action needs to account for spatial impacts [19]. Moreover, with a new dimension accounted, the interactions between these two dimensions also raise a potential issue for their mutual collective impacts.

4.2 Our formalization in a generalized game model

Therefore, to deliver an effective formalization of Feint in Multi-Player Games, it's essential to consider the temporal, spatial and their collective impacts comprehensively. We first discuss the Temporal Dimension, then we elaborate our considerations on Spatial Dimension, and finally we summarize the design for the collective impacts from both temporal and spatial dimensions.

Under commonly used MARL schemes, we define a K-agent *Non-transitive Active Markov Game Model* as a tuple $\langle K, S, A, P, R, \Theta, U \rangle$: $K = \{1, ..., k\}$ is the set of k agents; S is the state space;

$A = \{A_i\}_{i=1}^K$ is the set of action space for each agent, where there are no dominant actions; P performs state transitions of current state by agents' actions: $P : S \times A_1 \times A_2 \times ... \times A_K \to P(S)$, where $P(S)$ denotes the set of probability distribution over state space S; $R = \{R_i\}_{i=1}^K$ is the set of reward functions for each agent; $\Theta = \{\Theta_i\}_{i=1}^K$ is the set of policy parameters for each agent; and $U = \{U_i\}_{i=1}^K$ is the set of policy update functions for each agent.

4.2.1 Temporal dimension: Influence time

To formalize the temporal impacts of Feint behaviors based on our Palindrome-directed Templates and the Dual-Behavior Model, we use a *Dynamic Short-Long-Term* manner to emulate them, which differ from the prior works' formalization (Section 4.1). The short-term period refers to a complete Dual-Behavior Model (Section 3.2), including a Feint behavior followed by an intended high-reward behavior led by the Feint . The long-term periods are the time steps after this Feint cycle. The rationale behind such a design choice is that: the purpose of Feint is to obtain strategic advantages against the opponent in the temporal dimension, aiming to benefit the follow-up high-reward behavior. Hence, the *Dynamic Short-Long-Term* temporal impacts of Feint shall be (1) the actions that follow Feint actions (e.g. actual attacks) in a short-term period of time should have a strong correlation to Feint ; (2) the actions in the long-term periods explicitly or implicitly depend on the effect of the Feint and its following actions; and (3) for different Dual-behavior models in different gaming scenarios, the threshold that divides short-term and long-term should be dynamically adjusted to enable sufficient flexibility in strategy making.

For *Dynamic Short-Long-Term*, we use the time-step length of a Dual-Behavior Model st as the short-term planning threshold. For the short-term (the Dual-Behavior), which starts at time step t_0 with actions of a Feint behavior $\{a^i_{t_0}, ..., a^i_{t_0+sf}\}$ and actions of a high-reward behavior $\{a^i_{t_0+sf+1}, ..., a^i_{t_0+st}\}$ (sf denotes the Feint behavior length), we use a set of large weights $\alpha = \{\alpha_{t_0}, ..., \alpha_{t_0+st}\}$ are used to calculate the reward:

$$Rew_{short-term}(\pi_i^{'}, t_0, st, \alpha) = \alpha_t \sum_{t=t_0}^{t=t_0+st} R^i(s_t, a^i_t, a^{-i}_t) \tag{1}$$

since the purpose of Feint policy π_i' is to actively find effective combinations of Feint behaviors and high-reward behaviors in Dual-Behavior Models that could benefit in a short-term period. We then consider long-term planning after the short-term planning threshold st: we use a set of discount factor $\beta = \{\beta_{t_0+st+1}, ..., \beta_T\}$ on the long-term average reward calculation (proposed by [14]), to distinguish these reward from short-term rewards:

$$Rew_{long-term}(\pi_i^{'}, t_0, st, T, \beta) = \beta_t \frac{1}{T} \sum_{t=t_0+st+1}^{T} R^i(s_t, a^i_t, a^{-i}_t) \tag{2}$$

where T denotes the end time of the game.

Finally, we put them together to formalize the *Short-Long-Term* reward calculation mechanism, when an agent i plans to perform a Feint action at time t_0 with a short-term planning threshold st and the end time of game T as:

$$Rew_{temporal}(\pi_i^{'}, t_0, st, T, \alpha, \beta) = \lambda_{short} Rew_{short-term}(t_0, st, \alpha) + \lambda_{long} Rew_{long-term}(t_0, st, T, \beta) \tag{3}$$

where λ_{short} and λ_{long} are weights for dynamically balancing the weight of short-term and long-term rewards for different gaming scenarios. λ_{short} and λ_{long} are initially set as 0.67 and 0.33 and are adjusted to achieve better performance with the iterations of training.

4.2.2 Spatial dimension: Influence range

The spatial advantage of Feint behaviors refers to deceive the opponents (i.e., change the opponents' actions from their original plans). In a Multi-Player Game (i.e. usually more than two players), the strict one-to-one relationship between two agents is not realistic, since an agent can impact both the target agent and other agents. Therefore, the influences on all other agents shall maintain different levels [19]. Therefore, our work includes the spatial dimension of Feint impacts by fusing spatial distributions. The key idea of this design is to combine spatial distribution with the influence range during the game. More specifically, we incorporate Behavioral Diversity from [19], to mathematically calculate and maximize the diversity gain of Feint actions in terms of the influence range.

We formalize the influence range of an action policy on K agent based on $S \times A_i \times \ldots \times A_K$, which follows a distribution of multi-to-one relationships $T \rightarrow (\alpha_1 T_{(i,1)}, \alpha_2 T_{(i,2)}, \ldots, \alpha_K T_{(i,K)})$. The influence distribution can have different factors in different gaming scenarios. The spatial domain influence could be naturally represented by the observation space of agents. We demonstrate a set of commonly used observation parameters in boxing games [35] where agent i plays against opponent $-i$: chosen action k of agent i A_i^k and opponent A_{-i}^j, the relative positions $p(i,-i)$, relative moving orientations $o(i,-i)$, the linear velocities $l\_vel(i,-i)$, and angular velocities $a\_vel(i,-i)$. These observations could be composed in a vector $V = (A_i^k, A_{-i}^j, p(i,-1), o(i,-i), l\_vel(i,-i), a\_vel(i,-i))$. When a Feint policy π_i' is added, we aim to maximize the effective influence range under the influence distribution of Feint . Assuming an agent i maintains a policy pool $P_i = \{\pi_i^1, \pi_i^M\}$, such influence distribution can be fused into Behavior Diversity measurement of the effective influence range by maximizing the discrepancy between the old influence effectiveness of policy occupancy measure $\rho_{\pi_E}(T)$ and the influence effectiveness when adding Feint policy of new policy occupancy $\rho_{\pi_i', \pi_{E_{-i}}}(V')$:

$$max_{\pi_i'} Rew_{spatial}(\pi_i^{'}, V') = D_f(\rho_{\pi_i', \pi_{E_{-i}}}(V') \mid\mid \rho_{\pi_E}(V)) \tag{4}$$

where the general f-divergence is used to measure the discrepancy of two distributions.

4.3 Collective impacts: Influence degree

Solely relying on Temporal Dimension and Spatial Dimension overlooks the interactions between them, and these two dimensions are expected to have mutual influences for a realistic modeling [19]. Therefore, we consider the influence degree, so the collective impacts of these two dimensions can be aggregated in a proper manner.

We formulate the collective impacts for a Feint policy π_i' in a Multi-Player Game that starts at t_0 and end at T as:

$$Rew_{collective}(\pi_i^{'}) = \mu_1 \sum_{i=1}^{k} Rew_{temporal}(i, \pi_i^{'}, t_0, st, T, \alpha, \beta) + \mu_2 \sum_{t=t_0}^{st} max_{\pi_i^{'}} Rew_{spatial}(\pi_i^{'}, V', t) \tag{5}$$

where temporal impacts $Rew_{temporal}$ (Section 4.2.1) are aggregated on spatial domain and spatial impacts $Rew_{spatial}$ (Section 4.2.2) are aggregated on temporal domain. μ_1 and μ_2 denote the weights of aggregated temporal impacts and spatial impacts respectively, enabling flexible adaption to different gaming scenarios. They are initially set as 0.5 and are adjusted to achieve better performance with the iterations of training.

In addition to the collective impacts of Feint itself in terms of temporal domain and spatial domain, our formalized impacts of Feint can also result in response diversity of opponents, since different related opponents (spatial domain) at different time steps (temporal domain) can have diverse response. Such diversity can be used as a reward factor that make the final reward calculation

more comprehensive [25, 19]. Thus, to incorporate such diversity together with our final reward calculation model, we refer to [19] to characterize the diversity gain incurred by our collective impacts formalization. When the impact $Rew_{collective}$ of Feint policy π^{M+1} in a $M \times N$ payoff matrix $A_{P_i \times P_i}$ at when opponents choose policy π^j_{-i} is collectively calculated, the derived diversity gain can be measured as follows:

$$Rew_{collective-diversity}(\pi_i^{M+1}) = D(a_{M+1} \mid\mid A_{P_i \times P_i}) \tag{6}$$

$$a^T_{M+1} := (Rew_{collective}(\pi_i^{M+1}, \pi^j_{-i}))^N_{j=1}. \tag{7}$$

where $D(a_{M+1} \mid\mid A_{P_i \times P_i})$ represents the diversity gain of the Feint action on current policy space. We follow the method in [19] for the quantification of diversity gain.

5 Implementation

To provide a unified implementation scheme of Feint into most MARL frameworks, we choose to implement on the training iteration level and avoid changing the MARL models themselves. We create an additional policy model (e.g., MADDPG [20], MASAC [11, 13], MAD3PG [4, 6], MATD3 [2], etc.) for each agent as the Feint policy, which works together with the regular policy models for agents but is trained and inferenced differently.

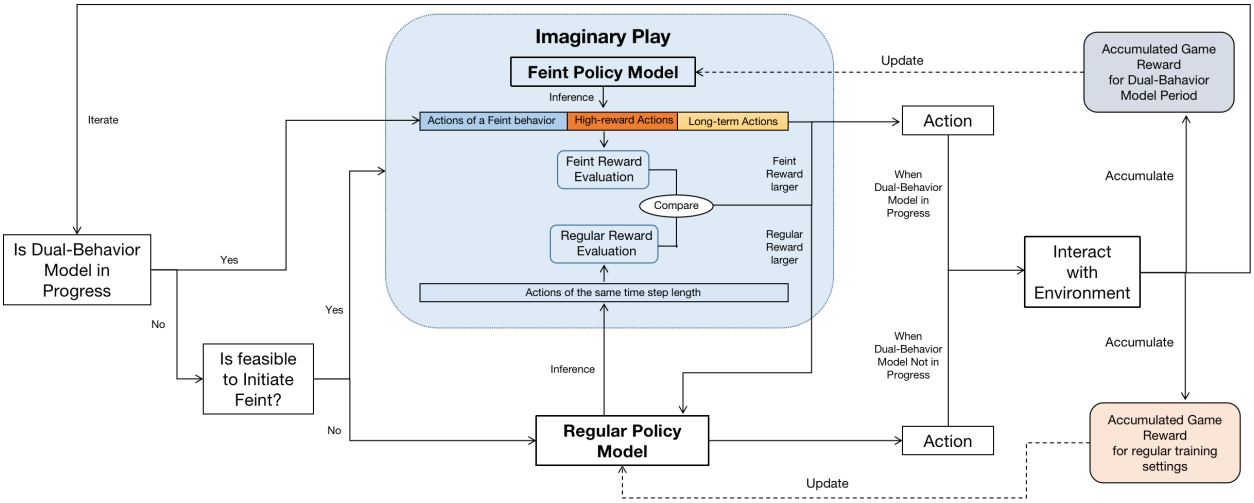


Figure 3: Illustration of Feint behavior implementation in a game step

We implement the Feint behavior generation in an imaginary play module in training iterations (i.e., each game step). The imaginary play module decides whether an agent should initiate a Feint behavior, composes a Dual-behavior Model using Palindrom-directed templates, and utilizes the Feint reward calculation to evaluate the quality of the generated action sequence in the Dual-behavior model. The imaginary play will only be activated when no Dual-Action Model is in progress and the current physical state s_c of an agent is close to a physical state s_r where it is physically realistic for the agent to perform a high-reward action a_i, while the possibility of performing a_i is relatively low according to its regular policy model (i.e., action a_i are highly likely to be diminished by other agents current actions). Thus, the purpose of Feint behavior is to lead the agent to a state s_r where the agent could maximize the game environment reward by performing the intended high-reward action a_i (i.e., other agents are deceived by Feint to perform other actions which cannot effectively diminish the high-reward actions performed by the agent).

When the imaginary play is activated, a series of actions that compose the Feint behavior is generated using the Palindrome-directed templates (Section 3.1), iteratively sampling actions from the agent's Feint policy model. Note that when the agent's Feint policy model would only select actions that are composed in offensive behaviors (set other actions possibilities to 0) in corresponding templates and use a reflection frame to compose a (semi-)palindrome which leads to the agent's physical state s_r. After composing the Feint behavior, a Dual-Behavior model is naturally created by performing the Feint behavior and followed by the some high-reward actions. The short-term reward can thus be calculated. After this Dual-Behavior action sequence, the imaginary play would play a few steps to incorporate the long-term reward. The collective reward (Section 4.2.2) can thus be calculated. This collective reward is then compared to an accumulated reward from an imaginary play using only the agent's regular policy model in the same number of time steps. If the Feint collective reward is higher, the action sequence of the dual action model will be applied

in the following real game steps. When a Dual-Action Model is in progress, the actions will not be sampled from the regular policy models.

In the real game steps, where all the agents' actions interact with the environment and the real game rewards are calculated, our formalization of Feint only changes the way to update the Feint policy models for agents. The Feint policy models are updated only when corresponding Dual-Behavior Models finish and are updated using the accumulated real game rewards for that period. The regular policy models are updated as usual settings (e.g., after some fixed steps - an episode).

6 Experiments and Results

6.1 Experimental Methodology

Testbed Implementations. Our main testbed game environment is a multi-player boxing game, which is based on an open-source environment Multi-Agent Particle Environment [23] but with heavy additional implementation to create a physically realistic scenario.This game resembles intense free fight scenarios in ancient Roman free fight scenarios [21], where interactions are intense and Feint is expected to be effective. We incorporate common boxing behaviors (action sequences) in boxing games. following the methodology in some animation and simulation works [34, 36]. This handcrafted scenario contains complex physics-based interaction systems and fine-grained time steps to enable learning and generating Feint behaviors. A detailed description of the reward gaining system, environment parameters, and agent settings is presented in Appendix B. We also re-implement and extend a strategic real-world game, AlphaStar [3], which is widely used as the experimental testbed in recent studies of Reinforcement Learning studies [28, 19]. We make extra efforts to emulate a six-player game, where players are free to have convoluted interactions with each other. And we implement Feint as dynamically generated policies, based on the 888 regular gaming policies.

Experiment Procedure. We choose 4 commonly used MARL models: MADDPG [20], MASAC [11, 13], MATD3 [2], and MAD3PG [4, 6] and incorporate them into testbed scenarios. Our implementation is based on [1], which provides a unified MARL frameworks for the above models. We aim to test whether Feint behaviors could be uniformly and effectively learned using all these commonly used MARL models and how could Feint affect the game rewards for agents. Note that our purpose is to verify the effectiveness of our formalization of Feint behaviors and not to compare or modify the MARL models themselves. We create two test scenarios, the first one with two players (one player per team) and the second one with six players (3 players per team). For all of these scenarios, we first train the agents without Feint as baselines using the 4 models. Then for the two-player scenario, we incorporate Feint on one player (shown as the Good player in Figure 4). For the six-player scenario, we select 1 agent in the Good team (labeled as Good 3 in Figure 5), to incorporate our formalization of Feint , and keep all other 3 agents regular. The reason for this design in the six-player scenario is that we want to not only test how Feint behaviors can affect the reward gain against direct opponents, but also see whether Feint could bring advantages for a player among its teammates. All the players are rewarded independently and the notion of the "Good" and "Adv" team does not mean that teammates have a shared reward (i.e., not explicit constraints that force them to cooperate). Note that all players have identical capabilities and are rewarded using the same mechanisms, thus Feint could be incorporated on any player. Our labeling choice here is to provide to a consistent way to track and analysis the game rewards. All experiments for the two-player scenario are trained for 75000 game iterations and all experiments for the two-player scenario are trained for 150000 game iterations.

Evaluation Metrics. We examine the effects of Feint using ❶ gaming rewards of training, ❷ diversity gain of policy space and ❸ overhead of computation load. We first examine the gaming outcomes when using the MADDPG, MASAC, MATD3, and MAD3PG MARL models, by comparing the per episode gaming rewards of agents across all scenarios. Note that these rewards are the actual game rewards (the reward that returned by the gaming environment), which are not the rewards that policy models used to select actions or update parameters. We then examine the effects of Feint actions on how Feint can improve the diversity of gaming policies (Section 4.3). Finally, we perform overhead analysis, incurred by fusing Feint formalization in strategy learning.

6.2 Experimental Results

6.2.1 Gaming Reward Gain

Figure 4 shows the game reward comparisons of using Feint behaviors or not in the Two-Player scenario (Section 6.1) for 4 MARL models. The first row shows the baseline results where all agents are trained normally, while the second row shows the results where the player labeled with "Good" incorporates Feint behaviors. In most of the baseline results (e.g., using MADDPG, MAD3PG, and MATD3), the two players' rewards tend to progress to a similar level when after enough training iterations. For MASAC, the "Good" player seems to gain higher rewards than its opponents when the training iterations are large, but the advantage is not stable and such a phenomenon could likely be the instability of the MASAC algorithm itself . For all the results where Feint behaviors are incorporated, we could see a significant advantage gain for the "Good" player. Thus, our formalization of incorporating Feint behaviors could effectively improve the actual game rewards in two-player combating scenarios.

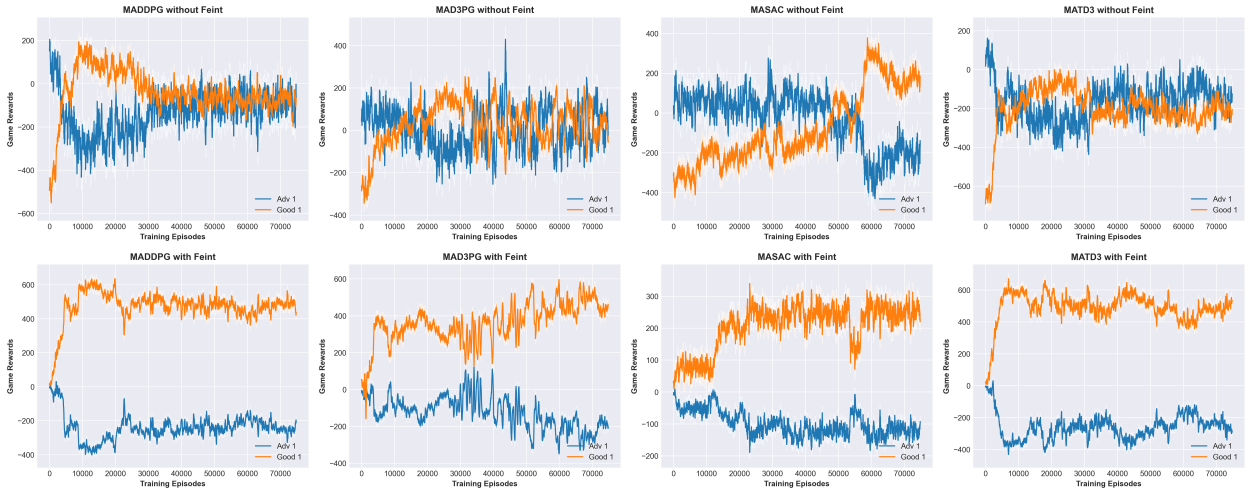


Figure 4: Comparison of Game Reward when using Feint and not using Feint in a 1 VS 1 scenario.

To further evaluate the effectiveness of our formalization of Feint behaviors in multi-player scenarios, Figure 5 shows the game reward comparisons in Six-Player scenario (Section 6.1) for 4 MARL models. The first row shows the baseline results while the second row shows the results where the player labeled with "Good 3" incorporates Feint behaviors. In all baseline results, all 6 players seem to achieve similar levels of rewards after enough training iterations. In comparison, in all results where the "Good 3" player incorporates Feint , it gains significantly more rewards than the opponents as well as its teammates. This result shows that our formalization of Feint could not only gain higher rewards towards the direct opponents, but also gain advantages among teammates who do not incorporate Feint . Another interesting observation is that there are no more symmetric patterns in the players' rewards, showing that the gaming interactions in multi-player scenarios have enough complexity (Note that the scenario is not designed to be a zero-sum game).

6.2.2 Diversity Gain

To examine the impacts on the policy diversity in games, we perform a comparative study between MARL training with and without Feint . Specifically, We use Exploitability and Population Efficacy (PE) to measure the diversity gain in the policy space. Exploitability [15] measures the distance of

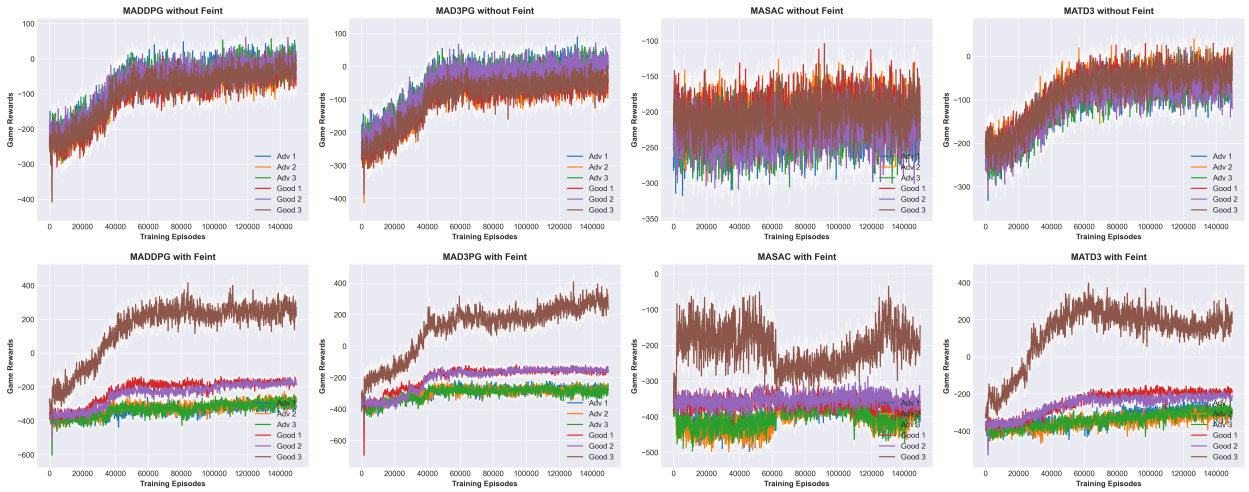


Figure 5: Comparison of Game Reward when using Feint and not using Feint in a 3 VS 3 scenario.

a joint policy chosen by the multiple agents to the Nash equilibrium, indicating the gains of players compared to their best response. The mathematical expression of Exploitability is expressed as:

$$Expl(\pi) = \sum_{i=1}^{N} (max_{\pi_i'} Rew_i(\pi_i', \pi_{-i}) - Rew_i(\pi_i', \pi_{-i})) \tag{8}$$

where π_i stands for the policy of agent i and π_{-i} stands for the joint policy of other agents. Rew_i denotes our formalized Reward Calculation Model (Section 4.3). Thus, small Exploitability values show that the joint policy is close to Nash Equilibrium, showing higher diversity. In addition, we also use Population Efficacy (PE) [19] to measure the diversity of the whole policy space. PE is a generalized opponent-free concept of Exploitability by looking for the optimal aggregation in the worst cases, which is expressed as:

$$PE(\{\pi_i^k\}_{k=1}^N) = min_{\pi_{-i}} max_{1^\top \alpha=1 \ a_i>=0} \sum_{k=1}^{N} \alpha_k Rew_i(\pi_i^k, \pi_{-i}) \tag{9}$$

where π_i stands for the policy of agent i and π_{-i} stands for the joint policy of other agents. α denotes an optimal aggregation where agents owning the population optimizes towards. Rew_i denotes our formalized Reward Calculation Model (Section 4.3) and opponents can search over the entire policy space. PE gives a more generalized measurement of diversity gain from the whole policy space.

Figure 6 shows the experimental results for evaluating diversity gains. From the figure, we obtain two observations. First, agents that can dynamically perform Feint actions (Agent 1, 2, and 3) achieve lower Exploitability (around 4.9×10^{-2}) compared to agents who perform regular actions (around 9.7×10^{-2}) and have higher PE (lower negative PE - around 5.3×10^{-2}) than those who only perform regular actions (around 1.2×10^{-2}). This result shows that our formalized Feint can effectively increase the diversity and effectiveness of policy space. Second, agents with Feint have slightly higher variations in both metrics. This is because Feint naturally incurs more randomness (e.g. succeed or not) in games, resulting in higher variations in metrics.

6.2.3 Overhead Analysis

Figure 7 shows the results of our overhead analysis. We make two observations. First, fusing Feint in MARL training do incur some overhead increment in terms of running time. This is because

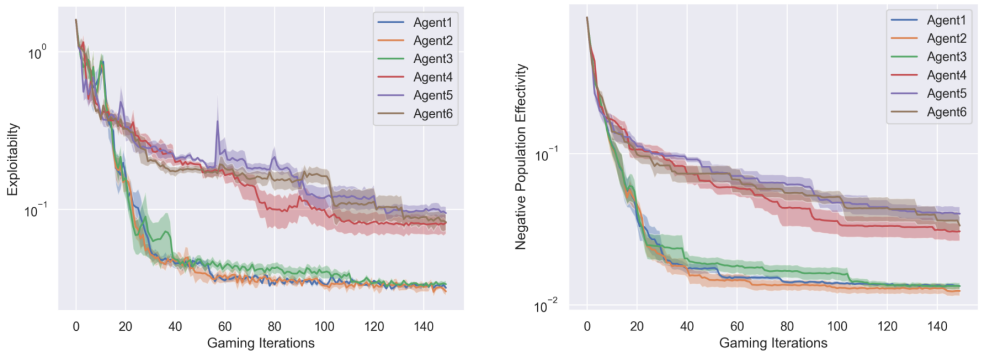


Figure 6: Diversity gain for agents, in terms of the exploitablity and the negative population efficacy.

the formalization and fusion of Feint in MARL incur additional calculation load. Secondly, in both MADDPG models and MAAC models, the increased overhead is generally lower than 5%, which still indicates that our proposed formalization of Feint actions can have enough feasibility and scalability on fusing with MARL models. Note that even we use two policy models for each agent in our implementation, our designs restrict that only one model is inferenced in each game step (Section 5), thus the overhead is low.

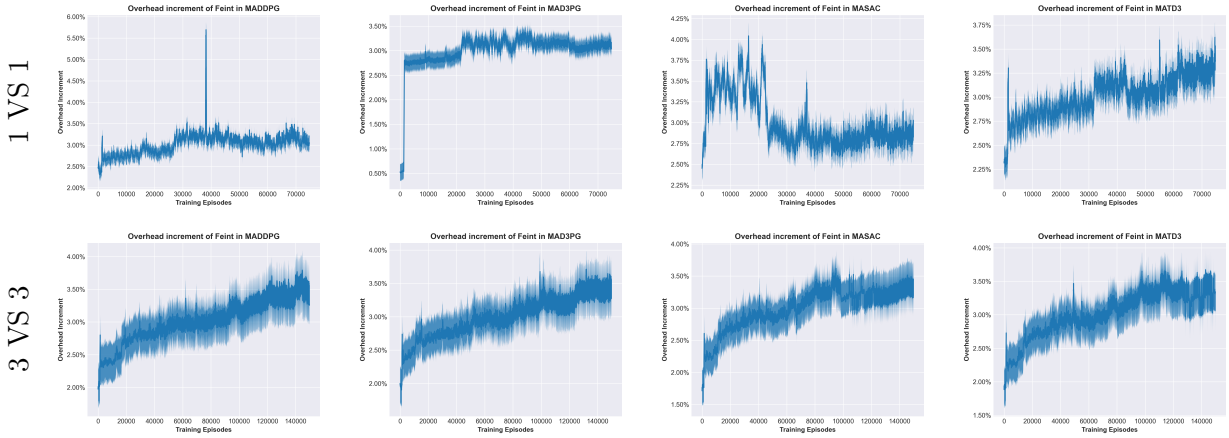


Figure 7: Overhead of Feint the 1 VS 1 and 3 VS 3 scenarios using 4 MARL models.

7 Conclusion

In this work, we introduce the first comprehensive formalization, implementation and quantitative evaluations of Feint in Multi-Player Games. We provide automatic generation of Feint behaviors using Palindrom-directed Templates and synergistically combine Feint with follow-up actions in Dual-Bahavior Model. The decision choices on the action-level are fused into strategy-level formalizations in game interactions. We provide a concrete implementation scheme to incorporate Feint into common MARL frameworks. The results show that our design of Feint can (1) greatly improve the reward gains from the game; (2) significantly improve the diversity of Multi-Player Games; and (3) only incur negligible overheads in terms of the time consumption. We conclude that our formalization of Feint is effective and practical, to make Multi-Player Games more interesting. Our formalization is also expected to be applicable for future models of Multi-Player Games due to its uniformed design.

References

[1] J. Ackermann, C. Badger, and T. Jiang, "Johannesack/tf2multiagentrl," November 2020. [Online]. Available: https://github.com/JohannesAck/tf2multiagentrl

[2] J. Ackermann, V. Gabler, T. Osa, and M. Sugiyama, "Reducing overestimation bias in multi-agent domains using double centralized critics," *arXiv preprint arXiv:1910.01465*, 2019.

[3] K. Arulkumaran, A. Cully, and J. Togelius, "Alphastar: an evolutionary computation perspective," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO 2019, Prague, Czech Republic, July 13-17, 2019*, M. López-Ibáñez, A. Auger, and T. Stützle, Eds. ACM, 2019, pp. 314–315. [Online]. Available: https://doi.org/10.1145/3319619.3321894

[4] G. Barth-Maron, M. W. Hoffman, D. Budden, W. Dabney, D. Horgan, D. TB, A. Muldal, N. Heess, and T. P. Lillicrap, "Distributed distributional deterministic policy gradients," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. [Online]. Available: https://openreview.net/forum?id=SyZipzbCb

[5] L. Critch and D. Churchill, "Sneak-attacks in starcraft using influence maps with heuristic search," in *2021 IEEE Conference on Games (CoG), Copenhagen, Denmark, August 17-20, 2021*. IEEE, 2021, pp. 1–8. [Online]. Available: https://doi.org/10.1109/CoG52621.2021.9619156

[6] D. Fan, H. Shen, and L. Dong, "Multi-agent distributed deep deterministic policy gradient for partially observable tracking," in *Actuators*, vol. 10, no. 10. MDPI, 2021, p. 268.

[7] J. N. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, S. A. McIlraith and K. Q. Weinberger, Eds. AAAI Press, 2018, pp. 2974–2982. [Online]. Available: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17193

[8] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, ser. Proceedings of Machine Learning Research, J. G. Dy and A. Krause, Eds., vol. 80. PMLR, 2018, pp. 1582–1591. [Online]. Available: http://proceedings.mlr.press/v80/fujimoto18a.html

[9] I. Güldenpenning, M. A. A. Alaboud, W. Kunde, and M. Weigelt, "The impact of global and local context information on the processing of deceptive actions in game sports," *German Journal of Exercise and Sport Research*, vol. 48, no. 3, pp. 366–375, 2018.

[10] I. Güldenpenning, W. Kunde, and M. Weigelt, "How to trick your opponent: A review article on deceptive actions in interactive sports," *Frontiers in psychology*, vol. 8, p. 917, 2017.

[11] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proceedings of the 35th*

International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, ser. Proceedings of Machine Learning Research, J. G. Dy and A. Krause, Eds., vol. 80. PMLR, 2018, pp. 1856–1865. [Online]. Available: http://proceedings.mlr.press/v80/haarnoja18b.html

[12] R. Hyman, "The psychology of deception," *Annual review of psychology*, vol. 40, no. 1, pp. 133–154, 1989.

[13] S. Iqbal and F. Sha, "Actor-attention-critic for multi-agent reinforcement learning," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 2019, pp. 2961–2970. [Online]. Available: http://proceedings.mlr.press/v97/iqbal19a.html

[14] D. Kim, M. Riemer, M. Liu, J. N. Foerster, M. Everett, C. Sun, G. Tesauro, and J. P. How, "Influencing long-term behavior in multiagent reinforcement learning," *CoRR*, vol. abs/2203.03535, 2022. [Online]. Available: https://doi.org/10.48550/arXiv.2203.03535

[15] M. Lanctot, V. F. Zambaldi, A. Gruslys, A. Lazaridou, K. Tuyls, J. Pérolat, D. Silver, and T. Graepel, "A unified game-theoretic approach to multiagent reinforcement learning," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017, pp. 4190–4203. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/3323fe11e9595c09af38fe67567a9394-Abstract.html

[16] J. Lee and K. H. Lee, "Precomputing avatar behavior from human motion data," *Graph. Model.*, vol. 68, no. 2, pp. 158–174, 2006. [Online]. Available: https://doi.org/10.1016/j.gmod.2005.03.004

[17] S. Lee, S. Lee, Y. Lee, and J. Lee, "Learning a family of motor skills from a single motion clip," *ACM Trans. Graph.*, vol. 40, no. 4, pp. 93:1–93:13, 2021. [Online]. Available: https://doi.org/10.1145/3450626.3459774

[18] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2016. [Online]. Available: http://arxiv.org/abs/1509.02971

[19] X. Liu, H. Jia, Y. Wen, Y. Yang, Y. Hu, Y. Chen, C. Fan, and Z. Hu, "Unifying behavioral and response diversity for open-ended learning in zero-sum games," *CoRR*, vol. abs/2106.04958, 2021. [Online]. Available: https://arxiv.org/abs/2106.04958

[20] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017, pp. 6379–6390. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/68a9750337a418a86fe06c1991a1d64c-Abstract.html

[21] D. Matz, *Ancient Roman Sports, AZ: Athletes, Venues, Events and Terms.* McFarland, 2019.

[22] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller, "Playing atari with deep reinforcement learning," *CoRR*, vol. abs/1312.5602, 2013. [Online]. Available: http://arxiv.org/abs/1312.5602

[23] I. Mordatch and P. Abbeel, "Emergence of grounded compositional language in multi-agent populations," *arXiv preprint arXiv:1703.04908*, 2017.

[24] A. Naik, R. Shariff, N. Yasui, and R. S. Sutton, "Discounted reinforcement learning is not an optimization problem," *CoRR*, vol. abs/1910.02140, 2019. [Online]. Available: http://arxiv.org/abs/1910.02140

[25] N. P. Nieves, Y. Yang, O. Slumbers, D. H. Mguni, Y. Wen, and J. Wang, "Modelling behavioural diversity for learning in open-ended games," in *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 2021, pp. 8514–8524. [Online]. Available: http://proceedings.mlr.press/v139/perez-nieves21a.html

[26] C. Nota and P. S. Thomas, "Is the policy gradient a gradient?" in *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems, AAMAS '20, Auckland, New Zealand, May 9-13, 2020*, A. E. F. Seghrouchni, G. Sukthankar, B. An, and N. Yorke-Smith, Eds. International Foundation for Autonomous Agents and Multiagent Systems, 2020, pp. 939–947. [Online]. Available: https://dl.acm.org/doi/abs/10.5555/3398761.3398871

[27] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa, "AMP: adversarial motion priors for stylized physics-based character control," *ACM Trans. Graph.*, vol. 40, no. 4, pp. 144:1–144:20, 2021. [Online]. Available: https://doi.org/10.1145/3450626.3459670

[28] S. Risi and M. Preuss, "Behind deepmind's alphastar ai that reached grandmaster level in starcraft ii," *KI-Künstliche Intelligenz*, vol. 34, no. 1, pp. 85–86, 2020.

[29] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *CoRR*, vol. abs/1707.06347, 2017. [Online]. Available: http://arxiv.org/abs/1707.06347

[30] H. P. H. Shum, T. Komura, and S. Yamazaki, "Simulating interactions of avatars in high dimensional state space," in *Proceedings of the 2008 Symposium on Interactive 3D Graphics, SI3D 2008, February 15-17, 2008, Redwood City, CA, USA*, E. Haines and M. McGuire, Eds. ACM, 2008, pp. 131–138. [Online]. Available: https://doi.org/10.1145/1342250.1342271

[31] O. So, K. Stachowicz, and E. A. Theodorou, "Multimodal maximum entropy dynamic games," *CoRR*, vol. abs/2201.12925, 2022. [Online]. Available: https://arxiv.org/abs/2201.12925

[32] Stefano Corazza and Nazim Kareemi, "Mixamo," 2022. [Online]. Available: https://www.mixamo.com/#/

[33] T. Team, "Toalha nerd-king of fighter xv: Trailer de ryo sakazaki e robert garcia!" 2021.

[34] K. Wampler, E. Andersen, E. Herbst, Y. Lee, and Z. Popovic, "Character animation in two-player adversarial games," *ACM Trans. Graph.*, vol. 29, no. 3, pp. 26:1–26:13, 2010. [Online]. Available: https://doi.org/10.1145/1805964.1805970

[35] J. Won, D. Gopinath, and J. K. Hodgins, "Control strategies for physically simulated characters performing two-player competitive sports," *ACM Trans. Graph.*, vol. 40, no. 4, pp. 146:1–146:11, 2021. [Online]. Available: https://doi.org/10.1145/3450626.3459761

[36] ——, "Control strategies for physically simulated characters performing two-player competitive sports," *ACM Trans. Graph.*, vol. 40, no. 4, pp. 146:1–146:11, 2021. [Online]. Available: https://doi.org/10.1145/3450626.3459761

[37] B. Yersin, J. Maïm, J. Pettré, and D. Thalmann, "Crowd patches: populating large-scale virtual environments for real-time applications," in *Proceedings of the 2009 Symposium on Interactive 3D Graphics, SI3D 2009, February 27 - March 1, 2009, Boston, Massachusetts, USA*, E. Haines, M. McGuire, D. G. Aliaga, M. M. Oliveira, and S. N. Spencer, Eds. ACM, 2009, pp. 207–214. [Online]. Available: https://doi.org/10.1145/1507149.1507184

A Demonstration of Feint Behaviors

A.1 Demonstration of Feint Behaviors in Dual-Beahvior Models

To explain the generation of physically realistic Feint behavior in a Dual-Behavior Model in detail, we use humanoid models: when selecting the corresponding actions (i.e. from Feint behaviors and then an attack behavior), the starting position (jointly connected body) of the second action should be the same as the ending position of the starting action. With such a principle, the joints of a character's body can perform natural movements during the transition between these two behaviors. Figure 8 demonstrates a physically realistic combination of a Feint behavior and a follow-up attack behavior. When checking the end of NPC A's Feint behavior and the beginning of the Agent's (left white agent) real attack, both the upper and lower body parts of NPC A perform the same postures (the left arm raised and the right arm charged, performing a punch for the upper body, and the left foot forward for lower body).

Figure 8 provides a detailed example of a successful Feint behavior in a Dual-Behavior Model. We refer to the Agent as the white player on the left and its Opponent as the black player on the right, and describe the Feint behavior from the Agent perspective. The agent first performs a Feint behavior which is fake punch towards its opponent's head, which leads the opponent to defend towards its head. However, the agent connects such Feint behavior with a follow-up hook towards the opponent's waist. Due to the temporal advantage gained by the quick Feint behavior and the spatial advantage gained by deceiving the opponents to defend to wrong directions, the opponent would be knocked down by the follow-up behavior of the agent. Thus, a successful Feint behavior is performed in this Dual-Behavior Model.

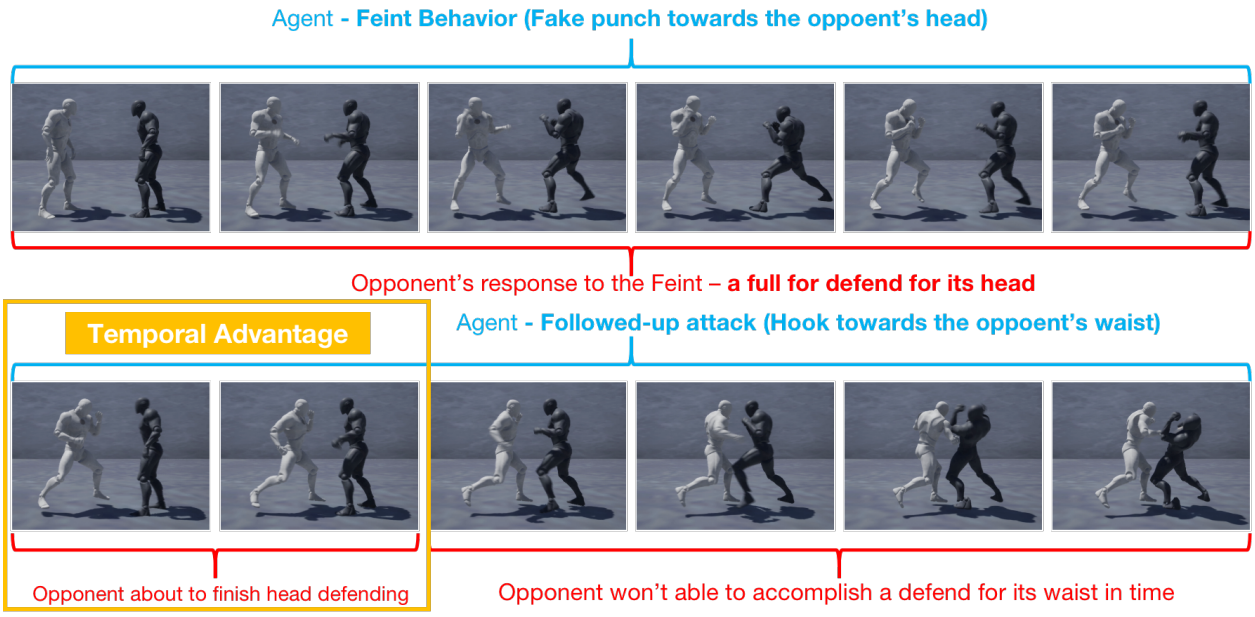


Figure 8: Dual-action Model - snapshots of the full process

A.2 Demonstration of Successful and Unsuccessful Feint Behaviors

To enable a successful Feint behavior in a Dual-Behavior Model, the temporal and spatial advantages should be properly formalized. The advantages of combining Feint behaviors with follow-up high-

reward actions stem from an appropriate time difference, incurred by Feint behaviors to mislead the opponents' actions. If the length of a Feint behavior is too short, the following attack actions might not gain much advantage compared to actions combinations without Feint behaviors; and if the length of a Feint action is too long, the process to perform a Feint behaviors can leave sufficient time for the opponent to react and even attack back. We provide examples for these scenarios in Figure **??**. We refer to the left white player as NPC A and describe the Feint from its perspective, and the right black agent NPC B is considered as its opponent.

We use the timeline of the Dual-Behavior Model in Figure 2 to analyze and evaluate the three Feint behaviors. We use three key time points that are highlighted in Figure 9, Figure 10, and Figure!11 to explain the action sequences, in which t_{B_1} indicates the end of defense behavior while t_{A_2} indicates the estimated start of reward in the second action sequence for NPC A and t_{B_2} indicates the estimated start of reward in second action for NPC B. The three consequences mainly differ in these three key time points.

1) **Very short Feint behaviors $t_{A_2} < t_{B_1}$:** The action sequence of simulation is shown in Figure 9, in which the Feint behaviorsduration is extremely short and the estimated start of reward in second action for NPC A (t_{A_2}) happens when NPC B is still in the first defense action (thus $t_{A_2} < t_{B_1}$). As the sequence shows, the second real action of NPC A would not benefit much since NPC B is still in defense.

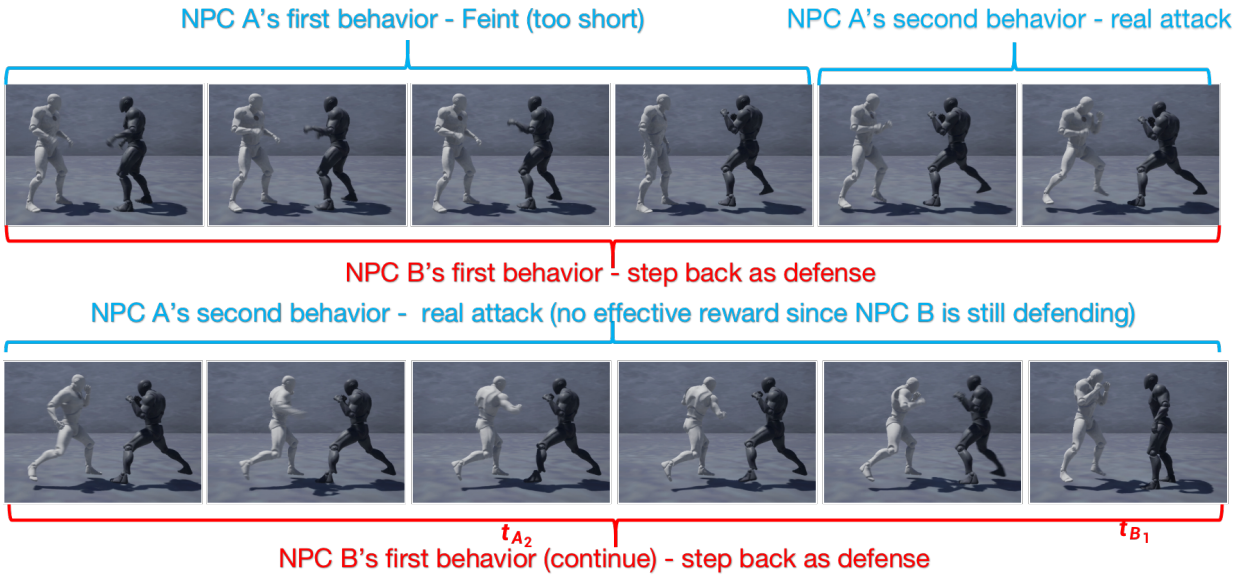


Figure 9: Demonstration of unsuccessful Feint behavior when its too short

2) **Proper length Feint behaviors $t_{B_1} < t_{A_2} < t_{B_2}$:** The action sequence of simulation is shown in Figure 10, in which the Feint behaviors have a moderate duration. The key difference of this duration is that the estimated start of reward in the second behavior for NPC A happens after the end of the defense behavior of NPC B and before the estimated start of reward in the second behavior for NPC B, thus showing the temporal advantages introduced in Section 3.2. With such temporal advantages, NPC A gains preemptive advantage over NPC B, inflicting rewards from NPC B (at time t_{A2} in Figure **??**) before NPC B's reward inflicting of second behavior starting (at time t_{B2} in Figure 3). When NPC A hits NPC B at t_{A2}, the ongoing action of NPC B will be interrupted and NPC B would be knocked down.

3) **Very long Feint behaviors $t_{A_2} > t_{B_2}$:** The action sequence of simulation is shown in

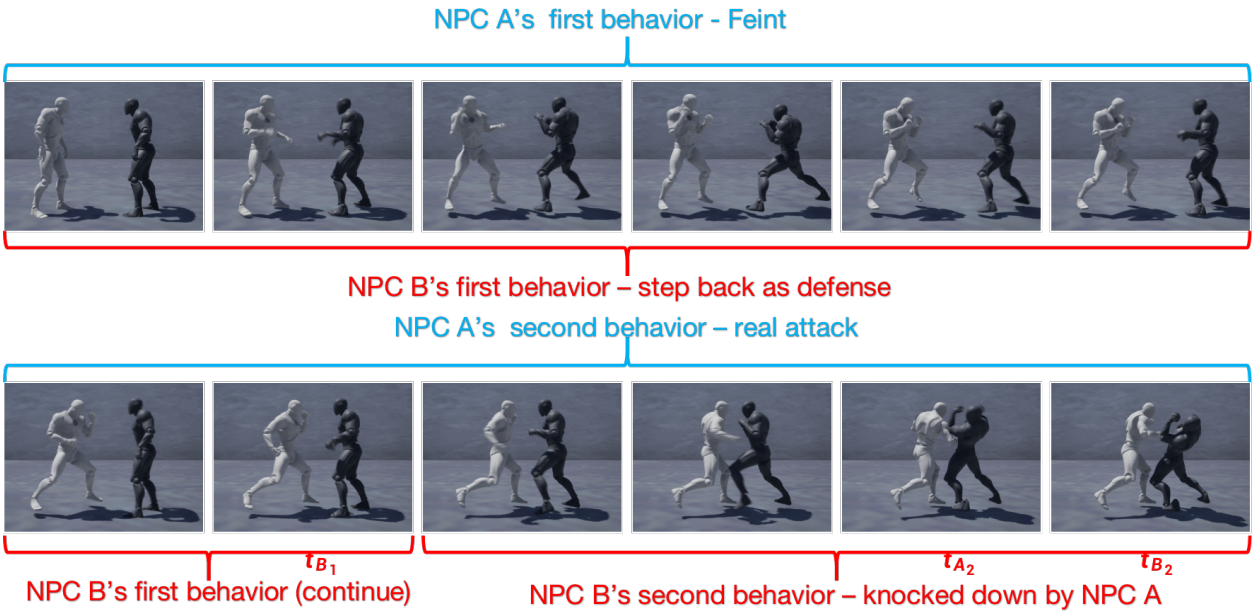


Figure 10: Demonstration of successful Feint behavior with proper length

Figure 11, in which the Feint actions duration is too long and the estimated start of reward in second behavior for NPC A (t_{A_2}) happens after the estimated start of damage in second action for NPC B (t_{B_2}). This condition has the opposite consequence of a moderate length Feint behaviors, in which NPC B can inflict rewards on NPC A before NPC A's reward inflicting of the second behavior starts. When NPC B hits NPC A at t_{A2}, the ongoing action of NPC A will be interrupted and NPC A would be knocked down.

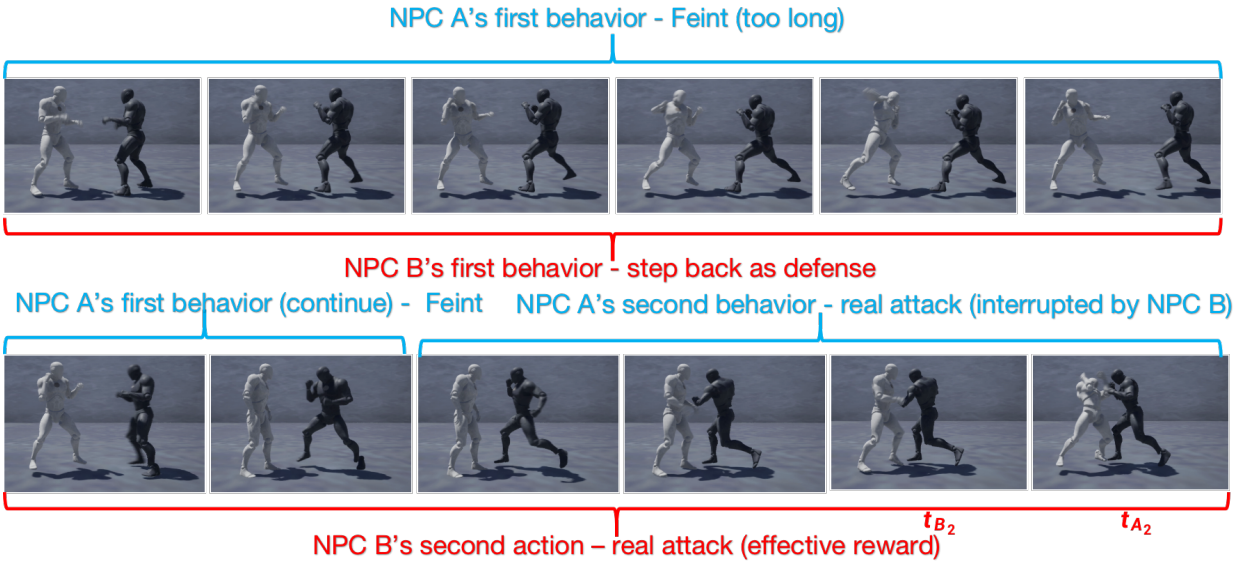


Figure 11: Demonstration of unsuccessful Feint behavior when its too long

Thus, the choice of the time duration for Feint actions highly depends on the action combinations and the estimation of opponents' actions, proving our observation in Section 3. Thus the learning

to formalize such a choice in the strategy learning scheme (Section 4) is important to construct effective Feint behaviors with corresponding Dual-Behavior Models.

B Details of Boxing Game Scenario

Our testbed game scenario is emulates a complex boxing game by modeling all the detailed combat behaviors except building the graphical rendering process. The reason we neglect the rendering process is that our main goal is to evaluate the effectiveness of formalization of Feint behaviors in multi-player games, and the building a real-time graphical rendering with such complex humanoid interactions would be a graphics paper itself. We fully emulate all the behavior details in our game simulation, thus our constructed game simulation is detailed enough to evaluate our formalization of Feint behaviors. We provide a detailed description of the game scenario here.

We follow a similar boxing game scenario construction approach as [34, 35], and model the full set of Mixamo [32] 22 behaviors (action sequences) which contain over 250 available full body actions (illustrated in Figure 12). We extensively construct a gaming environment based on Multi-Agent Particle System [23] to incorporate these behaviors, which then could be seamlessly integrated with common MARL models.

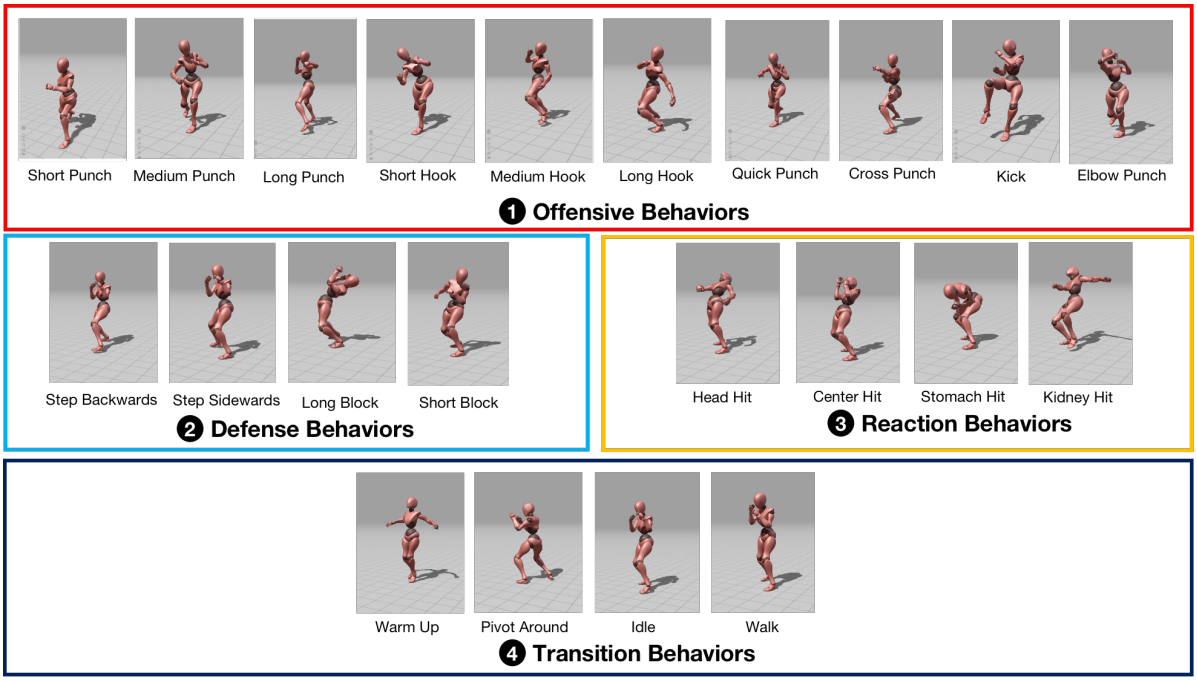


Figure 12: The full set of 22 behavior (action sequences) of a boxing game from Mixamo.

The players could move around in a 2D plane. We use a vector to model the physical state of players, which stores and tracks the body movements of a player. This vector tracks the positions of body parts: left and right limbs, the left and right legs, and the center body, which is used to select available combat behaviors (the transitions of body movements must be smooth as mentioned in Section 3.1 and Section 3.2). With this setting, Feint behaviors could be naturally generated and incorporated into suitable Dual-Behavior Models. We follow the exact Mixamo dataset to model the length of the behaviors (the length of action sequences) and rewards the behaviors (e.g., a successful long punch would gain more rewards than a short punch.) Specifically, we measure the number of frames contained in all behaviors and normalize them to define unit time steps for action space and thus get the action sequence lengths for all behaviors. An example of game rewards and action sequence length of 5 behaviors are provided in Figure 13.

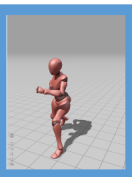
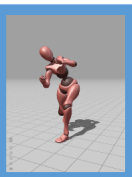
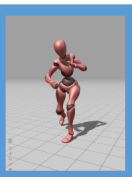
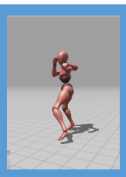
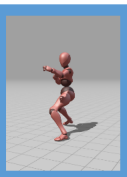
Example Behaviors	 Short Punch	 Short Hook	 Medium Punch	 Long Punch	 Cross Punch
Game Reward	7	8	12	20	23
Action Sequence Length	5	6	9	13	15

Figure 13: Demonstration of the game rewards and action sequence lengths of 5 Mixamo behaviors.