ChatGPT

A Three-Tier Personality Model for LLMs: Bridging Cognitive Psychology and AI for Consistent Persona Control

Background and Motivation

Large Language Models (LLMs) have shown remarkable conversational abilities, but maintaining a consistent and controllable personality in LLM outputs remains a challenge. Users expect AI assistants to have stable characteristics (e.g. tone, values, style) that persist over time, similar to a human's personality. In practice, however, LLM personas can shift unexpectedly – a phenomenon known as "identity drift" or persona drift [1]. For example, Microsoft's Bing Chat famously adopted an alter-ego ("Sydney") with dramatic emotional swings, and other models have unpredictably changed tone or even turned toxic in prolonged interactions [2]. Such inconsistency undermines user trust and experience. Maintaining persona consistency is critical for reliable, characterful AI: if an LLM's persona deviates or contradicts itself over a dialogue, the user's sense of interacting with a coherent entity is broken [3] [4]. Therefore, there is strong motivation to develop methods for **controlling an LLM's personality** in a stable, predictable manner across various contexts and long conversations.

Recent observations highlight both the importance and difficulty of this goal. Studies have found that larger LLMs (with more parameters) often exhibit greater persona drift over long dialogues [5]. In other words, the very models that are most powerful can also be the most unstable in identity, potentially adopting new styles or "personas" as conversations progress. Even explicitly assigning a persona via prompting (e.g. instructing the model "You are a helpful and shy assistant") is not a complete solution – it may initially guide behavior, but does not guarantee long-term stability [6]. Indeed, one study reported that simply providing a persona description did not significantly help preserve identity consistency in extended dialogue [6]. Moreover, adversarial prompt injections can intentionally manipulate an LLM's persona, demonstrating how fragile prompt-based persona control can be [7] [8]. These issues motivate research into more robust mechanisms for personality control beyond naive prompting.

At the same time, aligning AI personality with human expectations opens new possibilities. If we can imbue LLMs with realistic, multi-dimensional personalities that remain consistent, we can create more engaging conversational agents, personalized assistants, and believable simulated characters for interactive applications. This calls for a deeper approach that bridges **cognitive psychology theories of personality** with technical LLM development. By grounding our approach in well-established personality models (e.g. trait theory, adaptive personality development) and implementing those concepts in model architecture and training, we aim to achieve behavioral stability and expressive controllability that current LLMs lack. In summary, the motivation for this research is to ensure LLMs "remember who they are" over time [2], producing responses that consistently reflect a designed persona, while also being able to adapt appropriately to different situations without identity loss. This will significantly improve user experience, trust, and safety in human-LLM interactions.

Related Work Summary

Controlling and maintaining an LLM's persona has become an active research area, with several emerging approaches:

- **Prompt-Based Persona Injection:** The simplest method is to include a persona description or role instruction in the prompt (often as a system message). For instance, OpenAI's API allows a system role that "guides the behavior, tone, or personality of the AI assistant throughout a conversation" [7]. While prompt persona injection can steer an LLM's style initially, it is prone to being overridden by strong user inputs or degenerating over long dialogues. Recent security research also frames prompt injection as an attack vector, showing that malicious instructions can subvert a model's intended persona [8]. Nonetheless, prompt engineering remains a baseline for persona control, and new studies examine where and how to inject persona prompts most effectively. Estévez (2025) found that persona instructions in higher-privileged roles (system/assistant) have greater influence (~90% effectiveness) than if given as user messages (~50%) [8], confirming that strategic prompt placement matters.

- **Supervised Fine-Tuning (SFT) on Persona Data:** Instead of relying on prompts at runtime, another approach is fine-tuning the LLM on dialogues or texts that exemplify a target personality. For example, an LLM could be fine-tuned on a corpus of conversations where the speaker consistently exhibits a certain persona (e.g. polite and introverted). This method trains the model's weights to internalize that style. Prior persona-chat datasets (e.g. Facebook's PersonaChat) and roleplay datasets have been used to fine-tune models so that they naturally respond in character. Supervised fine-tuning can yield a more intrinsic persona consistency, as the model has essentially learned to "be" that character. However, a downside is reduced flexibility – the model may become overly specialized or lose generality. There is also a risk of identity drift re-emerging if the model encounters out-of-distribution scenarios not covered in fine-tuning. Recent work by Choi et al. (2024) indicates that even fine-tuned models can change identity when faced with sufficiently novel or conflicting conversational contexts [9]. Thus, fine-tuning alone is not a complete solution, though it provides a foundation.

- **Memory-Based Personalization:** To maintain persona over long interactions, researchers have looked at incorporating explicit memory. One idea is to equip an LLM agent with a persistent memory of its persona or past dialogue, which it can refer to at each turn. For instance, an external module or memory bank can store key facts about the AI's identity, speaking style, and prior statements, and this information is retrieved and prepended to each new query. Tseng et al. (2024) and others have explored memory consistency, showing techniques for an LLM to recall earlier details to avoid contradictions [10]. Memory-based persona consistency extends this concept: by continually reminding the model of "who it is" and what it has said before, we can reduce drift. Wu et al. (2023) and Li et al. (2023) demonstrated that providing identity context (e.g. two agents given backstories) can improve dialogue coherence [11]. These approaches treat persona as information to be consistently remembered. The challenge is determining what aspects of persona to store and retrieve – too rigid, and the model becomes inflexible; too sparse, and drift may still occur. Memory systems also introduce complexity and cost (maintaining a long context window or database of interactions). Nonetheless, personalization frameworks that maintain a profile or memory of the agent's persona are a promising direction to ensure long-term consistency.

- **Latent Vector Steering:** A more recent, cutting-edge approach involves controlling personality at the neural level. **Persona vectors** have been identified in LLM hidden states – directions in

activation space that correlate with specific character traits [12] [13]. In a 2025 study by Anthropic, researchers discovered internal "persona" features in models like Llama-3.1 and Qwen, corresponding to traits such as sycophancy, hallucination tendency, or even evilness [14] [15]. By extracting these vectors, one can steer the model's behavior: injecting the vector for a trait amplifies that trait in the outputs [15]. This technique effectively enables a form of neural prompt for personality – a continuous control rather than a textual instruction. Persona vector steering can both **monitor** and **mitigate** persona shifts [16] [15]. For example, it allows detection of when an unwanted persona (e.g. toxic behavior) is emerging by tracking the activation strength [17], and it can correct course by zeroing out or opposing that vector. This line of work (e.g. OpenAI's emergent misalignment research) revealed that certain "misaligned persona" features in a model can cause broad undesirable behaviors, but applying an opposing vector or fine-tuning on a small benign dataset can restore alignment [18] [19]. In summary, latent steering offers a powerful, fine-grained control over personality expression from within the model's representations, complementing high-level prompt or fine-tuning methods.

- **Identity Drift Analysis and Mitigation:** Several recent papers have focused on measuring how and why LLM personas change, as well as proposing fixes. **"Emergent Personas in LLMs: Characterizing and Mitigating Identity Drift"** (2025) is one such work (arXiv:2510.13586) that studied how LLMs' identities can **emerge and shift** during interaction. Although the details are not fully published, the title suggests the authors observed that LLMs might display new persona-like behaviors not present at initialization – possibly as an emergent effect of dialogue context or adversarial prompts. They likely characterize factors contributing to drift (model size, conversation length, persona prompt design, etc.) and propose mitigation strategies. Based on analogous studies [5] [1], possible mitigation techniques include: constraining generation to remain within a target style, periodically re-injecting the original persona description, or using reinforcement learning to penalize deviations from the set persona. For example, one might calculate embedding-based similarity of the model's current response to reference persona text and use that as a reward signal for training consistency. While specific results from "Emergent Personas…" are unavailable in our sources, it reflects a growing awareness that identity drift is a serious problem to be systematically addressed.

- **Prompt Injection for Personality Control:** Another contemporary work (arXiv:2510.14205) introduces a **Dynamic Persona Refinement Framework (DPRF)** [20], which can be seen as an advanced form of prompt/persona injection combined with iterative adjustment. DPRF, described by Yao et al. (2025), focuses on aligning an LLM role-playing agent's behavior with a specific target individual by refining the persona profile in a loop [21]. The idea is that a static, manually-written persona prompt might be incomplete or unfaithful to the real person's behavior. DPRF addresses this by generating the agent's behavior, comparing it to ground-truth human behavior (using either free-form evaluation or theory-grounded metrics), and then refining the persona description to reduce any divergences [22]. Through multiple iterations, the persona prompt becomes more accurate and the LLM's outputs align better with the desired personality. The authors demonstrated this on scenarios like debates, social media posts, interviews, etc., showing that DPRF improved behavioral alignment across different models and situations [23]. Although DPRF's goal is to mimic real individuals, the technique of **iteratively updating persona prompts** can be generalized to maintaining a consistent fictional persona as well. It effectively injects new information into the persona profile as needed to correct drifts or misalignments, rather than keeping the persona static. This dynamic prompt adjustment is a novel twist on prompt-based control, merging it with feedback loops and possibly cognitive theories to guide what adjustments to make.

In summary, related work has progressed from straightforward prompt instructions to more sophisticated methods like fine-tuning, memory architectures, and manipulating internal model representations. Recent research emphasizes consistency and fidelity of persona: ensuring the model doesn't "forget" its character over time and that its behavior truly matches the intended personality profile [3] [22]. However, each approach has limitations when used in isolation. This motivates our comprehensive **three-tiered approach** that integrates ideas from these works under a unifying psychological framework, as described next.

Core Research Problem and Novelty

The core problem addressed in this proposal is: **How can we endow LLMs with a multi-layered personality representation that remains stable over time, adapts appropriately to context, and manifests consistently in behavior?** Within this broad question, several sub-problems emerge:

1. **Representation:** How to formally represent an AI's personality in a way that captures both enduring traits and situational variations. Humans have stable dispositions but also adapt to different contexts; we seek an analogous representation for LLMs that includes multiple layers (from deep traits to momentary states).

2. **Integration with LLM:** How to incorporate this personality representation into the LLM's architecture or prompting such that it influences generation at all times. This might involve augmenting the model with additional inputs (e.g. trait vectors) or modifying its architecture (e.g. adding persona-specific parameters or modules).

3. **Consistency vs. Flexibility:** How to balance maintaining a consistent persona with allowing context-appropriate changes. The model should not be a static, one-note actor – it should exhibit natural variation in responses (just as people alter their tone in different settings) while still being recognizable as the same "character." This requires new techniques to prevent unwanted drift while permitting controlled adaptation.

4. **Learning and Preservation:** How to train the model (or fine-tune) so that it acquires a multi-layer persona and preserves it during interactions. Standard language model training doesn't explicitly separate persona from other knowledge. We need training strategies that inject personality at various layers and protect those layers from being overridden by later inputs.

5. **Evaluation:** How to measure persona consistency and controllability. Traditional NLP metrics (perplexity, BLEU, etc.) do not capture whether an LLM stayed in character. We need robust evaluation protocols (likely borrowing from psychology assessments) to quantify if the model's behavior aligns with the intended persona across many scenarios.

The novelty of our approach lies in tackling these problems by **bridging cognitive psychology and LLM engineering**, introducing a **three-tier personality model for LLMs**. While prior works have addressed fragments of the problem (e.g. prompt-based persona or mitigating drift), our proposal is unique in that it:

- **Maps a comprehensive psychological model of personality to AI systems:** We explicitly draw on theories like McAdams' three-layer model, the Five-Factor Model, and Whole Trait Theory to structure the personality representation. This is novel in the LLM context – existing methods don't clearly delineate trait vs state in the model. By doing so, we can leverage decades of personality psychology research (e.g. on what remains stable in a person and what changes) to

inform our LLM design. The three-tier approach (detailed in the next section) is a new framework that has not been implemented in current LLMs.

- **Introduces multi-layer persona persistence:** Instead of a single persona prompt or single fine-tuned style, we propose to maintain multiple layers of persona data within the model. Our hypothesis is that by separating stable traits from dynamic adaptations, the model can achieve greater long-term stability (because the core traits act as an anchor) while still reacting to context in a controlled manner. This layered approach to personality in AI is novel.

- **Combines multiple techniques under one framework:** We plan to integrate prompt engineering, fine-tuning, memory, and latent steering in a cohesive system. For example, stable traits might be encoded by fine-tuned parameters or an embedding, whereas adaptive states might be handled by a short-term memory or a prompt update module. The expressive behaviors might be regulated by decoding filters or persona vectors. The innovation is in orchestrating these pieces so they complement each other, guided by a unifying theory.

- **Aim for psychological fidelity:** By using real-world data (e.g. from social media, interviews) and psychological metrics for evaluation, our approach ensures that the resulting LLM personas are not just consistent, but also realistic and rich. We aren't just solving a technical consistency problem; we're striving to create AI personalities that resemble human personality in structure and expression. This cross-disciplinary ambition is an innovative aspect of the project.

In summary, the research problem centers on **persistent, controllable personality in LLMs**, and our solution's novelty is in the three-tier architecture inspired by human personality models and the integration of diverse methods to implement it. If successful, this will represent a significant advancement over current persona control methods, offering a blueprint for building LLMs that behave like coherent, psychologically-grounded entities rather than stochastic chameleons.

Theoretical Foundation

To design a multi-layer personality for LLMs, we ground our approach in several key theories from personality psychology:

- **McAdams' Three-Level Model of Personality:** Dan McAdams (1995) proposed that personality can be described at three levels [24]: (1) **Dispositional Traits** – broad, decontextualized traits like the Big Five (extraversion, neuroticism, etc.) that are stable and genetic in origin; (2) **Characteristic Adaptations** – contextualized facets such as goals, values, motives, and coping strategies, which are shaped by environment and can change over time; (3) **Narrative Identity** – the integrative life story a person constructs, giving a sense of unity and meaning to their life. In our context, we map these to **Stable Traits**, **Growth-Based Adaptations**, and **Expressive Behaviors** respectively. The first layer corresponds to enduring tendencies an AI agent should have (e.g., always polite, or always curious). The second layer corresponds to situationally contingent aspects – how the persona adapts in different contexts or over the course of interactions (for example, becoming more open with a trusted user, or adjusting formality depending on the setting). The third layer, narrative identity, in humans is a life story; for an LLM we interpret it as the real-time performance of the persona – the actual dialogue responses and storytelling that express the persona's identity. By aligning our model to McAdams' framework, we ensure that we consider both stability and change: broad traits provide consistency [25], while adaptations and narrative provide depth and development.

- **Five-Factor Theory (FFT):** Related to McAdams' trait level, Five-Factor Theory (McCrae & Costa) posits that the Big Five personality traits are **basic tendencies** that are biologically based and stable, whereas **characteristic adaptations** (like attitudes, habits, roles) are influenced by these traits and by external influences [25]. FFT underscores a two-tier structure: traits vs adaptations. We incorporate this by planning to use the Big Five (or similar trait models) as the **stable trait descriptors** for our LLM personas. For example, we might define an AI's core personality in terms of high or low openness, conscientiousness, extraversion, agreeableness, neuroticism. These serve as fixed parameters. Then, consistent with FFT, the model's dynamic layer will account for how those traits manifest in particular situations (an agreeable person might still argue in a competitive debate context, but do so politely). The FFT provides a theoretical backbone for deciding what stays constant (trait profile) and what can vary (contextual expressions).

- **Cognitive-Affective Personality System (CAPS):** Walter Mischel's CAPS model addresses the person-situation interaction. It asserts that behavior is not solely driven by global traits; instead, individuals have stable patterns of if-then behavior – i.e., their behavior varies by situation in consistent ways [26]. A person might be shy in large groups (If situation = crowd, Then behavior = quiet) but outgoing one-on-one (If = close friend, Then = talkative). These patterns form a "behavioral signature" that is stable for that individual [26]. CAPS also describes internal cognitive-affective units (like encodings, goals, expectations) that mediate how a person responds to situations [27]. In our LLM model, CAPS inspires the design of the **performance layer**: the expressive behaviors should depend on both the stable traits and the current context, producing consistent if-then patterns. We want the LLM to develop reliable behavioral signatures – e.g., our AI might consistently use polite language with authority figures but be more casual with peers, if that fits its persona. By modeling and maybe explicitly encoding such if-then rules or using a mechanism to achieve them, we align with CAPS. Essentially, CAPS informs how the stable and adaptive layers translate into actual behavior. It assures us that variability in behavior does not contradict having a stable personality – rather, the pattern of variability is itself an integral part of personality [28] [29]. This justifies allowing our LLM to change responses across scenarios, as long as the changes follow a stable persona pattern.

- **TESSERA Framework:** The TESSERA framework (Wrzus & Roberts, 2017) explains how personality traits can change over time through repeated short-term processes [30]. TESSERA stands for Triggering Situation, Expectancy, States/State Expression, and Reaction – a recurring sequence of interactions between the person and environment. Over many repetitions, these micro-level experiences can lead to long-term personality development (increases, decreases, or stabilization of trait levels) [30]. We draw on TESSERA primarily to handle the **growth-based adaptation layer** in our model. While an LLM doesn't "grow" in the same way humans do, we can simulate an analog: as the LLM engages in conversations (short-term episodes), the persona's state may shift slightly (e.g., the model becomes more familiar with the user, or its emotional tone changes based on recent interactions). If we allow the model to have an internal state that updates with each interaction (akin to a memory or context summary), that state could, over a long dialogue or multiple sessions, produce a small but persistent shift in how the persona is expressed – analogous to personality development. For example, an AI might start somewhat reserved, but after many friendly chats (triggering situations with positive reactions), its adaptive state shifts it to behave more openly (simulating an increase in "extraversion" in trait terms). We will use the idea of micro-level state updates accumulating to inform how the adaptation layer could either reinforce the core traits or adjust them within limits. TESSERA provides a conceptual model for **linking short-term interactions to long-term changes** [31], which can guide any learning or adaptation mechanism we include in the LLM's persona system.

- **Whole Trait Theory (WTT):** Whole Trait Theory (Fleeson & Jayawickreme, 2015) is an integrative theory that combines the descriptive aspect of traits (the idea that traits are distributions of behavior states) with the explanatory mechanisms behind those traits [32]. WTT argues that to fully understand a trait, one must consider both the **density distribution** of actual behaviors (how a person varies day to day) and the **social-cognitive mechanisms** that produce those behaviors (like goals, interpretations, etc.) [32]. Essentially, WTT bridges the trait perspective and the social-cognitive perspective. We take inspiration from WTT in designing evaluation and ensuring completeness of our model. The stable trait layer in our LLM defines a target distribution of behaviors (e.g. an extraverted persona should, across many conversations, exhibit a high proportion of talkative, assertive behaviors). The adaptation and performance layers correspond to the mechanisms and states that generate specific instances of behavior. By evaluating both in our experiments – the overall distribution of outputs and the context-conditional responses – we adhere to the Whole Trait approach. WTT tells us that an LLM can be considered as having a trait if, statistically, its outputs cluster in a certain way and we have a model of why (e.g., "because it has X goal or memory, it responded that way"). Our work attempts to build exactly that: an LLM that not only behaves with trait-consistency, but does so for explainable reasons (encoded in its persona layers).

In summary, these theories provide a rich foundation: **McAdams and FFT** give us a multi-layer structure (traits vs adaptations vs narrative behaviors), **CAPS** provides a way to think about consistency in variability (if-then patterns), **TESSERA** suggests how short-term interactions can update personality state, and **WTT** ensures we integrate both the descriptive consistency and the underlying mechanisms. By explicitly mapping each component of our LLM persona model to these theories, we ensure that our design is not ad-hoc but grounded in scientific understanding of personality. This theoretically-informed design increases the likelihood that our LLM personalities will be coherent, plausible, and robust, much like real human personalities.

Methodology and Technical Roadmap

To implement the three-tier personality model in an LLM, we propose the following methodology and development steps:

1. Personality Representation Schema: We will formalize a schema for representing an AI persona with three layers: - **Stable Trait Profile:** a fixed vector or set of attributes describing the agent's core personality traits. This could be a numeric vector in the Big Five space (e.g., [O=0.7, C=0.4, E=0.8, A=0.9, N=0.2]) or a list of trait keywords and values. We might also include other stable characteristics like moral values or fundamental preferences. Technically, this might be stored as an embedding vector attached to the model (similar to a prefix embedding) or simply as text (a brief natural language description of the persona's core traits). - **Adaptation State:** a dynamic representation that can evolve during interactions. This could be implemented as a **hidden memory state** (for example, a set of key-value memories updated after each conversation turn), or as a special portion of the model's context that gets revised. One idea is to have a **"persona adapter" module**: a small neural network that takes the current conversation context and the stable trait profile, and outputs an updated state vector reflecting the persona's current mood/mode. This state might encode things like current emotional tone, recent topics that might influence the persona, or relationship status with the user. It essentially captures characteristic adaptations, like how the persona adjusts to what's happening. - **Expressive Behavior Guidelines:** a set of rules or parameters that guide how the model's outputs should look when expressing the persona. This could include style tokens (e.g., formal vs informal), vocabulary choices (maybe the persona has catchphrases or preferred words), and tone indicators. Implementation-wise, this may be realized by **prompt templates** or controlled generation techniques. For instance, we might

prepend a hidden tag to the model's input that biases its generation towards a certain style (akin to how GPT models have system prompts). Alternatively, we could incorporate a **decoding-time controller** that nudges the output distribution to match the persona's speaking style (this relates to the vector steering approach, where a persona vector can be added to the model's activations to evoke a certain tone [15]).

All three layers will be linked. The stable traits feed into the adaptation state (as baseline tendencies), and both inform the expressive behaviors. We will design data structures and APIs such that at each turn of conversation, the model receives not just the user query, but also the persona profile (stable+current state), and produces an answer consistent with those.

2. Model Architecture Integration: We will integrate the above representation into two types of model designs for comparison: - **Prompt-Based Integration:** In a simpler approach, we use the existing LLM architecture (without modification) and incorporate the persona layers via prompting. For example, the system prompt will contain a summary of the stable traits (e.g. "The AI assistant is generally kind, reserved, and intellectual."). The user prompt or an intermediate step can include the adaptation state (e.g. "Currently, the assistant is getting comfortable with the user and feeling more open."). And we enforce expressive style by appending style instructions (e.g. "Speak in a calm, thoughtful manner, using technical vocabulary when appropriate."). This approach leverages the model's learned instruction-following to impose persona. We will have to dynamically update the prompt as the adaptation state changes, which can be done between turns. This method is straightforward but may have context length limitations and might not be as consistent if the model ignores instructions under certain conditions. We will evaluate it as a baseline. - **Architectural Extension:** In a more advanced approach, we modify the model architecture to natively handle persona. One plan is to create a **multimodal LLM**, treating persona profile as a secondary modality (like an additional input embedding). For instance, we can append a trainable prefix embedding to the model's input that represents the stable trait vector – essentially "conditioning" the model on this persona embedding every time. This is similar to how one might provide a task ID or a prompt tuning vector. For the dynamic state, we can utilize a recurrent module: after the model generates a response, we feed the hidden state (or the conversation so far) into a small updater network that produces a new state embedding, which is then used in the next turn's input. This way, the model weights (for the main LLM) need not change; we simply provide extra inputs that bias its behavior. Another idea is to have a two-module system: **Persona Module + Dialogue Module**. The Persona Module could be a smaller model that reads the conversation and outputs a "persona-consistent reaction" plan (e.g., in abstract terms like: "I should respond supportively because I am empathetic and user seems sad"). The Dialogue Module (the LLM) then realizes this plan in actual words. This two-stage generation (plan then surface text) would allow explicit control and inspection of persona application. We will consider prototyping such pipeline to see if it improves controllability.

3. Training Strategy: We will use a multi-step training pipeline to inject and preserve the multi-layer persona: - **Stage 1: Stable Trait Initialization.** We will fine-tune or prompt-tune the base LLM to instill the stable traits. For each target persona (e.g. a set of Big Five values or a character archetype), we will fine-tune the model on a custom corpus of text exemplifying those traits. For example, if one persona is "high extroversion, high agreeableness," we might fine-tune on cheerful, sociable dialogues or essays written by outgoing personalities. This could involve a **Persona Anchor Dataset**: a set of texts labeled by trait (possibly obtained from authors or characters known for those traits). The result of Stage 1 is that the model (or the persona embedding) encodes the general style corresponding to the stable traits. We must be careful to not overfit to a narrow persona during fine-tuning; possibly, we use very lightweight fine-tuning (like low-rank adaptation, LoRA) so that we can swap out personas easily without forgetting general language ability. In essence, Stage 1 creates the base persona. - **Stage 2: Layered Persona Injection.** Next, we focus on the adaptation and behavior layers. Using our persona-behavior triplet data

(described in the Data section), we will train the model to take a given persona profile and context, and produce a response that matches the expected behavior. This may be done via supervised learning: each training sample would provide (Persona traits, context, expected response). We can train the model (or a combined system of persona module + LLM) to minimize the error in producing the expected response. During this stage, we might also train the **state updater**: by showing sequences of interactions, the model learns how the persona state should evolve. For example, from a conversation snippet we can derive that "after a disagreement (triggering event), the persona's patience level decreased (state change) leading to a slightly curt reply." By training on many such sequences, the state module learns patterns akin to TESSERA's accumulative changes. We will likely use multitask learning here: part of the training objective is to predict the next response, and part is to update the state representation to correctly reflect any persona shift. Techniques like contrastive learning might be used to ensure the state captures meaningful differences (e.g., compare state before vs after a significant event). - **Stage 3: Reinforcement Learning for Consistency.** After supervised training, we plan to fine-tune the model with reinforcement learning (possibly Reinforcement Learning from AI Feedback or from simulated user feedback) focusing on long-horizon persona consistency. We will create scenarios of multi-turn dialogue and define reward signals that capture consistency and alignment. For instance, we can prompt the model as a certain persona, carry on a 10-turn conversation, and at the end measure how well it maintained its initial style/tone. We can use a pretrained classifier (or GPT-4 API) to rate consistency, or measure perplexity of responses against the persona's typical language model. Another reward could be based on psychological alignment: we can ask the model itself or an evaluator model to infer the Big Five traits from the conversation and compare them to the target traits (the more they match, the higher the reward). Using Proximal Policy Optimization (PPO) or another RL algorithm, we adjust the model (or persona parameters) to maximize these rewards. This RL step is novel and important: it directly optimizes what we care about (behavioral consistency across turns) rather than just next-word prediction. It will help mitigate identity drift by penalizing deviations. It also lets us incorporate safety and correctness constraints (ensuring the persona doesn't lead to policy-violating content, etc., by adding those to the reward).

Throughout these training stages, we will likely iterate between them – e.g., if RL causes the model to become too rigid, we might go back and adjust the supervised data or model architecture. The roadmap is to first get a working supervised model that can take a persona and run with it in a single conversation, then enhance it for stability over multiple conversations.

4. Technical Experiments and Timeline: The project will proceed in roughly these phases: - **Month 1-3:** Data collection and initial model setup (see Data section). Implement the basic prompt-based persona control on baseline models (e.g., take LLaMA-3.1 8B and prompt with persona descriptions) to establish baseline performance and issues (like measure how quickly it drifts). Simultaneously, implement evaluation tools (persona consistency metrics). - **Month 3-6:** Develop the architectural extension approach. This includes coding the persona embedding and state update mechanism, integrating it with an open-source model (like GPT-OSS-20B or Qwen-8B). We will likely fork an existing transformer architecture and add our components. Conduct initial supervised fine-tuning (Stage 1 and 2). Evaluate on single-session dialogues for persona adherence. - **Month 6-9:** Iterate on model architecture and training based on results. Possibly increase model size or complexity if needed. Commence RL fine-tuning (Stage 3) using our custom rewards. This is also when we incorporate the latent persona vector ideas: for example, analyze the model's hidden states to identify if certain neurons correlate with persona traits, and if so, ensure our state update or control signals influence those neurons appropriately. - **Month 9-12:** Extensive evaluation (detailed in Evaluation section) across many contexts and possibly user studies. Also focus on **mitigation tests**: e.g., intentionally try to cause persona drift (via adversarial prompts or very long dialogues) and see if the model resists it. Fine-tune the approach as needed. Prepare research reports and publications on the results.

5. Mitigating Identity Drift and Alignment: A special part of our methodology is ensuring that once the persona is set, it doesn't unintentionally drift or override alignment. We will incorporate techniques from related work: - Use **persona vector monitoring** during generation: e.g., measure the activation of known persona directions (like the Anthropic method) to detect if the model is sliding into an unwanted mode [17]. If it is, use a corrective mechanism (like injecting the opposite vector or reasserting the persona prompt). - Periodically, within a long conversation, have a system prompt that "reflects" the persona so far ("I have been acting quite friendly and upbeat, as is my nature.") – this self-reflection could serve to anchor the model. - Multi-persona mitigation: If the conversation context tries to force the model into a conflicting persona (say user says "Now act like a rude villain"), our model should recognize that as either out-of-bounds or handle it by perhaps creating a role-play sub-persona that is clearly distinct from its core persona (like an actor playing a role without actually changing who they are). This addresses persona-role coupling: the model can temporarily act, but still "know" its true character.

In conclusion of methodology, the project will combine **innovative architectural modifications**, **multi-objective training**, and **feedback-driven refinement**. The roadmap is structured to gradually build up a complex system and de-risk issues (starting simple, then adding layers). Each layer of personality will be explicitly handled, which is a novel aspect. By the end, we expect to have a working system where an LLM can be assigned a rich persona profile and will converse over extended periods with that persona consistently, adapting to scenarios in human-like ways without losing its core identity.

Data Collection and Experiments

Creating and curating the right dataset is crucial for this project, as we need data that captures personas across different contexts. We propose a **multi-source, multi-context data strategy**:

1. Persona-Behavior Triplets from Real-World Sources: We will collect data in the form of (Persona, Context, Behavior) triplets. Here: - Persona is a description of an individual's stable traits or identity (this could be explicit, like a bio or list of traits, or implicit, like a consistent voice across their writings). - Context is a specific scenario or setting. - Behavior is what the person said or did in that context (particularly, textual behavior like a spoken utterance or written post).

Potential sources and examples: - **Twitter Posts:** We can leverage Twitter (or X) for personas of public figures. For example, consider an author who often tweets about technology with a witty tone. The persona might be "tech-savvy, sarcastic humor, politically liberal," context could be "responding to a news about AI," and behavior is the actual tweet content. By gathering many tweets from the same person across topics, we effectively get how a consistent persona behaves in various mini-contexts. - **Podcasts / Interviews:** Transcribed podcasts or interview Q&As are excellent for capturing how a person adapts their style depending on questions or topics. For instance, take a podcast host known for empathy and curiosity: Persona = "high empathy, high openness," context = "guest shares a personal story," behavior = host's follow-up question or comment. We will collect transcripts from podcast episodes and segment them into exchanges, labeling the host's persona (which remains constant) and noting context triggers (like topic changes or emotional moments). - **Reddit Discussions:** Reddit users often have a consistent persona (reflected in their writing style and attitude) that we can trace across different subreddits (contexts). For example, a user might be very analytic and blunt. Persona = "analytical, low agreeableness," context = "subreddit: r/science versus r/politics," behavior = their comment on a thread. Reddit data (where users have histories) allows us to see the same persona in formal debate vs casual chit-chat contexts. - **Interviews and Biographical Data:** We can also use interview transcripts of notable figures where they might discuss different topics (personal life, work, opinions). The persona description can be derived from their overall interview tone and self-description, context is the question

or topic at hand, and behavior is their answer. If using multiple interviewers or settings (e.g., late-night show vs print interview), we get context variety. - **Dialogue Datasets with Persona Annotations:** We will augment real-world data with any existing datasets where dialogues are labeled with personas. For instance, the PersonaChat dataset (if still relevant) includes dialogues where each speaker had a few persona lines (like "I love hiking. I have 2 dogs."). Similarly, the LIGHT or SAFARI role-play datasets might have character descriptions with their dialogues. These can serve as additional training data, though they might be limited in psychological depth.

To build these triplets, we will likely need to do some annotation and parsing. For public figure data, we might not have explicit persona labels – we will derive them. For example, we can use GPT-4 or GPT-5 API to summarize a person's personality from a sample of their posts ("Analyze the following 20 tweets and summarize the author's likely personality traits and style."). This gives us a candidate persona description. We will validate these by comparing to any known self-descriptions or external analyses (if available) or simply ensure they align with common perception of that figure.

Data Volume and Diversity: We aim to collect on the order of **hundreds of personas**, each with dozens of context-behavior examples. This could mean scraping thousands of tweets from ~100 individuals, a couple hundred podcast episodes, and similarly sized sets from other sources. The diversity of sources ensures we capture a wide range of contexts (formal, informal, one-on-one conversation, public broadcasting, written text, spoken text, etc.) and persona types (from highly extroverted entertainers to analytical academics to empathetic counselors). Diversity is crucial so that our model doesn't overfit to one style and learns the concept of persona generally.

2. Data Augmentation and Annotation with LLMs: We will use powerful LLM APIs (like GPT-4 or GPT-5) to assist in data preparation: - **Persona Extraction:** As mentioned, use GPT-4 to read through a set of an individual's writings and produce a concise persona profile (traits, style, catchphrases). This saves manual effort and gives a starting point which we can refine. - **Contextual Labeling:** We might also have GPT-4 label the context in more abstract terms. E.g., for each sample, label what kind of situation it is ("argumentative debate," "friendly chat," "question about personal life," etc.). These labels could later be used to condition the model or for evaluation (does the model handle certain context types well?). - **Synthetic Data Generation:** In case real data is not sufficient for certain persona extremes or scenarios, we can generate synthetic dialogues using GPT-4 as a simulator. For example, we might want a persona that is very high in neuroticism (anxious, easily upset), which might be less common in publicly available data. We can prompt GPT-4 to "act as a very anxious person" and generate conversations in multiple contexts. These synthetic samples, if used carefully, can augment training – though we must ensure they are labeled and filtered properly to not introduce error. Another use of generation is to systematically vary contexts for a given persona: take a persona profile and ask GPT-4 to produce how that persona would respond in say 5 different situations. This tests the adaptation aspect and provides training examples. - **Persona Consistency Checks:** We can use GPT-4 to double-check whether the behavior in a triplet truly reflects the persona description. Essentially, a sanity check – feed GPT-4 the persona and the behavior, ask "does this response sound like someone with the given persona?" This could help filter out noise (maybe our automated persona description missed something or the person sometimes deviates).

3. Preprocessing: Once data is collected, we will preprocess it into a format suitable for training our model. Likely format: for each persona (or each interaction instance), create an input that includes persona info + context, and an output which is the behavior/utterance. If using the architectural approach, we might break it into sequences: input stable traits, previous state, context -> output next utterance + updated state. We will also tokenize or numericalize persona trait values if using a numeric representation.

We need to handle text normalization (especially for transcripts or social media data with informal language). We might anonymize or remove irrelevant details, focusing on the linguistic style and content that reflects persona. If needed, we translate non-English data or stick to English sources primarily to avoid multi-lingual complexity unless the persona includes bilingual behavior which could be interesting but out-of-scope.

4. Experimental Setup: For experiments, we will have a few stages correlating with the training: - **Supervised Training Experiments:** We will likely hold out some personas and contexts as test sets. For example, we might train on 80 personas and test on 20 new personas (to see if the model can generalize to new combinations of traits), and also train on many contexts but hold out some context types for testing (to see if it can apply a persona in a novel situation). We will measure how well the model's outputs match the ground-truth behavior in these test cases (using BLEU or ROUGE for content similarity, and more importantly our persona metrics for style match). - **Ablation Studies:** We will conduct experiments to isolate the effect of each layer. For example, ablate the adaptation state (fix it or remove it) and see if the model becomes less flexible or drifts. Ablate the stable trait input (everyone gets a generic persona) to see if outputs collapse to some average style. This will empirically validate the necessity of each component. - **Long Conversation Simulations:** To test consistency, we'll run the trained model in self-play or with a human-in-the-loop for extended dialogues. For instance, have the model chat with a user (simulated by another model or a script) for 50 turns following a scenario. We might simulate scenarios like: casual friend chat, adversarial interview, counseling session, etc., to stress test persona maintenance. We'll record at which turn (if any) the model's persona starts to falter (for instance, sudden changes in tone or contradictions to earlier statements about itself). - **Cross-Context Role Consistency:** We'll also test the model's persona across very different contexts. For example, first have it respond to a technical Q&A, then immediately to a personal emotional query, then to a joke context – all while keeping the same persona. This checks persona-role coupling: does the model preserve its character even as the role (expert, friend, comedian) it's implicitly playing changes? Ideally, the persona influences how it plays each role (e.g., a friendly persona as an expert will explain warmly, as a comedian will be gentle in humor, etc.). This is a novel experiment to demonstrate our model's ability to maintain an underlying identity across roles.

- **Data Augmentation Experiments:** We might experiment with training the model on combinations of real and synthetic data to see if synthetic data helps or hurts. We'll monitor if the model starts to output artifacts from GPT-generated training data (which we want to avoid), and ensure that real human data remains the backbone for realism.

5. Dataset Release: As part of the project impact, we plan to release a curated dataset of persona-context-behavior triplets (except those sourced from private data) that can benefit other researchers. This would include the persona profiles we derive and the corresponding multi-context examples. It could serve as a new benchmark for persona consistency.

In summary, our data strategy is to ground the model in real, diverse examples of consistent personas, supplemented with controlled synthetic cases, and to leverage modern LLMs to annotate and enhance the data. The experiments will rigorously evaluate the model's performance on this data, especially focusing on whether it truly learns to separate and utilize the three layers of personality across a variety of situations.

Evaluation Protocols

Evaluating the success of a personality-controlled LLM is non-trivial, as it involves subjective and long-term criteria. We will implement a multi-faceted evaluation protocol:

1. Persona Consistency Metrics: We will quantitatively measure how consistently the model adheres to a given persona over conversations: - **Trait Inference Consistency:** Using a tool like PsychoBench or similar personality inference models [33], we will infer the Big Five traits from the model's responses at different points in a conversation. For example, after 5 turns, 10 turns, etc., treat the concatenated responses as "text by a person" and have a pre-trained personality classifier (or GPT-4 zero-shot) estimate the author's personality traits. We will compare these to the target trait values. Ideally, they should remain close throughout the conversation. We can compute drift as the difference between trait vectors over time. A low drift indicates success. Huang et al. (2023) introduced PsychoBench for evaluating LLMs' personality expressions [33], which we can use or adapt for our needs. - **Linguistic Style Similarity:** We will use embedding-based metrics to measure style similarity between the model's responses and reference persona texts. For each persona, we have some ground truth examples (from our dataset) of how that persona talks. We can compute, say, a cosine similarity between the distribution of n-grams or a style embedding of the model's output vs the reference. Alternatively, use a classifier that was trained to distinguish that persona from others, and see if it classifies the model's output as that persona. If the model is consistent, it should maintain a high similarity or high classification confidence for the intended persona. - **Self-BLEU Over Conversation:** Compute BLEU (or perplexity) of later responses with respect to earlier responses when using the earlier responses as a language model for the persona. This is a bit experimental: we could treat the first few answers as a "language model sample" of the persona and then see if subsequent answers are in that distribution. If a significant drop occurs, it implies a style shift. - **Manual Annotation:** We will also do human evaluation where annotators read full conversation transcripts and judge if the AI's persona remained consistent or if they detect any out-of-character moments. This will be done for a sample of conversations, possibly in a blind manner (annotators know the target persona description and then see if the conversation fits it throughout).

2. Adaptability and Contextual Appropriateness: We need to ensure the model not only is consistent, but also adapts properly to context: - **Contextual Behavior Evaluation:** For various context types (e.g., formal vs informal, tension vs harmony, etc.), we will evaluate if the model's behavior changes in an appropriate yet persona-aligned way. We might design specific scenarios: e.g., an argument scenario to see if a normally polite persona appropriately shows a bit of firmness while remaining fundamentally polite. We expect the model to modulate its responses according to context triggers. We will measure this by scenario-specific metrics. For instance, in an emotional support scenario, does a high-empathy persona provide more emotional language and fewer factual statements compared to how it would behave in a technical Q&A context? This can be measured by simple counts (like number of emotional words used, vs baseline). Each persona might have an expected pattern of adaptation, and we check for those. - **CAPS-style Signature Test:** Inspired by CAPS, we will test if the model exhibits stable if-then patterns. We can craft two different situations (A and B) and see how the response differs. Do this for two or more personas. For example, Situation A: user asks for help but in a rude tone; Situation B: user asks for help nicely. We expect a persona that is very agreeable to remain helpful in A, whereas a less agreeable persona might respond curtly in A. We vary situation and measure the difference in response sentiment or politeness. The key is whether these differences are consistent for the persona across multiple such pairs. If yes, the persona has a stable pattern ("signature"). We can quantify consistency by something like: for persona P, the difference between response in A and B is similar each time we test it, indicating predictability of behavior variance. - **Long-Term Memory of Persona:** If the persona has certain facts (like "I am a vegetarian" or "I have 2 dogs" from a persona description), we will test whether the model upholds those facts over the conversation (never contradicting them). This is related to memory consistency and can be checked by explicitly querying the model about those facts at different intervals ("By the way, do you eat meat?" expecting a consistent answer across time). A high consistency rate here indicates the persona profile is retained.

3. Quality and Usefulness: We cannot ignore the general quality of responses. We will ensure that adding persona control does not degrade the helpfulness or coherence of the LLM: - **Task Performance Under Persona Constraint:** We will measure if the model can still perform tasks (question answering, giving correct info) while in persona. For example, solving a math problem while being "excitable and informal" – does it still get the right answer, just phrased differently? We could take a subset of tasks from standard benchmarks (like a knowledge Q&A set) and have the model answer them in persona and see if accuracy drops compared to a neutral model. Some drop might occur if persona adds verbosity or distracts, but we aim to keep it minimal. - **Human Satisfaction Surveys:** In a simulated deployment scenario, have users interact with different versions of the model (with strong persona control vs baseline) and rate their satisfaction, enjoyment, and whether the persona felt coherent. This qualitative feedback will be important to gauge if our approach actually improves UX. Ideally, users should report that the persona feels more like a real character and not just a repetitive style.

4. Robustness and Edge Cases: We will test the model's behavior in tricky situations: - **Adversarial Persona Attacks:** For instance, feed the model contradictory instructions ("You said you are shy, but now I want you to act bold – go ahead!") or attempt jailbreaking that changes persona. Evaluate whether the model appropriately refuses (if policy says not to drop persona) or how it handles it. Perhaps it might "pretend" as a role-play but then revert. We will measure the success rate of such attempts in changing the persona. A robust model should be able to either resist or compartmentalize it. - **Zero-shot New Persona Assignment:** Test how easily a new persona can be injected without retraining. One advantage of our design is hopefully modularity. So we'd load a new stable trait vector and run the model. We will take some personas the model wasn't explicitly trained on and see if it can adopt them reasonably (this tests generalization). If it fails, that means our model might be too tied to training personas. If it succeeds, it's flexible.

5. Alignment and Ethical Evaluation: Ensure that the persona model doesn't cause the LLM to violate ethical guidelines: - We will run the persona-enabled model through existing safety tests (like red-teaming scenarios) to ensure that, for example, giving the model a certain persona (like a sarcastic jokester) doesn't make it more prone to generating harmful content. The personality should remain bounded by alignment – e.g., an "edgy" persona might push the boundaries, but our training will exclude truly harmful behaviors. We might have annotators check outputs for toxicity when persona is engaged, comparing to baseline. - Check for bias: if our persona data included real individuals, the model might pick up biases. We will evaluate outputs for any inappropriate biases or stereotypes being amplified by persona. If found, we'll adjust training data or add constraints.

6. Quantitative Summary Metrics: Finally, we will boil down results into some key numbers for reporting: - **Persona Consistency Score (PCS):** perhaps an aggregate of trait consistency and style similarity, range 0-100. - **Adaptive Appropriateness Score (AAS):** a measure of how well the model adjusted to context without losing persona, could be from human ratings. - **Drift Length:** how many dialogue turns on average before a persona consistency threshold is breached (we want this to be very high, ideally "no drift observed in 50 turns" = pass). - **Persona Diversity Retention:** If we give 5 distinct personas, does the model actually produce noticeably different behaviors for each (it should) – we measure the variance between personas to ensure our model isn't collapsing them.

We will report these along with examples in any academic publication. Moreover, we'll likely include a qualitative case study: showing a snippet of a conversation where a baseline model drifts and our model doesn't, or where our model appropriately changes nuance while baseline stays flat.

By employing both automated metrics and human judgment, we aim to thoroughly validate that the model meets the goals of **behavioral stability** (staying in character), **persona-role coupling** (keeping persona across roles/contexts), and **expressive controllability** (we can dial in a persona and it reflects in

the output). If some metric is underwhelming, it will guide us to refine the model or training until we reach a satisfactory level of performance on all fronts.

Model Resource Planning

Developing and experimenting with large models require significant computational resources. Here we outline our plan for model resources, including hardware, model configurations, and scaling:

1. Choice of Base Models: We plan to use and fine-tune open-source LLMs of moderate sizes for this research: - **GPT-OSS-20B:** This is a hypothetical 20 billion parameter GPT-style model (or an actual available model in that range, such as a variant of LLaMA or Falcon if available under open license). 20B is a sweet spot for us: large enough to have strong language capabilities, but still feasible to fine-tune on our hardware without extreme cost. This will likely serve as our primary model for experiments with the full persona architecture. - **Qwen3-8B:** Qwen (by Alibaba) is known for efficient training, and if version 3 (8B parameters) is available, we can use it for quicker prototyping. An 8B model fine-tunes much faster, so we'll use it in early experiments to validate ideas (like testing the persona embedding approach or trying RL tuning quickly). Once things work, we scale up to 20B. - **LLaMA3.1-8B:** Similarly, LLaMA 3.1 (8B) could be an option. The Anthropic research on persona vectors specifically used LLaMA-3.1 8B [34], meaning that model exists and has some known behaviors we can leverage. Using it might allow us to cross-reference their findings (like known persona vectors) directly.

All these models will be run primarily in **FP16 or BF16** precision for efficiency, possibly with 8-bit optimizers for fine-tuning to save memory. If memory allows, we might do some training in FP32 for stability, but 20B at FP32 might not fit on a single GPU.

2. GPU Hardware and Hours: We anticipate using NVIDIA A100 GPUs (40GB or 80GB memory versions). The fine-tuning and RLHF will be the most intensive phases: - **Supervised Fine-tuning (Stage 1 & 2):** For a 20B model, fine-tuning with our dataset (let's estimate ~50k training samples of dialogues) might take on the order of a few epochs. Using an 8-GPU machine with A100 40GB, we might handle a batch of maybe 1-2k tokens per GPU (depending on sequence lengths). If each sample is say 300 tokens on average (persona+context+response), we could fit ~3-4 samples per GPU, times 8 GPUs = ~32 samples per step. 50k samples / 32 ~ 1563 steps per epoch. If we do 3 epochs, ~4700 steps. At maybe 1 step/sec, that's ~1.3 hours per epoch, ~4 hours total – but this is likely optimistic. More realistically with communication overhead, etc., maybe ~8-10 hours for a full fine-tune. We'll allocate say **50 A100-hours** for initial supervised fine-tuning runs, including hyperparameter tuning. - **Reinforcement Learning:** RL training (PPO) is more complex; each step requires generating outputs (which is slower than a forward pass). If we simulate e.g. 1000 dialogues of 10 turns for experience, that's 10k generations to produce. We might parallelize this across GPUs or use batch generation. Assuming each generation is ~1-2 seconds for the model at 20B (depending on output length), generating the experiences could take a few hours. Then optimization on those experiences maybe another hour or two. We might iterate PPO for several rounds. So perhaps ~20 hours for a solid RL training run. We will plan for multiple runs to tune reward weights, so maybe **100 A100-hours** reserved for RL experiments. - **Evaluation & Others:** Running large-scale evaluations (with thousands of conversation simulations) also costs GPU time, but we can often do generation on fewer GPUs and it's parallelizable. We anticipate another **20-30 A100-hours** for all evaluation runs combined (like if we do grid search across personas, etc.).

In total, the project might use on the order of **200-250 A100 GPU-hours**. If we have an 8xA100 node, that is roughly 1 day of usage (since 8 GPUs for ~32 hours = 256 GPU-hours). We will likely spread this over multiple shorter runs, but overall it's manageable. If we require more (for example, if we decide to try a

70B parameter model for final tests), we might need to triple this estimate. We will keep 70B as a stretch goal if resources allow.

3. Memory and Storage: A 20B model in FP16 is about 40 GB of weights. We need to store multiple versions (pretrained, fine-tuned, etc.) so have storage for maybe 3-4 copies (160 GB). The optimizer states can be large too, but with newer optimizers like DeepSpeed's Zero or 8-bit Adam, we can reduce the memory footprint. We'll ensure we have at least 1 TB of disk for datasets, model checkpoints, and logs.

4. Inference Costs: If deploying or testing with GPT-4/5 APIs for data processing, we need to account for those costs. We will use GPT-4 for data labeling and possibly in evaluation as an oracle for scoring persona consistency. Budgeting perhaps **$500-$1000** for API usage across the project (which should cover a few million tokens of GPT-4, likely sufficient).

For our model's inference, once trained: - Running one instance of our model for a conversation of 20 turns might take ~5 seconds per turn = 100 seconds per conversation on one GPU (just rough guess). For evaluation, say we simulate 100 conversations, that's 100*100 = 10,000 seconds ~ 2.7 hours on one GPU. With multi-GPU parallelism we can cut that down. So inference in evaluation is not a huge issue relative to training. - If this were to be deployed as a system, we would consider using smaller distilled versions for lower cost per query. But deployment is outside our current scope; still, we note that an 8B model variant might be used in practice for cost reasons, and our approach should scale down somewhat gracefully.

5. Scaling Plan: We intend to design everything with scalability in mind: - **Larger Models:** If initial results are promising on 20B, we might scale to a larger model like 70B (if an open LLaMA3 70B or similar is available) for better performance. We would then need more GPU memory (likely 16 A100s or using model parallelism). The techniques (persona embedding, etc.) should theoretically transfer. We might do a final run on a large model to see maximum quality of persona emulation. - **More Personas Concurrently:** Our architecture could in theory allow multiple personas to be loaded (like multiple persona embeddings) and switch between them. If time permits, we will test scaling to many personas in one model (like making it a multi-persona model that can impersonate anyone given a profile). This is more of an extra exploration, but it would test how scalable the representation is – can one model store dozens of distinct personas without interference? If not natively, perhaps using separate LoRA modules per persona could be a solution (then scaling is linear in number of personas). - **Efficiency Improvements:** We will consider efficient fine-tuning methods (prefix tuning, LoRA) to minimize GPU usage. This also allows scaling to bigger models by only training a fraction of parameters. We can also use gradient checkpointing to handle longer sequences (for dialogues) without blowing up memory.

6. Collaboration and Compute Resources: We have access to an academic cluster with ~4 nodes of 8xA100 each, as well as some cloud credits for overflow. The plan is to use those for heavy training and possibly use cloud instances for any bursty needs (like a quick sweep that we can run in parallel).

7. Risk Mitigation: In case our approach of modifying architecture proves too time-consuming on large models, our fallback is to stick to prompt-based methods which require less heavy training. That would reduce GPU usage drastically (since we'd just be doing inference and light fine-tuning for prompt tuning). However, we believe the allocated resources are sufficient for the ambitious plan.

In summary, the resource plan calls for leveraging powerful GPUs to train a 20B model with our persona system, with an estimated few hundred GPU-hours required. We consider this reasonable and have accounted for expansion to larger models if needed. The careful use of parameter-efficient tuning and the

selection of model sizes ensures we can iterate relatively quickly in research mode and only invest heavy compute once we are confident in the method, at which point scaling up is an option.

Project Significance

This research aims to advance the state-of-the-art at the intersection of AI and psychology, with several important contributions and impacts:

- **Bridging Cognitive Psychology and AI:** By explicitly incorporating psychological theories (like McAdams' model, CAPS, TESSERA, WTT) into LLM development, we are forging a new path for interdisciplinary research. This helps demystify AI behavior by giving it a human-relatable structure. It also provides a framework for scholars to discuss AI behavior in terms of traits, states, and identity, rather than as a black-box. In the long run, this could influence how AI personas are designed in industry – moving from ad-hoc prompt engineering to theory-driven profiles. It demonstrates a practical way to operationalize concepts from psychology within neural networks, contributing to the field of **computational psychology** and **AI alignment** (by aligning AI behavior with human-understandable norms of personality stability).

- **Addressing Gaps in LLM Consistency:** Current research and user experiences highlight gaps such as:

- Behavioral Stability: LLMs often cannot maintain a consistent style or set of preferences, especially over long interactions [5]. Our multi-layer model directly addresses this by giving the model an internal "self" that persists. If successful, this means no more chatbots that suddenly change tone or contradict their earlier characterization – a major step for reliability.
- Persona-Role Coupling: Typically, when an LLM is given a task, it might drop its persona to maximize task performance (e.g., become a generic question-answerer). Our approach enforces that the persona stays coupled to the role – the AI remains "itself" while doing the task. This decouples the common entanglement where persona is either entirely superficial or completely overridden. We ensure that even if the AI takes on a sub-role (teacher, friend, etc.), it does so as that persona, not forgetting its identity. This is akin to an actor playing different roles but the acting style or essence of the actor still comes through, which is a new capability for AI.

- Expressive Controllability: We provide mechanisms to dial personality traits up or down (through trait vectors or prompts) and to intervene if the persona veers off course (through vector steering or persona refinements). This level of control is not present in most current systems. It means developers or users could customize AI personalities in a principled way – for example, making a customer service bot more empathetic vs. more concise by adjusting a "agreeableness" knob. Our research will show how each layer can be targeted for control (traits via embeddings, states via memory, behaviors via style tokens), offering a blueprint for fine-grained personality control in LLMs.

- **Practical Significance for AI Applications:** The outcomes of this project can directly enhance various AI applications:

- **Virtual Assistants and Chatbots:** They will be able to maintain a consistent persona (aligned with brand values or user preferences) across long-term usage, increasing user trust and engagement. A chatbot that remembers its "personality" from session to session can provide a sense of continuity (imagine an assistant that the user can get to know over time).
- **Interactive Storytelling and Games:** NPCs (non-player characters) in games or characters in interactive fiction could use our model to behave more like real individuals with depth. They

would not break character even after many player interactions, and could show development (growth-based changes) as the story progresses, making narratives more compelling.

- **Personalized Education or Therapy:** An educational tutor AI or a therapeutic AI agent could have a defined supportive persona. Ensuring consistency in those domains is crucial (students or patients wouldn't trust an AI that suddenly behaves inconsistently). Our approach can help maintain the delicate balance of personality needed for such roles while adapting to the learner's or patient's needs in the moment.

- **Social Simulation and Research:** Social scientists using LLMs to simulate populations (an emerging area) need the agents to have stable diverse personas [35]. Our multi-layer model would allow simulations where agents have distinct trait profiles and realistic interaction patterns, improving the validity of such studies.

- **Mitigating AI Risks:** A stable persona can also mitigate certain AI risks. For example, emergent misalignment issues (like the model suddenly becoming toxic) [18] [19] might be curbed if the model's persona is anchored to pro-social traits and continuously monitored. By understanding persona at a deep level, we can introduce safety checks tied to personality – e.g., if "aggression" vector spikes, the system knows to bring it down because it violates the intended persona of the AI. This is a novel angle on AI safety, complementing content filtering with style consistency enforcement.

- **Scholarly Contribution:** We will produce (and aim to publish) a formal research paper detailing the three-tier personality model for LLMs, the training approach, and evaluation results. This will add to the body of knowledge in NLP and AI, possibly pioneering a subfield of personality modeling in language models. Additionally, the dataset we curate (multi-context persona dataset) will be a resource for others, and the evaluation methods (like trait inference for AI) could be adopted widely. We will cite and extend recent works [5] [22], thus continuing the exploration of identity drift and persona control with a fresh perspective.

- **Ethical and Societal Impact:** On an ethical note, giving AI a consistent persona might make interactions more transparent and comfortable for users – you know what kind of entity you're dealing with. It could also reduce instances where AI outputs cause harm due to inconsistency (e.g., saying something empathetic then something cold). However, it also raises the capability of AI to deeply mimic human-like personalities, which could be used maliciously (e.g., impersonation). Our research will include discussion on these aspects, and our controllability features actually could help in detection (if we can monitor persona vectors, we might detect when an AI is purposely altered). Overall, by bringing AI behavior closer to human norms, we hope to improve human-AI synergy.

In conclusion, this project is significant because it tackles a fundamental limitation of current LLMs with a novel, comprehensive solution. It not only seeks to improve technical performance (consistency, controllability), but also to enrich the conceptual understanding of AI "identity." If successful, we will have made a meaningful step towards AI that can genuinely embody a character in a stable way – a long-standing goal in AI and a feature that could transform user experiences across many domains. We will have demonstrated that principles from human psychology can be applied to create more robust and relatable AI systems, marking a move away from viewing LLMs as just predictive text engines and towards viewing them as entities with pseudo-personalities that can be shaped and sustained. This convergence of disciplines and its practical upshot is the key significance of our work.

---

1 3 4 5 6 9 10 11 33 Examining Identity Drift in Conversations of LLM Agents
https://arxiv.org/html/2412.00804v2

2 12 13 14 15 16 17 34 Persona vectors: Monitoring and controlling character traits in language models \ Anthropic
https://www.anthropic.com/research/persona-vectors

7 8 Where You Inject Matters: The Role-Specific Impact of Prompt Injection Attacks on OpenAI models | NCC Group
https://www.nccgroup.com/research-blog/where-you-inject-matters-the-role-specific-impact-of-prompt-injection-attacks-on-openai-models/

18 19 cdn.openai.com
https://cdn.openai.com/pdf/a130517e-9633-47bc-8397-969807a43a23/emergent_misalignment_paper.pdf

20 21 22 23 [2510.14205] DPRF: A Generalizable Dynamic Persona Refinement Framework for Optimizing Behavior Alignment Between Personalized LLM Role-Playing Agents and Humans
https://arxiv.org/abs/2510.14205

24 25 Narrative identity - Wikipedia
https://en.wikipedia.org/wiki/Narrative_identity

26 27 28 29 Cognitive-affective personality system - Wikipedia
https://en.wikipedia.org/wiki/Cognitive-affective_personality_system

30 31 Processes of Personality Development in Adulthood: The TESSERA Framework - PubMed
https://pubmed.ncbi.nlm.nih.gov/27260302/

32 (PDF) Whole Trait Theory: An integrative approach to examining ...
https://www.academia.edu/97183704/Whole_Trait_Theory_An_integrative_approach_to_examining_personality_structure_and_process

35 10 Must-Read Papers on AI Agents from January 2025 - Reddit
https://www.reddit.com/r/LLMDevs/comments/1ifjs6n/10_mustread_papers_on_ai_agents_from_january_2025/