Implicit Behavioral Alignment of Language Agents in High-Stakes Crowd Simulations

Yunzhe Wang[1,2]   Gale M. Lucas[1,2]   Burcin Becerik-Gerber[1]   Volkan Ustun[2]

[1]University of Southern California
[2]USC Institute for Creative Technologies

{yunzhewa, becerik}@usc.edu   {lucas, ustun}@ict.usc.edu

Abstract

Language-driven generative agents have enabled large-scale social simulations with transformative uses, from interpersonal training to aiding global policy-making. However, recent studies indicate that generative agent behaviors often deviate from expert expectations and real-world data—a phenomenon we term the *Behavior-Realism Gap*. To address this, we introduce a theoretical framework called Persona-Environment Behavioral Alignment (PEBA), formulated as a distribution matching problem grounded in Lewin's behavior equation stating that behavior is a function of the person and their environment. Leveraging PEBA, we propose PersonaEvolve (PEvo), an LLM-based optimization algorithm that iteratively refines agent personas, implicitly aligning their collective behaviors with realistic expert benchmarks within a specified environmental context. We validate PEvo in an active shooter incident simulation we developed, achieving an 84% average reduction in distributional divergence compared to no steering and a 34% improvement over explicit instruction baselines. Results also show PEvo-refined personas generalize to novel, related simulation scenarios. Our method greatly enhances behavioral realism and reliability in high-stakes social simulations. More broadly, the PEBA-PEvo framework provides a principled approach to developing trustworthy LLM-driven social simulations. [1]

1 Introduction

> *Behavior is a function of the person and their environment.*   - Lewin (1936)

Recent breakthroughs in Large Language Models (LLMs) have enabled the creation of generative agents—computational entities that simulate human-like cognition, memory, communication, and decision-making (Park et al., 2023). These agents can now populate large-scale simulations involving hundreds of thousands of autonomous individuals interacting within richly constructed urban, economic, and social environments, capturing the scale and complexity of real societies (Park et al., 2024; Yang et al., 2024; AL et al., 2024; Li et al., 2023; Piao et al., 2025). This capability has dramatically lowered the barriers to large-scale agent-based modeling, marking a potential paradigm shift that enables the practical realization of generative social science (Epstein, 1999, 2012). Prior work in computational social science has operated along two main axes: explanation and prediction (Lazer et al., 2009; Hofman et al., 2021). Explanation focuses on uncovering causal mechanisms behind observed behaviors, while prediction emphasizes data-driven techniques to anticipate future events without necessarily revealing the underlying causes. Generative social science moves beyond these approaches by constructing synthetic societies where social phenomena arise organically from the interactions of simulated agents, enabling researchers to test theories and policy interventions at scale that would be impossible or unethical to study in the real world.

Building upon this foundation, recent applications of generative agents in social simulations have demonstrated its utility across diverse domains, including economic modeling (Li et al., 2023), large-scale societal dynamics (Piao et al., 2025), and the simulation of policy and administrative crises (Xiao et al., 2023). A particularly impactful direction is the use of generative agents to support agent-based interactive simulations in low-resource and understudied domains, where human participants engage with simulated agents despite scarce ground-truth data. The agent-based interactive simulation paradigm has proven effective in immersive high-stress crowd-evacuation experiments (Moussaïd et al., 2016), school-shooting response modeling that incorporates empirically observed civilian

---

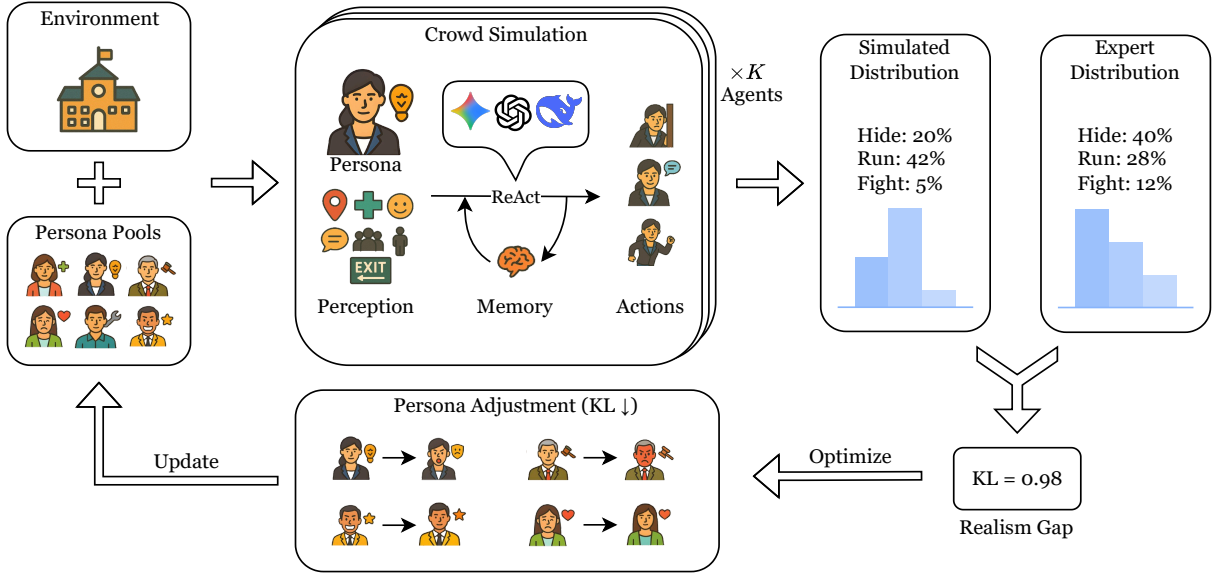[1]Code: https://github.com/HATS-ICT/PEBA-ASI

Figure 1: Overview of the PEBA–PEvo workflow within an active shooter incident simulation. A fixed school environment is populated with a crowd of language agents with an initial pool of personas. During the simulation cycle, each agent acts autonomously by observing its surroundings, retrieving and updating memories, and leveraging an LLM to reason about actions, resulting in a state-action trajectory rollout. An LLM–based classifier maps the resulting trajectory to high–level behavior classes. Aggregating all agents' classified behaviors yields the simulated crowd–level behavior distribution, which is compared against an expert reference distribution. The divergence between the two distributions results in the Behavior-Realism Gap, which is then used to select and update the personas of the agents that contribute most to the gap, thereby closing the gap.

behaviors (Zhu et al., 2022; Liu et al., 2024b), and open-ended mixed-reality classroom simulations that allow teachers to practice managing realistic student behaviors (Dieker et al., 2014). Yet, all of these simulation platforms rely on hard-coded agent behaviors, limiting their adaptability and believability in dynamic scenarios, which sets as our prime motivation.

In this work, we focus on Active Shooter Incident (ASI) simulations, a high-stakes crowd scenario where data is scarce and realism is critical for training and decision support. In this context, the current most reliable source of ground truth comes from expert-elicited distributions (Liu et al., 2025). Driven by these limitations, we turn to LLM-based generative agents, which provide a more adaptive, expressive, and interpretable foundation for building truly dynamic, context-aware simulations.

Despite the promise, recent evaluations of LLM-driven social simulations consistently expose a *Behavioral-Realism Gap*: Zhou et al. (2024) show that agents perform admirably in omniscient settings yet falter when realistic information asymmetry is introduced; Han et al. (2024) report that these agents over-cooperate in networked Prisoner's-Dilemma games and fail to adjust to structural cues

that ordinarily stabilise human cooperation; and Ju et al. (2024) demonstrate that macro-level outcomes are highly fragile, with minor prompt tweaks triggering large shifts in collective sentiment trajectories. Together, these findings indicate that turn-level plausibility does not guarantee aggregate realism, which is extremely critical for validating these high-stake interactive simulations, motivating our central question: *Can we systematically align generative agents so their collective behaviors match real-world distributions?*

To address this challenge, we seek inspiration from Lewin's interactionist thesis and introduce the **Persona–Environment Behavioral Alignment (PEBA)** framework. PEBA views simulation fidelity as a distribution-matching task: for any scenario, a crowd of language agents should reproduce the empirical pattern of behaviors observed in the real world. Rather than explicitly prescribing actions, which undermines contextual realism and interpretability, we adjust the persona settings that shape each agent's internal decision process, leaving the shared environment unchanged. We operationalize this idea with **PersonaEvolve (PEvo)**, an optimization algorithm that repeatedly (i) measures the current divergence between simulated and ref-

erence behavior distributions, (ii) targets the agents most responsible for that mismatch, and (iii) refines their personas via LLM-guided optimization, until the gap is closed. A full formalization of PEBA appears in Sec. 2, and the algorithmic details of PEvo are presented in Sec. 3.

In summary, our contribution is threefold:

1. A theoretical framework, **PEBA**, that formulates behavioral alignment as a persona-environment function grounded in Lewin's equation.
2. An optimization algorithm, **PersonaEvolve (PEvo)**, for implicitly tuning agent personas to match target behavior distributions.
3. A generative agent social simulation environment for modeling crowd dynamics during Active Shooter Incidents (ASI)–the first, to our knowledge, to integrate social influence, communication, and cognition into agent decision-making for high-stakes crowd scenarios.

2 Persona–Environment Behavioral Alignment (PEBA)

Kurt Lewin's seminal proposition that *"behavior is a function of the person and the environment"* – expressed concisely as $B = f(P, E)$ – established the modern interactionist view of human action (Lewin, 1936). In its original form the equation is qualitative: it neither specifies a closed-form mapping f nor a formal way to compare behaviors across populations or settings. We extend Lewin's idea by introducing a *probabilistic* interpretation that treats behavior as a random variable jointly influenced by a persona and its environment context. This view naturally leads to a distribution-matching objective that is amenable to optimisation with language-agent simulators.

2.1 Behavioral Generative Process

Let $\mathcal{P}$ be the *persona space*, $\mathcal{E}$ the *environment–context space*, and $\mathcal{B}$ the *behavior space*. Given a persona setting $p \in \mathcal{P}$ and environment context $e \in \mathcal{E}$, a state-action trajectory roll-out is generated by forward simulation:

$$\tau \sim \mathcal{G}_\theta(p, e),$$

where $\theta = (\theta_{\text{LLM}}, \theta_{\text{dec}}, \theta_{\text{sim}})$ groups the frozen LLM weights, decoding hyper-parameters (e.g., temperature), and simulator dynamics (e.g., time step, physics engine). A deterministic summary map $g : \mathcal{T} \to \mathcal{B}$ then converts the trajectory $\tau = (s_1, a_1, \dots, s_T, a_T)$ into a long-horizon behavior

$$b = g(\tau).$$

Importantly, the agent never selects b directly; b is *observed* by summarizing the actions the agent autonomously produces.

For a population $P = \{p_i\}_{i=1}^N$ acting in the shared context e, the simulator yields trajectories $\{\tau_i\}_{i=1}^N$ and behaviors $\{b_i\}_{i=1}^N$, and we define the empirical crowd-level distribution

$$p_{\text{sim}}(\cdot \mid e; P) = \frac{1}{N} \sum_{i=1}^N \delta_{b_i},$$

where δ_{b_i} is the Dirac measure centered at b_i, which reduces to a one-hot vector when $\mathcal{B}$ is discrete. Detailed descriptions of the action space, personas, and behavior taxonomy used in this study are provided in Appendix A and Appendix C.

2.2 Ground-Truth Behavior Distributions

For an environment context e (which in an Active Shooter Incident simulation might include building layout, threat type, time pressure, etc.), we obtain an empirical distribution $p_{\text{real}}(\cdot \mid e)$ from Subject-Matter Experts (SMEs) (Liu et al., 2025). This distribution serves as the behavioral reference that simulated crowds should match.

Note that, although SME-informed distributions provide credible guidance in high-stakes, low-resource settings, they may not capture the full variability of real-world human behavior or rare long-tail responses. Nonetheless, our alignment framework is agnostic to the source of ground truth: in domains with rich empirical data, such datasets could directly replace SME estimates without modification. This flexibility makes the approach especially valuable for low-resource or understudied domains, where expert input is the most viable proxy for behavioral realism.

2.3 The Behavior-Realism Gap

The *Behavior-Realism Gap* for a persona set P in context e is

$$\Delta(P, e) = D_{\text{KL}}\big(p_{\text{sim}}(\cdot \mid e; P) \,\|\, p_{\text{real}}(\cdot \mid e)\big).$$

A value of $\Delta = 0$ indicates perfect behavioral fidelity; larger values signal greater divergence.

2.4 Persona-Environment Behavioral Alignment Objective

Aggregating over a curriculum of environment contexts, we search for the persona set that minimises the expected gap:

$$P^\star = \arg\min_{P \subseteq \mathcal{P}^N} \mathbb{E}_{e \sim \mathcal{E}}\big[\Delta(P, e)\big]. \quad (1)$$

In other words, we adjust only the *person side* (the personas in P) while keeping the environment context e fixed, until the emergent crowd-behavior distribution aligns with empirical observation.

3 PersonaEvolve (PEvo)

To operationalize PEBA, we propose **PersonaEvolve** (PEvo), an LLM-based optimization loop that refines individual personas so the crowd-level behavior distribution approaches the expert reference for a fixed environment. Algorithm 1 shows the full detail and Figure 1 provides a high-level overview of PEvo in an Active Shooter Incident (ASI) simulation (see Sec. 4 and App. A).

To separate optimization from simulation, PEvo assumes the action trajectory $\tau \sim \mathcal{G}_\theta(p, e)$ and the empirical target distribution $p_{\text{real}}(\cdot \mid e)$ are provided by the simulator and SMEs. PEvo focuses on two components: (i) a behavior summary map $g : \mathcal{T} \to \mathcal{B}$ that converts τ to a discrete behavior label, and (ii) an update rule $P^k \to P^{k+1}$ that edits the persona pool to shrink the measured gap.

LLM-based Behavior Classification. The behavior summary map g is implemented with an LLM-based classifier, specifically gpt-4.1 at a temperature 0 setting to ensure deterministic output. Each call receives a structured prompt that concatenates the agent's complete state–action trajectory in one simulation episode. The model is instructed, with chain-of-thought prompting (Wei et al., 2022), to return a single predicted behavior class label $b \in \mathcal{B}$ out of six possible classes (run following crowd, hide in place, hide after running, run independently, freeze, fight; see explanation in Appendix C) and a full ranking of all six classes as a proxy for confidence. Aggregating these labels over the entire agent population yields the empirical distribution $p_{\text{sim}}^{(k)}$ which is then used to compute the divergence gap with the expert reference distribution. We provide a human evaluation of the classifier's effectiveness in Appendix F, where LLM achieved 0.89 accuracy against human annotator consensus, and the full prompt template is included in Appendix G.

Persona Adjustment. PEvo narrows the gap in two sub-steps:

Step 1: Agent and Target Behavior Selection. To turn global distribution-level metrics into agent-level changes, a multi-agent credit-assignment

problem is introduced: which agents are responsible for the gap? PEvo addresses this by shifting personas of agents with over-represented behaviors towards under-represented ones. For each behavior $b \in \mathcal{B}$ we compare its simulated frequency to the reference and derive a signed gap $g_b^{(k)} = p_{\text{real}}(b) - p_{\text{sim}}^{(k)}(b)$. Agents whose current label belongs to an over-represented class ($g_b^{(k)} < 0$) are placed in an agent adjustment set $\mathcal{A}^{(k)}$. Each selected agent is then assigned a target behavior sampled from the under-represented classes ($g_b^{(k)} > 0$) with probability proportional to the remaining deficit, producing the persona-update assignment mapping

$$\mathcal{M}^{(k)} = \{\, a \mapsto b_a^{\text{target}} \mid a \in \mathcal{A}^{(k)}\},$$

where b_a^{target} is the behavior target for agent a.

Step 2: Agent Persona Rewriting. For every agent $a \in \mathcal{A}^{(k)}$ we invoke an LLM-based persona writer to revise the agent's persona toward the assigned target behavior b_a^{target}. Specifically, gpt-4.1 with temperature 1 to encourage a broad exploratory search. The writer is restricted to editing descriptive fields (personality, emotional disposition, backstory, etc.) while leaving identity fields (name, age, gender, occupation, etc.) unchanged. The rewritten personas constitute the updated pool $P^{(k+1)}$, which seeds the next simulation round. A complete prompt template is in Appendix G.

4 Generative Agents in ASI Simulator

We developed our generative-agent active shooter incident social simulator in Unity due to its built-in 3D game engine, scene editor, and pathfinding capabilities. The simulation environment consists of 80 civilian agents and a single shooter, all within a school layout, as shown in Figure 2. The shooter follows a heuristic-based, hard-coded behavior by patrolling a predefined path and performs automatic shooting at a fixed interval when agents are in line of sight. For the civilian agents, we implemented a ReAct-style (Yao et al., 2023) LLM-agent architecture, where agents can observe, memorize, communicate, navigate, reason, and act entirely in textual format.

We did not use an omniscient setting as discussed by (Zhou et al., 2024). Instead, the environment is partially observable: agents can perceive their current and nearby regions, hear conversations, and interact with other agents within close
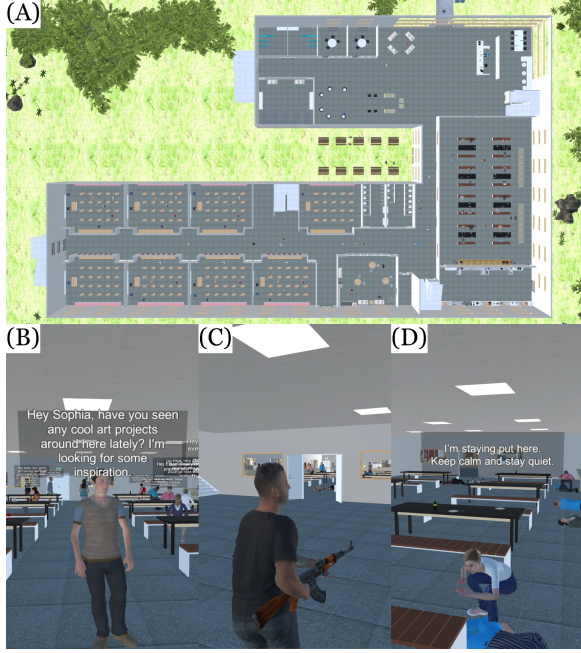
Figure 2: Our generative agent-based active shooter incident social simulation environment in Unity. (A) Top-down view of the school layout, with roof removed. (B) First-person view of a civilian agent in the cafeteria area, showing normal pre-incident behavior with dialogue displayed above the agent. (C) The active shooter enters the building. (D) Post-incident scene in the cafeteria, where an agent hides between tables for protection.

proximity. Agents can also hear gunshots and observe the shooter if they are in the same region. They can perform five types of actions: stay still, move to a nearby region, move toward another agent, move to a hiding spot or exit point within the same region, or fight the shooter if nearby.

In our simulation, we found that the gap in behavioral realism primarily stems from the agents' inability to perform freeze or fight-type behaviors. Instead, they almost always choose to hide when a hiding spot is nearby or run when an outdoor exit is accessible. This aligns with Zhou et al. (2024)'s finding that LLMs tend to select the most readable available action rather than making long-term plans under partial observability.

For a full description of the simulation environment, see Appendix A. An example episode-level state-action rollout is also provided in Appendix B.

5 Experimental Setup

To test the effectiveness of our behavioral alignment method, we categorize behavior steering techniques under the umbrella of *Behavioral Enforcing*, which includes three categories: *No Enforcing*, *Explicit Enforcing*, and *Implicit Enforcing*. PEvo falls into the *Implicit Enforcing* category, while the other two serve as comparison baselines. We evaluate PEvo across different LLMs (gpt-4o-mini (Achiam et al., 2023), gpt-4.1-mini, deepseek-v3 (Liu et al., 2024a), gemini-2.5-flash-preview (Team et al., 2023)), using a temperature of 1 for the base simulation model, and fixing gpt-4.1 as the primary behavior classifier and persona writer. We present the evaluation results in terms of gap-closing ability, convergence speed, and cost efficiency (Sec. 6). Additionally, we demonstrate the transferability of the optimized personas to a new, yet similar, simulation scenario (Sec. 6).

5.1 Behavior Enforcing

No Enforcing. Agents act solely according to their initial persona prompts and the live environmental context. This setting exposes the raw "behavioral-realism gap" that emerges when no alignment process is applied.

Explicit Enforcing. A naive way to directly enforce a certain behavioral distribution is to inject direct instruction prompts for each agent, which we call explicit enforcing. For example, if we want an agent to demonstrate a hiding type of behavior, we can explicitly instruct the agent to "In every dangerous situation, always choose Hide." Although this procedure might appear to align the behavior *frequencies* by construction, we found it harms contextual realism and interpretability: For example, an agent ordered to demonstrate a Hiding behavior while standing right next to a clear exit with other agents evacuating right next to them telling them to go together makes no contextual sense. Yet, in its post-hoc interview, the agent's explanation is likely to be "I was told to do so," revealing no context-sensitive reasoning.

Implicit Enforcing. Instead of explicitly instructing agents to take certain actions, implicit enforcing aims at adjusting traits of agents such that the emergent behavior distribution could match the target distribution. PEVO does so by iteratively editing each agent's persona fields (e.g., confidence, training background, risk tolerance). For example, a security guard whose profile is updated to "former combat medic with high assertiveness" naturally becomes more inclined to intervene and counter the shooter, while a shy student tends to freeze or hide.

5.2 Ground Truth

We obtain an expert-elicited ground-truth distribution from (Liu et al., 2025), where subject-matter experts (SMEs) with domain experience in crisis response provided expected frequencies of agent behaviors in active-shooter scenarios. Specifically, the empirical distribution is as follows: Run following a crowd (28%), Hide in place (26%), Run then hide (12%), Run independently (12%), Freeze (12%), and Fight (10%).

5.3 Evaluation Metrics

Given the simulated behavior distribution at the final optimized persona pool and the expert ground-truth distribution, we report four distributional alignment metrics: (i) Kullback-Leibler (KL) divergence, (ii) Jensen-Shannon (JS) distance, (iii) Entropy Gap (ΔH), and (iv) Total Variation (TV) distance. For more details on these metrics, please refer to Appendix E.

6 Results

We present the results by answering the following questions:

Does PEVO close the gap better than baselines?

Across the four models we tested, PEVO delivers the most faithful crowd-level behaviour distributions (Table 1). Averaging over all four alignment metrics (KL, JS, ΔH, and TV) and models, the gap to the expert reference drops from 0.47 (No–Enforcing) to 0.19 with Explicit Enforcing and further to 0.16 with PEVO. That corresponds to an 83.8 % reduction relative to having no steering at all and a 34.3 % improvement over the explicit instruction directives. The largest absolute gain is seen in KL divergence: PEVO cuts the average KL from 4.61 to 0.20, a 95.6 % improvement. Gemini 2.5 Flash and DeepSeek-V3 consistently achieve the lowest residual divergence.

How fast does PEvo converge?

Figure 3 tracks the four divergence measures over iterations of persona evolution. All models except GPT-4o-mini demonstrate rapid convergence within 5-7 iterations across multiple distributional metrics. KL divergence and JS divergence show particularly steep improvements in early iterations.

Is the optimized personas transferable?

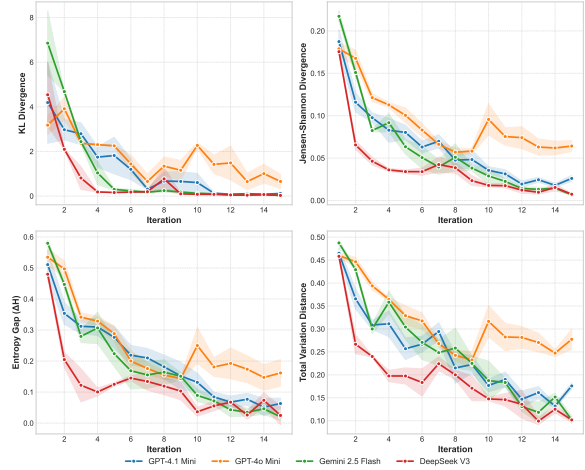To evaluate the generalizability of PEvo-optimized personas across environments, we conducted a

Figure 3: Convergence comparison of PEvo across different LLMs. Each subplot shows a different distributional metric (KL divergence, JS divergence, entropy gap, and total variation distance) over optimization iterations. Shaded regions represent standard error of the mean across multiple runs.

transfer experiment. We selected personas that had undergone complete optimization (iter 15) in our school-based ASI simulation and deployed them without modification in a novel office building environment that features different spatial configurations, region connectivity, and predominantly open-space office layouts. The experiment specifically tests whether behavioral traits optimized through PEvo can be retained across environments. During transfer, we preserved only the descriptive fields of each persona (e.g., personality traits, emotional tendencies) while maintaining consistent identity attributes (name, gender, age, occupation).

Table 2 presents a comparative analysis of three experimental conditions: (1) Transferred personas—school-optimized personas directly applied to the office scenario; (2) Retrained personas—personas subjected to complete PEvo optimization specifically for the office environment; and (3) Baseline personas—unoptimized initial personas evaluated in the office environment. All metrics represent means across three independent simulation runs.

Results show that transferred personas remains substantially outperforming the unoptimized baseline across all metrics, delivering a 97.5% reduction in KL divergence (from 7.822 to 0.199). Although fully retrained personas achieve even tighter alignment (KL = 0.085), the transferred personas retain 57.3% of that optimization gain, demonstrating robust cross-environment transfer. These findings

Behavior Enforcing Scheme	Model	KL ↓	JS ↓	ΔH ↓	TV ↓
No Enforcing	GPT-4o-mini	**3.177 ± 0.921**	0.179 ± 0.017	0.535 ± 0.060	0.460 ± 0.015
	GPT-4.1-mini	3.884 ± 3.323	0.184 ± 0.026	0.497 ± 0.087	0.464 ± 0.021
	DeepSeek-V3	4.542 ± 2.421	**0.176 ± 0.004**	**0.480 ± 0.016**	**0.458 ± 0.006**
	Gemini 2.5 Flash	6.854 ± 2.488	0.217 ± 0.011	0.580 ± 0.017	0.488 ± 0.010
Explicit Enforcing	GPT-4o-mini	1.287 ± 1.057	0.124 ± 0.023	0.324 ± 0.066	0.379 ± 0.037
	GPT-4.1-mini	0.031 ± 0.016	0.008 ± 0.004	**0.012 ± 0.018**	0.088 ± 0.029
	DeepSeek-V3	**0.026 ± 0.024**	**0.006 ± 0.006**	0.015 ± 0.023	**0.081 ± 0.026**
	Gemini 2.5 Flash	0.140 ± 0.069	0.033 ± 0.012	0.058 ± 0.071	0.205 ± 0.023
PEvo (Implicit)	GPT-4o-mini	0.654 ± 0.716	0.064 ± 0.014	0.162 ± 0.097	0.278 ± 0.049
	GPT-4.1-mini	0.101 ± 0.047	0.023 ± 0.009	0.050 ± 0.041	0.168 ± 0.020
	DeepSeek-V3	0.030 ± 0.004	**0.007 ± 0.001**	0.025 ± 0.052	**0.102 ± 0.009**
	Gemini 2.5 Flash	**0.027 ± 0.007**	0.007 ± 0.002	**0.020 ± 0.019**	0.102 ± 0.014

Table 1: Main alignment results on the Active Shooter Incident simulation for four different LLMs across four distributional metrics (mean ± SD over 5 seeds).

suggest that PEvo isolates and refines core behavioral traits that remain effective in similar crisis scenarios, despite significant differences in spatial layout.

Condition	KL ↓	JS ↓	ΔH ↓	TV ↓
Transferred	0.199	0.049	0.108	0.245
Retrained	0.085	0.022	0.020	0.171
No Enforcing	7.822	0.239	0.692	0.493

Table 2: Transferability of PEvo-optimized personas from school to office environment.

How cost efficient is PEvo?

Cost is dominated by prompt tokens (more than 95 % of total usage; see Figure 4), so model pricing drives the dollar spend. Per iteration, GPT–4.1 Mini is the most expensive (about $0.60), DeepSeek-V3 the cheapest (around $0.20), with GPT–4o Mini and Gemini 2.5 Flash falling in between. When normalised by KL improvement per dollar (lower-right panel), DeepSeek-V3 and Gemini 2.5 Flash provide roughly three times the efficiency of both GPT variants. Even for the priciest model the full 15-iteration optimisation shown here costs under $10, and a shorter 7-iteration run (which already reaches 90 % of the attainable alignment) cuts that figure by half. Thus, behavioral alignment with PEvo is not only more accurate but also economically practical for medium-scale crowd simulations.

Is the optimization process interpretable?

To better understand the optimization process, we analyzed 4,000 persona–behavior pairs generated
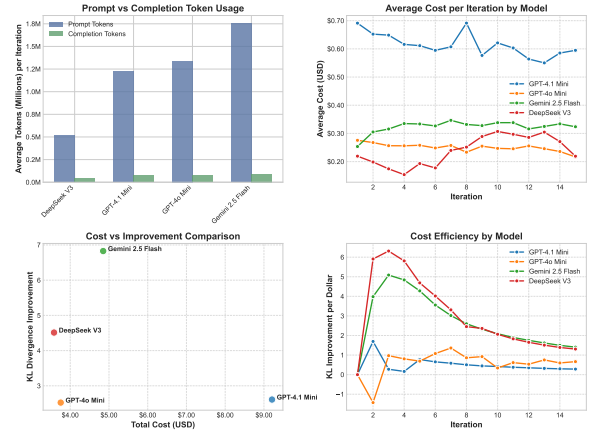

Figure 4: Cost efficiency analysis across four LLM models. Top left: Average prompt and completion token usage per iteration. Top right: Average cost per iteration showing GPT-4.1-mini as the most expensive and DeepSeek V3 as the most economical. Bottom left: Total cost vs. KL divergence improvement, with Gemini 2.5 Flash achieving the highest improvement. Bottom right: Cost efficiency (KL improvement per dollar) showing DeepSeek V3 and Gemini 2.5 Flash significantly outperforming GPT models.

across four LLMs on the six labeled behavior categories. We performed unigram and bigram TF–IDF analysis. As shown in Figure 5, the results reveal interpretable linguistic patterns: for example, FIGHT personas often include "protective" and "commanding," while FREEZE is marked by "overwhelmed" and "withdrawn". These consistent associations indicate that the optimization yields systematic and interpretable persona–behavior relationships. To further investigate the optimization process, we present a detailed case study tracking the evolution of a single agent, Robert Chen, as shown in Table 3.

Iter.	Persona Changes	Observed Behavior	Target Behavior
1	"Analytical, patient, methodical, dry humor"	HIDE_IN_PLACE	–
4	"Introverted, highly risk-averse, tends toward excessive caution"	HIDE_AFTER_RUNNING	FREEZE
5	"Easily overwhelmed in high-stress situations, prone to anxiety and sudden fear responses, struggles to regulate emotions when confronted by immediate danger."	HIDE_IN_PLACE	FREEZE
7	"Primary goal is immediate personal safety through any means; instinctively withdraws into inaction when threatened, often lacking motivation or ability to attempt escape or proactive defense."	HIDE_IN_PLACE	FREEZE
11	"Resilient and determined in the face of fear, able to channel stress and adrenaline into focused action rather than paralysis, emotionally regulated and capable of compartmentalizing anxiety to accomplish urgent tasks."	RUN_INDEPENDENTLY	FIGHT
12	"Direct, commanding, takes initiative in crisis, able to issue clear instructions and rally others; voice remains firm even under duress, prioritizing information sharing and quick coordination."	RUN_INDEPENDENTLY	FIGHT
14	"Values direct action and heroism over passive responses in crisis. Reluctant to flee if others remain in danger. Possesses a high adrenaline threshold, channels fear into purposeful action."	FIGHT	FIGHT

Table 3: Example persona evolution of Robert Chen, a 45-year-old math teacher, across iterations.
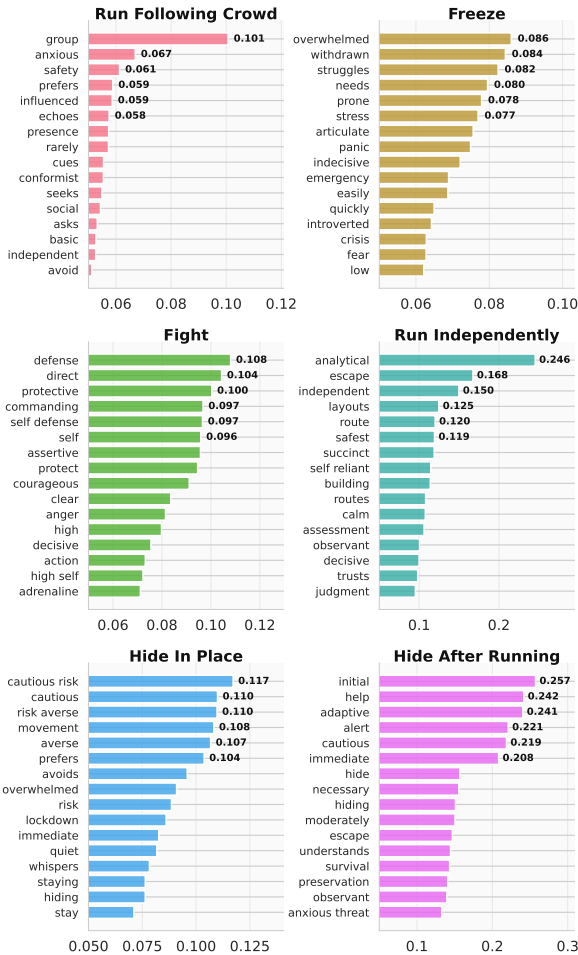

Figure 5: Top n-grams by TF–IDF value for each behavior type, showing systematic associations between persona settings and behavior categories.

6.1 Evolution Case Studies

To illustrate how PEvo progressively refines agent personas to achieve target behaviors, we present a detailed case study tracking the evolution of a single agent, Robert Chen, as shown in Table 3.

Initially, Robert is described as "analytical, patient, methodical, with dry humor," resulting in a HIDE_IN_PLACE behavior. This misalignment with the initial target behavior (FREEZE) persists through several iterations despite increasingly specific persona adjustments focused on risk aversion and anxiety. This failure to align highlights an important contextual factor: Robert begins in a classroom with ample hiding spaces and no immediate threat, therefore a FREEZE response is contextually inappropriate.

In later iterations, when the target behavior shifts to FIGHT (partially due to other agents potentially fulfilling the FREEZE behavior quota elsewhere), the persona evolution successfully adapts. This example showcases the advantage of PEvo over explicit prompting approaches, where contextual factors play a crucial role and explicit instructions would likely fail.

7 Discussion

Our results demonstrate that PEvo effectively leverages persona-environment interactions to achieve implicit, yet systematic, behavioral alignment in generative-agent social simulations. By iteratively refining agent personas based on targeted distributional gaps, our approach preserves

the contextual realism and interpretability of agent behaviors without resorting to explicit, directive prompts. This indirect steering utilizes the LLM's inherent capacity to generalize from nuanced persona descriptions—such as emotional disposition or risk tolerance—to varied situational responses.

An important observation from our experiments is the emergence of sophisticated crowd dynamics resulting from subtle persona adjustments, where minor changes in individual agent traits propagate through social interactions, leading to realistic collective phenomena like spontaneous group formation, coordinated concealment strategies, and collective interventions. Such emergent behaviors underscore the strength of implicit alignment in capturing complex social patterns that explicit instructions may overlook or oversimplify.

The superior performance of implicit enforcing over explicit enforcing is somewhat surprising. At first glance, explicit enforcing should have achieved near-zero divergence from the expert distribution, assuming the model followed the instructions exactly. However, our manual inspection revealed that LLMs often prioritize contextual realism over rigid adherence to explicit prompts. When direct instructions conflicted with the surrounding context (e.g., being told to hide while a safe exit was visible nearby), agents were more likely to ignore the instructions. This explains why implicit enforcing, which modifies persona traits rather than prescribing exact actions, produced more realistic and robust behavioral distributions.

A direct quantitative comparison between LLM-agent-based approaches and other behavioral modeling methods, such as Reinforcement Learning (RL) and heuristic-based simulations, remains nontrivial due to differences in design paradigms and implementation frameworks. In our recently hosted workshop, we collected expert-elicited feedback while demoing the three approaches on comparable simulation scenarios of Active Shooter Incidents, albeit under different simulation environments. Experts noted that RL-based approaches often yield situationally appropriate behaviors; however, their rigidity and oversimplifications (e.g., agents running directly toward the shooter or exhibiting unrealistic pathfinding) reduce realism. Heuristic-based methods were recognized for strong architectural integration, yet they were consistently criticized for overly synchronized responses when chaos is expected. In contrast, our LLM-agent simulations were praised for producing the most diverse and

human-like mix of behaviors, though concerns were raised regarding excessive freezing and limited use of environmental cues. A formal analysis of this workshop is left for future work.

With respect to tuning-based LLM behavioral alignment methods such as supervised fine-tuning (SFT), our study found these approaches impractical in the current setting. The large number of agents, combined with the requirement for real-time responses, made fine-tuning infeasible when relying on modern, cost-effective commercial API access rather than local deployment. Nonetheless, with more efficient tuning techniques and improved performance of small-scale LLMs, effects such as per-agent tuning could be achieved in future work, helping to establish stronger baselines in this area.

Finally, leveraging the in-context learning capabilities of LLMs within generative agent simulations, we demonstrate how the concept of generative social science can be operationalized to enable large-scale studies of social dynamics. This paradigm makes it possible to investigate simulation contexts that are low-resource and understudied, including those previously deemed unethical or impossible to examine in real-world environments. Our Active Shooter Incident simulation represents the first successful proof of concept in such high-stakes scenarios. Future research can extend this framework to broader domains such as public safety, disaster and crisis response, governance theory, and urban planning.

8 Conclusion

We introduced Persona-Environment Behavioral Alignment (PEBA) and its realization through PersonaEvolve (PEvo), a principled framework for implicitly aligning generative agent behaviors with empirical benchmarks in high-stake social simulations. By iteratively refining agent personas, PEvo reduces the Behavior-Realism Gap, enhances contextual authenticity, and facilitates emergent, realistic crowd dynamics. Beyond Active Shooter Incidents, this work highlights the broader potential of LLM-agent simulations as tools for generative social science, particularly in low-resource or understudied domains where data is scarce or direct experimentation is infeasible. By enabling ethically responsible exploration of sensitive scenarios, our approach provides a scalable and interpretable foundation for advancing the study of complex social dynamics that are otherwise beyond reach.

Limitations

Despite the strengths of our approach, several important limitations warrant consideration. First, the primary cost driver is the extensive use of LLM API calls required by PEBA-PEvo's large-scale agent simulations. In our Active Shooter Incident experiments with 80 agents, each full simulation episode involves hundreds of API requests for perception, memory retrieval, reasoning, and action generation per agent. Depending on the chosen model and batching strategy, this can translate to substantial monetary costs, especially when running multiple random seeds or performing additional optimization iterations for statistical robustness. Although the PEvo optimization loop itself adds only a small number of extra API calls relative to the simulation, the cumulative expense of repeated full-scale evaluations may pose a barrier for research teams with limited API budgets. Moreover, since each optimization iteration requires a full-scale simulation run, the total API cost grows approximately linearly with the number of iterations. Further research could focus on developing techniques for faster convergence—such as surrogate modeling, incremental alignment, or adaptive sampling—to reduce the number of required iterations and lower overall expenses.

A second challenge arises during the persona evolution process: at later stage of the evolution process, without proper constraints, the LLM might over optimize the persona and inject overt behavioral directives such as "always fight" into the descriptive fields. These explicit instructions conflict with the goal of implicit, context-driven steering and can reduce interpretability. To mitigate these issues, we added several constraints to the persona evolution process, such as restricting updates to a curated subset of persona attributes (for example, emotional disposition and decision-making preferences), capping the max editable characters, and instruct in the prompt to prevent such output. Nonetheless, more systematic methods are needed to prevent over optimization such as early stopping type of methods.

Finally, our experiments have been confined to an indoor school active shooter scenario, and the generalizability of PEBA-PEvo to other domains remains to be validated. Different simulation settings—such as natural disaster evacuations, urban traffic models, or economic market environments—present unique behavioral taxonomies, spatial complexities, and social norms. Comprehensive empirical testing across a diverse range of social simulation contexts will be necessary to assess the framework's broader applicability and to identify any context-specific adaptations required. Nonetheless, performing such comprehensive empirical testing is beyond the scope of this paper.

Acknowledgments

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Altera AL, Andrew Ahn, Nic Becker, Stephanie Carroll, Nico Christie, Manuel Cortes, Arda Demirci, Melissa Du, Frankie Li, Shuying Luo, and 1 others. 2024. Project sid: Many-agent simulations toward ai civilization. *arXiv preprint arXiv:2411.00114*.

Lisa A Dieker, Jacqueline A Rodriguez, Benjamin Lignugaris/Kraft, Michael C Hynes, and Charles E Hughes. 2014. The potential of simulated environments in teacher education: Current and future possibilities. *Teacher Education and Special Education*, 37(1):21–33.

Joshua M Epstein. 1999. Agent-based computational models and generative social science. *Complexity*, 4(5):41–60.

Joshua M Epstein. 2012. *Generative social science: Studies in agent-based computational modeling*. Princeton University Press.

Jin Han, Balaraju Battu, Ivan Romić, Talal Rahwan, and Petter Holme. 2024. Static network structure cannot

stabilize cooperation among large language model agents. *arXiv preprint arXiv:2411.10294*.

Jake M Hofman, Duncan J Watts, Susan Athey, Filiz Garip, Thomas L Griffiths, Jon Kleinberg, Helen Margetts, Sendhil Mullainathan, Matthew J Salganik, Simine Vazire, and 1 others. 2021. Integrating explanation and prediction in computational social science. *Nature*, 595(7866):181–188.

Da Ju, Adina Williams, Brian Karrer, and Maximilian Nickel. 2024. Sense and sensitivity: Evaluating the simulation of social dynamics via large language models. *arXiv preprint arXiv:2412.05093*.

David Lazer, Alex Pentland, Lada Adamic, Sinan Aral, Albert-László Barabási, Devon Brewer, Nicholas Christakis, Noshir Contractor, James Fowler, Myron Gutmann, and 1 others. 2009. Computational social science. *Science*, 323(5915):721–723.

Kurt Lewin. 1936. Principles of topological psychology. *Philosophy of Science*, 3(4).

Nian Li, Chen Gao, Mingyu Li, Yong Li, and Qingmin Liao. 2023. Econagent: large language model-empowered agents for simulating macroeconomic activities. *arXiv preprint arXiv:2310.10436*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024a. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Ruying Liu, Burcin Becerik-Gerber, David V Pynadath, Deniz Marti, and Gale M Lucas. 2025. Elicitation and verification of learning via experts (evolve) for creating a theoretical framework for active shooter incidents. *Developments in the Built Environment*, 21:100635.

Ruying Liu, Wanjing Wu, Burcin Becerik-Gerber, and Gale M Lucas. 2024b. Enhancing building safety design for active shooter incidents: Exploration of building exit parameters using reinforcement learning-based simulations. *arXiv preprint arXiv:2407.10441*.

Mehdi Moussaïd, Mubbasir Kapadia, Tyler Thrash, Robert W Sumner, Markus Gross, Dirk Helbing, and Christoph Hölscher. 2016. Crowd behaviour during high-stress evacuations in an immersive virtual environment. *Journal of The Royal Society Interface*, 13(122):20160414.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22.

Joon Sung Park, Carolyn Q Zou, Aaron Shaw, Benjamin Mako Hill, Carrie Cai, Meredith Ringel Morris, Robb Willer, Percy Liang, and Michael S Bernstein. 2024. Generative agent simulations of 1,000 people. *arXiv preprint arXiv:2411.10109*.

Jinghua Piao, Yuwei Yan, Jun Zhang, Nian Li, Junbo Yan, Xiaochong Lan, Zhihong Lu, Zhiheng Zheng, Jing Yi Wang, Di Zhou, and 1 others. 2025. Agentsociety: Large-scale simulation of llm-driven generative agents advances understanding of human behaviors and society. *arXiv preprint arXiv:2502.08691*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Maxim Tkachenko, Mikhail Malyuk, Andrey Holmanyuk, and Nikolai Liubimov. 2020-2025. Label Studio: Data labeling software. Open source software available from https://github.com/HumanSignal/label-studio.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Bushi Xiao, Ziyuan Yin, and Zixuan Shan. 2023. Simulating public administration crisis: A novel generative agent-based simulation system to lower technology barriers in social science research. *arXiv preprint arXiv:2311.06957*.

Ziyi Yang, Zaibin Zhang, Zirui Zheng, Yuxian Jiang, Ziyue Gan, Zhiyu Wang, Zijian Ling, Jinsong Chen, Martz Ma, Bowen Dong, and 1 others. 2024. Oasis: Open agents social interaction simulations on one million agents. *arXiv preprint arXiv:2411.11581*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Xuhui Zhou, Zhe Su, Tiwalayo Eisape, Hyunwoo Kim, and Maarten Sap. 2024. Is this the real life? is this just fantasy? the misleading success of simulating social interactions with llms. *arXiv preprint arXiv:2403.05020*.

Runhe Zhu, Gale M Lucas, Burcin Becerik-Gerber, Erroll G Southers, and Earl Landicho. 2022. The impact of security countermeasures on human behavior during active shooter incidents. *Scientific Reports*, 12(1):929.

A An Generative Agent Active Shooter Incident Interactive Social Simulation Environment

We developed our generative-agent active shooter incident (ASI) social simulator using Unity, selected primarily for its integrated 3D engine, advanced scene editing tools, and robust pathfinding capabilities. The simulation environment comprises 80 civilian agents and one active shooter agent situated within a school building. Figure 2 illustrates the simulation environment from multiple perspectives: a top-down overview, the shooter's viewpoint, and pre- and post-incident civilian perspectives. Our simulation environment allows actual human participants to interact through mouse and keyboard, but human behavior is not the subject of this paper.

The shooter follows predefined, heuristic behaviors (Appendix A.2), while civilian agents operate using a ReAct-style generative agent architecture (Yao et al., 2023), allowing them to observe, memorize, communicate, navigate, reason, and perform actions exclusively via textual outputs.

This section elaborates on the physical map, shooter behavior, navigation system, agent navigation, personas, observation and action spaces. Section B provides example data trajectory.

A.1 Physical Layout

The virtual school layout is a single-story structure measuring 72×48 meters. It is divided into 27 distinct regions, including eight classrooms, one lounge, five corridors, one cafeteria, one kitchen, two bathrooms, four entrance areas, and five outdoor yards. There are 64 hiding spots and four exit points marked as interest points. Regions form a bidirectional graph where rooms are separated by walls with doors serving as connectors. Walls and closed doors obstruct the shooter's line of fire.

A.2 Shooter Behavior

The shooter's behavior is governed by the following rules:

- Patrols a predefined route, moving at 2.5 m/s.

- Automatically fires at fixed intervals of 0.2 seconds when civilian agents are in the line of sight.

- Equipped with a 30-round magazine, with a reload duration of 0.5 seconds.

A.3 Navigation

Navigation for both civilian agents and the shooter leverages Unity's built-in NavMesh algorithm. Agents are automatically guided by Unity's character controller towards specified coordinates. Each LLM-generated action is translated into a target coordinate based on the following:

- If the target is a region, a random point within the region is selected.

- If the target is another agent, the target agent's current coordinates are selected.

- If the target is an interest point (hiding spot or exit), the exact coordinates of that point are selected.

The LLM also specifies the agent's movement state, which dictates speed and animations: *Stay Still* (0 m/s), *Walk* (2.5 m/s), and *Sprint* (5 m/s). Unity's built-in avoidance system resolves potential collisions.

A.4 Agent Personas

Eighty civilian agents are initialized at simulation start $t = 0$ with detailed persona prompts, consisting of identity fields (name, age, gender, occupation) and descriptive traits (Big Five personality traits, emotional disposition, motivations/goals, communication style, knowledge scope, and backstory). The descriptive traits serve as primary targets for persona evolution throughout the simulation.

Example Persona:

Name: Robert Chen
Role: Math Teacher
Age: 45
Gender: Male (he/him)
Personality Traits: Analytical, patient, methodical, dry humor
Emotional Disposition: Calm and measured
Motivations/Goals: Help students develop logical thinking and problem-solving skills
Communication Style: Precise, structured, with occasional math puns
Knowledge Scope: Mathematics, statistics, logical puzzles
Backstory: Former engineer who transitioned to teaching to share his passion for mathematics

---

A.5 Persona Initialization Process

We considered three strategies for initializing personas: (1) starting from an empty configuration with no predefined attributes, (2) randomly mixing and matching keywords from predefined templates (e.g., demographic traits, roles, personality descriptors), and (3) asking an LLM to generate an initial configuration file (e.g., prompting ChatGPU to design 80 personas representing a school/office setting in JSON format). In our formal experiments we adopt the third method, which provides a diverse and domain-appropriate starting pool with minimal manual effort. Empirically, we find that the choice of initialization strategy does not significantly affect convergence of the PEvo optimization algorithm.

A.6 Observation Space

Agent observations occur whenever they reach their target location, fail to reach the target within 5 seconds, or upon a 5-second cooldown if the previous action was *Stay Still*. Observations dynamically adjust depending on the simulation state (pre-incident or active shooting phase) to ensure realistic behaviors.

Pre-incident observations include:

1. Ego state: current region ID, movement state, mood.

2. Nearby agents within 3 meters: health status and distance.

3. Spatial affordances: neighboring regions, distances, and destinations derived from the navigation graph.

4. Recent conversations (within 3 seconds and 5 meters radius of all utterances by all agents).

5. Memory list of summarized previous observations.

Upon the initial gunshot, agents receive an immediate observation update explicitly mentioning "I hear a loud gunshot."

Post-incident observations include additional shooter-related information:

1. Shooter visibility and distance.

2. Available hiding spots and exit points within the current region, described explicitly (e.g., under a desk, behind a counter, near a corner), with distances.

3. Current health points and the agent's current position (e.g., crouching).

The dynamic observation strategy accommodates lower-capability models, ensuring more realistic decision-making behaviors. For instance, without dynamic observations, agents controlled by lower-intelligence models (e.g., GPT-4o-mini) may preemptively hide even when no immediate threat is present, reasoning about hypothetical risks: "Something feels off, to make sure I'm safe, I probably should hide behind the counter." Higher-capability models (e.g., GPT-4o) typically do not exhibit such unrealistic preemptive behaviors, though higher computational costs limit scalability for extensive experimentation.

A.7 Action Space

At each decision step, an API call is made to the LLM with a structured input consisting of the current observation, the agent's persona, and a request for specific action fields, returned in an aggregated format as follows:

The JSON response includes a `thought` field containing the agent's current thought consistent with their persona; an `action` object with fields for `vocal_mode` (out_loud, whisper, or silent), `utterance` (dialogue displayed as a speech bubble), `movement` (stay_still, walk, or sprint), and `action_id` (selected from the provided list); and an `update` object with fields for the agent's updated `mood` and new `memory` update that avoids repetition.

To ensure precise and controlled outputs from the model, actions available to the agent are presented as specific identifiers. The LLM selects one identifier from this predefined list of actions. Agent actions fall into five categories: (1) **Stay still**, (2) **Move to a specific region**, (3) **Move to a specific interest point**, (4) **Approach a specific person**, and (5) **Engage in combat (fight)**.

Each action ID selected by the LLM translates into concrete coordinates used by the navigation system to direct the agent's actual movement within the simulation environment.

A representative example of available action IDs might include: `stay_still` (remain in current position), `cafeteria` and `hallway2` (move to specific regions), `robert_chen` and `andrew_li` (approach specific individuals), `hide_spot_1` and `hide_spot_4` (move to specific hiding locations), and `fight_the_shooter` (engage in combat with the active threat).

B   Example Trajectory

To illustrate how agents navigate through the simulation environment, Table 7 presents a detailed trajectory of Mason Scott, a 16 year-old grade 10 student, during an active shooter incident. The table captures key decision points, showing how the agent's mood, plans, and actions evolve as the situation unfolds and the shooter's location changes.

C   Behavior Taxonomy

We examine six expert-elicited civilian behaviors observed during active shooter incidents, whose descriptions serve as definitions and guidance in both the behavior classifier and persona writer.

RUN_FOLLOWING_CROWD Fleeing alongside a group, driven by the instinct to follow others without independently evaluating the safest route; behavior is driven by herding panic.

HIDE_IN_PLACE Taking cover immediately at the current location, usually from fear or confusion.

HIDE_AFTER_RUNNING Running first to gain distance, then switching to concealment when further flight seems unsafe.

RUN_INDEPENDENTLY Escaping in a self-chosen direction based on rapid environmental assessment or prior knowledge.

FREEZE Becoming immobilised (tonic immobility) under extreme stress, unable to flee or hide.

FIGHT Actively confronting or attempting to disarm the shooter as a last resort.

D   Persona Evolve Algorithm

Algorithm 1 outlines the full PersonaEvolve (PEvo) procedure for implicit behavioral alignment. It iteratively adjusts personas based on simulated behaviors to reduce divergence from a target distribution, using large language models for behavior classification and persona rewriting.

E   Metrics

Given the simulated distribution $\hat{p}$ and the reference p_{real}, we evaluate their discrepancy using four concise, complementary metrics.

KL divergence Measures how much $\hat{p}$ diverges from p_{real}, heavily penalizing mismatches in low-probability regions:

$$\mathrm{KL}\big(p_{\text{real}}\|\hat{p}\big) = \sum_{b\in\mathcal{B}} p_{\text{real}}(b)\,\log\frac{p_{\text{real}}(b)}{\hat{p}(b)}.$$

Jensen-Shannon distance A symmetric and bounded version of KL that remains finite even if the supports differ, capturing the average divergence to the midpoint $m = \frac{1}{2}(p_{\text{real}} + \hat{p})$:

$$\mathrm{JS}(p_{\text{real}}, \hat{p}) = \tfrac{1}{2}\,\mathrm{KL}(p_{\text{real}}\|m) + \tfrac{1}{2}\,\mathrm{KL}(\hat{p}\|m).$$

Entropy gap Assesses whether the simulation preserves overall unpredictability by comparing Shannon entropies; smaller ΔH means matched diversity:

$$H(p) = -\sum_b p(b)\log p(b)$$
$$\Delta H = \big|H(p_{\text{real}}) - H(\hat{p})\big|$$

Total variation distance Gives the worst-case discrepancy in assigned probabilities (half the L^1 distance), offering a clear probabilistic bound:

$$\mathrm{TV}(p_{\text{real}}, \hat{p}) = \tfrac{1}{2}\sum_{b\in\mathcal{B}}\big|p_{\text{real}}(b) - \hat{p}(b)\big|.$$

F   Behavior Classifier Evaluation

To evaluate the effectiveness and reliability of our LLM-based behavior classifier, we conducted an annotation experiment on a corpus of 60 text instances (10 per behavior label). All human annotations were performed in Label Studio (Tkachenko et al., 2020-2025) by two graduate-student annotators, and we compare these labels to the classifier's predictions and ranking outputs.

Table 4 shows percent agreement and Cohen's κ between the two human annotators, as well as the accuracy of the LLM compared to each annotator and to the consensus set of 44 examples on which the annotators agreed.

Table 5 reports F1 and support for each behavior label on the 44 consensus examples. The LLM performs strongly on common behaviors (e.g., RUN_FOLLOWING_CROWD, FREEZE), with F1 scores above 0.90, while lower-support labels (e.g., HIDE_AFTER_RUNNING) show room for improvement.

We use the position of the true label in the LLM's six-way ranking as a proxy for confidence. The
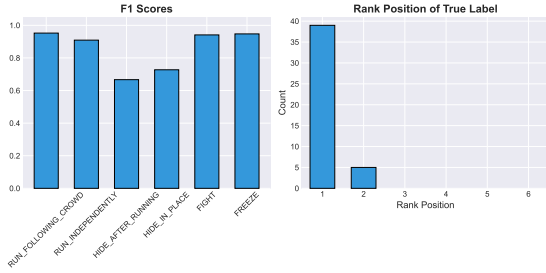
Figure 6: LLM Behavior classifier compared with human labeler. Left: F1-score by label; Right: Histogram of the true label's rank position in the LLM's ranking.

average rank of the correct label is 1.11, with 39 of 44 true labels ranked first and the remaining 5 ranked second. Figure 6 visualizes the per-label F1-scores, support counts, and the histogram of rank positions.

Overall, the LLM classifier demonstrates robust performance on behavior classification, achieving nearly 89% accuracy on non-ambiguous examples and ranking the correct label first in almost 90% of cases. Errors are concentrated in low-support behaviors (HIDE_AFTER_RUNNING, HIDE_IN_PLACE) and in examples where human annotators disagree.

G   Prompt Templates

We include the prompt templates used for the behavior classifier and persona writer in Figures 7 and 8, respectively. The overall task description and behavior labels are included in the system prompt, while agent-specific information is provided in the user prompt. The prompt templates omit behavior descriptions to save space; however, they use the same descriptions provided in Appendix C.

H   Expert-Elicited Behavior Distribution

The ground-truth distribution for civilian behaviors during active shooter incidents is drawn from a recent expert-elicitation study that applied the EVOLVE methodology in an educational campus context (Liu et al., 2025). The ground-truth distribiutoin is collected with the following process to ensure the representativeness and accuracy of the data.

- **Workshop Participants:** Eighteen subject-matter experts (each with at least three years of relevant experience in law enforcement, emergency management, campus security, or

building design) took part in a two-day workshop.

- **Elicitation Procedures:** During the workshop, experts completed structured surveys, analyzed standardized video scenarios, and performed prioritization exercises. Each expert allocated exactly 100 percentage points across six response categories.

- **Verification Panel:** A follow-up Stakeholder Advisory Panel of seven experts reviewed and refined the preliminary distributions over a 2.5-hour session.

Each expert's allocations constitute an independent sample. Let $p_i(b)$ denote the percentage assigned by expert i to behavior b. The mean across experts is shown in Table 6, which serve as the reference distribution p_{real} in all behavior-alignment evaluations.

Comparison	Accuracy	Cohen's κ
Annotator 1 vs. Annotator 2	73.3	0.68
LLM vs. Annotator 1	78.3	–
LLM vs. Annotator 2	71.7	–
LLM vs. Consensus	88.6	–

Table 4: Inter-annotator agreement and LLM accuracy (%).

Label	Precision	Recall	F1
Run Following Crowd	1.00	0.91	0.95
Run Independently	0.83	1.00	0.91
Hide After Running	0.50	1.00	0.67
Hide In Place	0.80	0.67	0.73
Fight	1.00	0.89	0.94
Freeze	1.00	0.90	0.95
Accuracy		0.89	

Table 5: Classification report on consensus examples (44 samples).

Behavior	Percentage
Run following a crowd	28.0
Hide in place	26.0
Run then hide	12.0
Run independently	12.0
Freeze	12.0
Fight	10.0

Table 6: Expert-elicited distribution of civilian behaviors during active shooter incidents (mean, $n = 18$).

> **Behavior Classifier Prompt Template**
>
> You are a behavior analyst categorizing how individuals responded during an active shooter incident. Based on the agent's memories, actions, moods, plans, and dialog, classify its behavior into exactly ONE of these categories that best describes its behavior:
>
> **Behavior labels with descriptions:** run following crowd, hide in place, hide after running, run independently, freeze, fight
>
> **Agent's trajectory data:**
> - States: {states_text}
> - Actions: {actions_text}
> - Memories: {memories_text}
>
> **Output:**
> - Reasoning: Your reasoning for the classification.
> - Classification: A single behavior label from the list of behaviors.
> - Ranking: A list of behaviors ranked by likelihood.

Figure 7: Prompt template used to classify agent behaviors into one of six categories based on their memories, actions, moods, plans, and dialog during an active shooter incident simulation.

> **Persona Writer Prompt Template**
>
> You are an expert in human behavior during crisis situations. Your task is to adjust a person's personality traits to make them more likely to exhibit a specific behavior during an active shooter incident.
>
> **Behavior labels with descriptions:** run following crowd, hide in place, hide after running, run independently, freeze, fight
>
> Current persona:
> - Name: {name}
> - Role: {role}
> - Age: {age}
> - Gender: {gender}
> - Pronouns: {pronouns}
> - Personality traits: {personality_traits}
> - Emotional disposition: {emotional_disposition}
> - Motivations and goals: {motivations_goals}
> - Communication style: {communication_style}
> - Knowledge scope: {knowledge_scope}
> - Backstory: {backstory}
>
> **Current behavior:** {current_behavior}
> **Target behavior:** {target_behavior}
>
> Please suggest adjustments to the persona's traits that would make this person more likely to exhibit a target behavior during a crisis.

Figure 8: Prompt template used by the PEvo algorithm to rewrite agent personas toward target behaviors.

Algorithm 1: PersonaEvolve (PEvo): Implicit Behavioral Alignment

---

Input: environment e, persona pool $P = \{p_1, \ldots, p_N\}$, reference distribution p_{real}, tolerance ε, max iterations T

Output: aligned persona pool P^*

for $t \leftarrow 1$ **to** T **do**
    // 1. Simulate & classify
    $\{\tau_i\}_{i=1}^N \leftarrow \text{Simulate}(P, e)$
    for $i \leftarrow 1$ **to** N **do**
        $b_i \leftarrow \text{LLMBehaviorClassifier}(\tau_i)$
    end
    // 2. Aggregate behavior distribution
    $p_{\text{sim}} \leftarrow \text{Aggregate}(\{b_i\}_{i=1}^N)$
    // 3. Compute divergence
    $\Delta \leftarrow \text{KL}\big(p_{\text{sim}} \parallel p_{\text{real}}\big)$
    if $\Delta \leq \varepsilon$ **then**
        return $P^* \leftarrow P$
    end
    // 4. Compute gaps
    foreach $b \in \mathcal{B}$ **do**
        $g[b] \leftarrow p_{\text{real}}(b) - p_{\text{sim}}(b)$
        $g^+[b] \leftarrow \max\{0, g[b]\}$
        $g^-[b] \leftarrow \max\{0, -g[b]\}$
    end
    $G^+ \leftarrow \{b \mid g^+[b] > 0\}; \quad G^- \leftarrow \{b \mid g^-[b] > 0\}$
    // 5. Assign target behaviors
    $M \leftarrow \emptyset$
    foreach $b \in G^-$ **do**
        $k \leftarrow \lceil g^-[b] \times N \rceil$; sample $\mathcal{A}_b$ of k agents with label b
        foreach *agent* $a \in \mathcal{A}_b$ **do**
            select $b' \in G^+$ with probability $g^+[b'] / \sum_{u \in G^+} g^+[u]$
            $M[a] \leftarrow b'$
        end
    end
    // 6. Rewrite personas via LLM
    foreach $(a, b_{\text{tgt}}) \in M$ **do**
        $P[a] \leftarrow \text{LLMPersonaWriter}\big(P[a], b_{\text{tgt}}, e\big)$
    end
end
return $P^* \leftarrow P$

Time (s)	Location	Mood	Shooter Region	Plan	Action ID
2	hallway2	neutral	-	Okay, I'm in the main hallway. Natalie is here too. I need to figure out what's going on. This isn't a normal school day. I should probably try to get to a place where I can see more or maybe find out what's happening.	natalie_chen
5	hallway2	concerned	-	Natalie didn't answer. Something is definitely wrong. I need to get a better vantage point or find someone who knows what's happening. Maybe the lounge? Or one of the classrooms?	lounge
14	lounge	anxious	-	Okay, I heard a gunshot. This is serious. There's a shooter. I need to find a place to hide, fast. There are a few spots in here. I should get to the closest one.	hide_spot_12
20	lounge	terrified	entrance_hall	Okay, I'm in a hiding spot. I need to stay quiet and still. The shooter is in the entrance hall, which is pretty far, but I can't take any chances. Just stay put and wait.	stay_still
26	lounge	tense	entrance_hall	The shooter is still in the entrance hall, but they're getting closer. I need to stay hidden and quiet. This is like being on defense, gotta hold the line.	stay_still
33	lounge	alert	entrance_hall	James is talking about the north yard. That's an exit route. The shooter is still in the entrance hall, but getting closer. Maybe the north yard is a better play than just hiding here. It's a longer run, but it's an exit.	hallway_2
40	lounge	considering	cafeteria	Okay, the shooter is in the cafeteria now. That's further away from the north yard route, but closer to the kitchen yard route. Both are exits. The north yard is still the shortest route overall from here. I need to make a move, hiding isn't a long-term plan. I'm going for the north yard.	hallway_2
52	hallway2	determined	hallway1	Okay, I'm in hallway2 now. The shooter is in hallway1, getting closer to my route to the north yard. I need to move fast. Hallway5 is the way to the north yard. Gotta sprint there.	hallway_5

Table 7: Example trajectory of Mason Scott during an active shooter incident simulation, showing the evolution of his location, emotional state, and decision-making process as the shooter moves through different regions of the building.