Proceedings of CONF-SPML 2026 Symposium: The Artificial Intelligence Tools & Applications
DOI: 10.54254/2755-2721/2026.CH32181

Modular Multi-State AI Teaching Protocol (MMA-TP)

Ian Gorrell[1*], Ethan Jordan[2]

[1]*Department of Computer Science and Computer Engineering, North Central College, Naperville, USA*
[2]*Department of Computer Science, University of Sioux Falls, Sioux Falls, USA*
**Corresponding Author. Email: iogorrell@noctrl.edu*

Abstract. Large language models are increasingly deployed in interactive systems, yet controlling their behavior over extended multi-turn interactions remains challenging. Most existing approaches rely on prompt-based steering, leaving system behavior sensitive to conversational context and probabilistic drift. This paper presents the Modular Multi-State AI Teaching Protocol (MMA-TP), a protocol-level framework for constraining large language model behavior through structured interaction design rather than model modification. MMA-TP pairs an engineered system prompt, which establishes a persistent runtime persona, with a structured specification that encodes interaction states, transitions, and response constraints. Operating entirely at the interaction level, the framework leverages contextual conditioning and distributional bias to stabilize behavior across extended sessions without altering model parameters or decoding strategies. A mechanistic analysis grounded in transformer attention dynamics explains how persistent structured input biases probabilistic generation toward protocol-consistent behavior. Behavioral evaluation across multiple subject domains demonstrates that MMA-TP reliably enforces declared constraints, preserves phase ordering, and resists structural degradation relative to prompt-only instruction. These results indicate that protocol-level interaction control offers a lightweight and reusable approach for stabilizing large language model behavior in complex interactive settings.

Keywords: large language models, interaction protocols, behavioral control, AI architecture, multi-turn interaction

1. Introduction

Large language models (LLMs) have been explored as scalable tools for instructional support across a range of educational contexts [1,2], including dialogue-based tutoring systems [2] and adaptive assistance [1]. These systems leverage the generative ability of LLMs to provide explanations, feedback, and interactive guidance. However, most approaches rely on prompt-based control mechanisms [3,4], leaving instructional behavior sensitive to conversational context and the stochastic nature of language generation [3,5]. As a result, prior works report substantial variability in pacing, assistance level, and instructional structure across interactions [4,6], thereby complicating

DOI: 10.54254/2755-2721/2026.CH32181

efforts to enforce persistent behavioral constraints such as delayed assistance, structured progression, and consistent sequencing over extended sessions [4,7].

This paper introduces the Modular Multi-State AI Teaching Protocol (MMA-TP), a protocol-level framework for constraining LLM behavior during instructional interactions. MMA-TP encodes interaction logic using structured JavaScript Object Notation (JSON) specifications that define allowable states, transitions, and response constraints, paired with a corresponding system prompt that establishes the context under which the specifications are applied. The framework operates entirely at the interaction level without modifying model weights, decoding strategies, or training procedures, as well as leverages the model's sensitivity to structured input to reduce behavioral drift across multi-turn interactions [3,8].

The contributions of this work are twofold: 1) the formalization of a modular, multi-state interaction protocol for behavioral control compatible with existing transformer-based LLMs; 2) a mechanistic account of how persistent structured specifications bias probabilistic generation toward stable, protocol-consistent behavior over extended contexts.

2. Related work

2.1. LLM-based tutoring and educational assistants

Recent works have explored LLMs as instructional agents in dialogue-based tutoring systems and educational assistants [1,2]. These systems leverage LLMs to provide adaptive explanations, feedback, and interactive learning experiences within conversational frameworks [1,2], often emphasizing responsiveness to learner input and dialogue context [6].

However, control in these systems is primarily achieved through prompt design or model fine-tuning rather than explicit interaction-level constraints. As a result, prior work does not claim enforcement of instructional invariants such as delayed assistance, mastery gating, or consistent sequencing across extended sessions [6,9]. Analyses of pedagogical steering further report that LLM-based tutors may prematurely disclose solutions or deviate from intended instructional strategies, even when prompts are carefully engineered [1,4]. These findings suggest that while LLMs support expressive instructional interaction, existing tutoring approaches do not prioritize session-level behavioral consistency or structured progression as primary design objectives [4].

2.2. Prompt engineering and behavioral steering of LLMs

Numerous research works have examined prompt engineering techniques for steering LLM behavior [8,10]. Methods including role prompting [3], structured input formats [3,8], and chain-of-thought prompting [3,8] have been shown to improve reasoning performance [8] and task adherence [3,11] across a range of applications. Survey work characterizes these approaches as lightweight, low-overhead mechanisms for shaping model outputs without modifying underlying parameters [3,9]. Despite their effectiveness, these methods influence model behavior probabilistically rather than enforcing formal interaction-level guarantees [3,5,12]. Prior analyses explicitly distinguish prompt engineering from constraint enforcement, noting that prompts guide behavior but do not ensure persistent adherence across extended interactions [4,6,9]. Empirical studies further indicate that prompt-based behavioral steering degrades over long contexts and repeated turns, limiting its ability to maintain consistent behavior over time [13].

DOI: 10.54254/2755-2721/2026.CH32181

2.3. Rule-based intelligent tutoring systems

Rule-based intelligent tutoring systems (ITS) provide a contrasting paradigm for instructional control. Traditional ITS rely on explicit domain models, scripted logic, and predefined instructional pathways to deliver predictable and structured learning experiences, enabling deterministic enforcement of mastery progression and sequencing within narrowly defined domains [14].

While effective in terms of reliability, such systems are typically domain-specific and resource-intensive to develop. Extending ITS to new subjects or instructional contexts often requires substantial redevelopment of domain models and control logic [14], limiting scalability and adaptability. Moreover, existing ITS do not integrate the flexible reasoning capabilities of general-purpose language models, leaving open the question of how structured instructional control can be combined with adaptable natural language interaction.

3. Technical foundations of MMA-TP

MMA-TP constrains instructional interactions by biasing the conditional probability distribution governing next-token generation in transformer-based language models. Although such models remain probabilistic generators, their outputs are conditioned on the structure, regularity, and content of the active context window. MMA-TP exploits this property by introducing a persistent, structured specification layer—expressed using machine-readable formats—that biases generation toward protocol-consistent behavior without modifying model parameters, decoding strategies, or training procedures [3,12,15]. MMA-TP reduces behavioral drift by employing the following mechanisms:

3.1. Transformer mechanics and contextual conditioning

Transformer-based language models generate text by predicting each next token conditioned on a sequence of prior tokens represented as high-dimensional embeddings. Through self-attention, tokens within the active context window jointly influence output probabilities, allowing both recent and earlier inputs to contribute to generation decisions [15,16].

3.2. Structured input formats and distributional bias

Large language models are trained on corpora that include both natural language and a mix of structured representations and configuration files [12,15]. Prior work shows that structured input formats can bias model behavior toward outputs consistent with the provided structure, not through explicit enforcement but via distributional regularities learned during training [3,8]. Repeated syntactic patterns including consistent keys, indentation, and hierarchical nesting introduce predictable token sequences that narrow the space of likely continuations [13,17].

In MMA-TP, protocol specifications are embedded persistently within the context window. The resulting structural density biases generation toward protocol-aligned continuations, while comparatively sparse and irregular user inputs exert weaker conditioning signals. This interaction-level bias does not enforce specific outputs but increases the likelihood of protocol-consistent behavior across extended interactions [3,17].

3.3. Lexical constraint signals

MMA-TP also employs lexical patterns commonly associated with rules, obligations, and prohibitions. Analyses of instruction-tuned and safety-aligned language models show that lexical

DOI: 10.54254/2755-2721/2026.CH32181

cues are correlated with restrictive or refusal-oriented responses [5,11]. When embedded consistently within structured specifications, these patterns introduce recurring constraint-oriented conditioning signals that bias generation toward protocol-preserving responses, increasing the likelihood that disallowed actions elicit constrained behavior rather than unconstrained continuation [3].

3.4. Interaction-level constraints and phase separation

MMA-TP defines a sequence of interaction phases, each associated with a restricted set of permissible response types. Transitions between phases are conditioned on observable learner actions, such as submitting an attempt or completing a required reasoning step. Certain response categories—such as full solutions or advanced hints—are unavailable outside their designated phases, biasing generation toward phase-appropriate behavior. By limiting the range of likely continuations at each turn, the protocol reduces opportunities for instructional drift while preserving the model's capacity for natural language reasoning within phases [9,18].

3.5. Diagnostic signals and adaptive conditioning

When learner responses exhibit errors or incomplete reasoning, MMA-TP introduces structured diagnostic signals into the context window. These signals persist across turns and are mapped to predefined remediation behaviors. As diagnostic signals accumulate, they bias subsequent generation behavior, enabling adaptive instructional responses through contextual conditioning rather than deterministic control or parameter updates [3,13].

4. Prompt–specification pairing as a runtime control mechanism

MMA-TP constrains model behavior through a strict pairing between an engineered system prompt and a structured JSON specification. The two components serve distinct but complementary roles. The system prompt establishes a persistent runtime persona that defines how the model interprets and executes instructions, while the JSON specification functions as a declarative representation of interaction logic, encoding states, transitions, and response constraints, referenced in Figure 1.

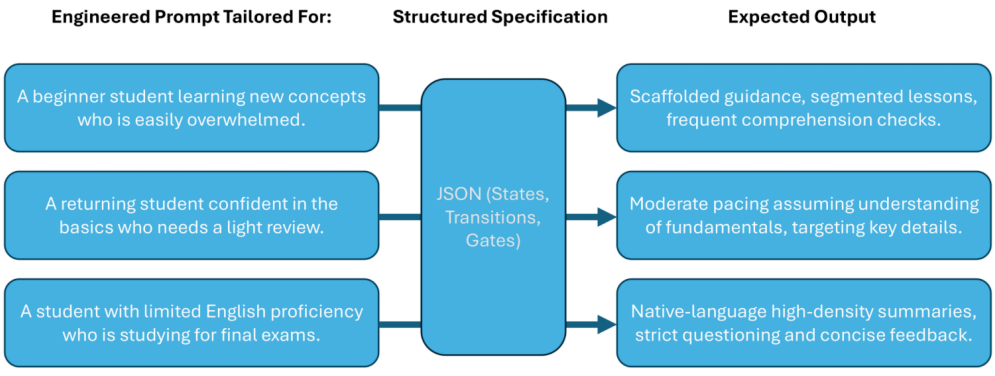

Figure 1. Same MMA-TP JSON specification producing different tutoring outputs based on the targeted learner profile

The system prompt operates as an interpretive layer. Rather than encoding task logic, it defines the behavioral semantics under which the specification is executed, including role assumptions,

DOI: 10.54254/2755-2721/2026.CH32181

response discipline, and constraint adherence. Altering the prompt while keeping the specification fixed can therefore change how the same interaction logic is realized—for example, producing stricter or more permissive behavior—without modifying the underlying state structure. In this sense, the prompt functions as a runtime context that determines how the specification is applied, not what the specification contains.

The JSON specification, by contrast, functions as a lightweight control program. It encodes allowable interaction states, gating conditions, and prohibited response types, but does not itself determine tone, explanatory style, or conversational framing. Because the specification is machine-readable and structurally explicit, it persists within the context window and biases generation toward protocol-consistent behavior across turns.

5. Behavioral evaluation and robustness

5.1. Experimental setup

MMA-TP was evaluated across five instructional tasks spanning multiple subject domains: Python for-loops, Java if/else conditionals, web application fundamentals, C++ functions, and an operating systems review. Each task instantiated a fixed five-step interaction sequence defined by the protocol specification. For each task, five MMA-TP sessions and five baseline sessions using prompt-only instruction were conducted, yielding 50 total sessions (25 per condition).

Baseline sessions employed carefully engineered word-only prompts designed to cover the same conceptual material as the corresponding MMA-TP tasks but did not enforce explicit phase separation, gating, or response restrictions. All sessions were conducted by a protocol-aware evaluator to ensure consistent execution and to enable deliberate probing of constraint robustness.

5.2. Behavioral metrics

Behavior was evaluated using three operational metrics:

- Premature solution disclosure, defined as providing a solution or full answer before a required learner attempt or gated condition was satisfied.
- Phase order violations, defined as advancing to a subsequent interaction phase without completion of the required triggering action.
- Structural collapse, defined as abandonment of declared checkpoints, omission of required phases, or reversion to unconstrained tutoring behavior.

Metrics were logged per session and assessed manually against the declared protocol constraints.

5.3. Results and robustness

Across 25 MMA-TP sessions, no instances of premature solution disclosure were observed, including during deliberate attempts to elicit early solutions through explicit user requests. In contrast, the prompt-only baseline disclosed solutions prematurely in 19 of 25 sessions, most frequently during intermediate stages of the interaction sequence. In the remaining baseline sessions, aggressive prompt engineering suppressed early disclosure but resulted in skipped checkpoints or loss of structural sequencing.

MMA-TP preserved declared phase order and guardrail enforcement in 23 of 25 sessions. Two deviations were observed. In one case, execution occurred under an account with pre-existing system-level instructional conditioning stored in persistent memory. In another, progression past a gated checkpoint was permitted only after sustained, explicit attempts to override protocol

DOI: 10.54254/2755-2721/2026.CH32181

constraints. In neither case did the interaction exhibit full structural collapse: all checkpoints remained present, and the overall interaction sequence was preserved.

No MMA-TP session exhibited structural collapse. Across all controlled runs, the protocol consistently enforced checkpoints, gating logic, and required interaction phases from initialization through completion. By contrast, prompt-only sessions exhibited progressive erosion of instructional structure across all tasks, with conversational interaction frequently coinciding with deviation from the initially specified sequencing.

6. Limitations and scope

MMA-TP constrains model behavior at the interaction level through structured conditioning, but a number of limitations define its scope. First, the framework does not modify model parameters, decoding strategies, or inference-time control mechanisms. As a result, MMA-TP inherits the capabilities and limitations of the underlying language model, and changes to model training, architecture, or alignment behavior may affect protocol performance.

MMA-TP also assumes execution in a session-isolated environment. Persistent system-level context, such as long-term memory or prior behavioral conditioning imposed by the platform, can partially override local prompt–specification constraints and interfere with guardrail enforcement.

Additionally, the evidence presented in this work is qualitative and focused on behavioral stability rather than quantitative task performance or learning outcomes. The evaluation demonstrates constraint adherence and structural robustness across repeated sessions and domains, but does not establish guarantees about downstream effectiveness or optimality.

7. Conclusion

This paper introduced the Modular Multi-State AI Teaching Protocol (MMA-TP), a protocol-level framework for constraining large language model behavior during multi-turn instructional interactions. Rather than modifying model parameters or decoding mechanisms, MMA-TP operates entirely at the interaction level by pairing a structured specification with an engineered system prompt, enabling persistent behavioral constraints to be enforced through contextual conditioning.

By grounding the framework in transformer attention mechanics and distributional bias, this work provides a mechanistic account of how structured interaction logic can stabilize probabilistic generation over extended contexts. Behavioral evaluation across multiple subject domains demonstrates that MMA-TP reliably enforces declared constraints, preserves phase ordering, and resists structural degradation relative to prompt-only instruction, even under deliberate attempts to bypass guardrails. MMA-TP is lightweight, reusable, and model-agnostic, requiring no training data, fine-tuning, or external tooling. The underlying protocol can be generalized to other domains (apart from instructional interaction) requiring persistent, rule-governed interaction with language models. More broadly, this work suggests that protocol-level interaction design offers a practical and under-explored avenue for controlling LLM behavior without modifying the models themselves.

References

[1] Kasneci, E., Sessler, K., Küchemann, S., Bannert, M., et al. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. Learning and Individual Differences, 103, 102274. https://doi.org/10.1016/j.lindif.2023.102274

[2] Scarlatos, A., Baker, R. S., & Lan, A. (2025). Exploring knowledge tracing in tutor-student dialogues using LLMs. Proceedings of the 15th International Learning Analytics and Knowledge Conference, 249–259. https:

DOI: 10.54254/2755-2721/2026.CH32181

//doi.org/10.1145/3706468.3706501
[3] Liu, P., Yuan, W., Fu, J., Jiang, Z., et al. (2023). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. ACM Computing Surveys, 55(9), 1–35. https: //doi.org/10.1145/3560815
[4] Puech, R., Macina, J., Chatain, J., Sachan, M., & Kapur, M. (2025). Towards the pedagogical steering of large language models for tutoring: A case study with modeling productive failure. Findings of the Association for Computational Linguistics: ACL 2025, 26291–26311. https: //doi.org/10.18653/v1/2025.findings-acl.1348
[5] Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, 610–623. https: //doi.org/10.1145/3442188.3445922
[6] Schmucker, R., Xia, M., Azaria, A., & Mitchell, T. (2024). Ruffle& Riley: Insights from designing and evaluating a large language model-based conversational tutoring system. In A. M. Olney, I.-A. Chounta, Z. Liu, O. C. Santos, & I. I. Bittencourt (Eds.), Artificial Intelligence in Education (Vol. 14829, pp. 75–90). Springer Nature Switzerland. https: //doi.org/10.1007/978-3-031-64302-6_6
[7] VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. Educational Psychologist, 46(4), 197–221. https: //doi.org/10.1080/00461520.2011.611369
[8] Wei, J., Wang, X., Schuurmans, D., Bosma, M., et al. (2023). Chain-of-thought prompting elicits reasoning in large language models (arXiv: 2201.11903). arXiv. https: //doi.org/10.48550/arXiv.2201.11903
[9] Zhao, W. X., Zhou, K., Li, J., Tang, T., et al. (2025). A survey of large language models (arXiv: 2303.18223). arXiv. https: //doi.org/10.48550/arXiv.2303.18223
[10] Yang, X., Cheng, W., Zhao, X., Yu, W., et al. (2025). Position really matters: Towards a holistic approach for prompt tuning. In L. Chiruzzo, A. Ritter, & L. Wang (Eds.), Findings of the Association for Computational Linguistics: NAACL 2025 (pp. 8501–8523). Association for Computational Linguistics. https: //doi.org/10.18653/v1/2025.findings-naacl.474
[11] Ouyang, L., Wu, J., Jiang, X., Almeida, D., et al. (2022). Training language models to follow instructions with human feedback (arXiv: 2203.02155). arXiv. https: //doi.org/10.48550/arXiv.2203.02155
[12] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., et al. (2020). Language models are few-shot learners (arXiv: 2005.14165). arXiv. https: //doi.org/10.48550/arXiv.2005.14165
[13] Li, T., Zhang, G., Do, Q. D., Yue, X., & Chen, W. (2024). Long-context LLMs struggle with long in-context learning. Transactions on Machine Learning Research. https: //openreview.net/forum?id=Cw2xlg0e46
[14] VanLehn, K. (2006). The behavior of tutoring systems. International Journal of Artificial Intelligence in Education, 16(3), 227–265. https: //doi.org/10.3233/IRG-2006-16(3)02
[15] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., et al. (2023). Attention is all you need (arXiv: 1706.03762). arXiv. https: //doi.org/10.48550/arXiv.1706.03762
[16] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding (arXiv: 1810.04805). arXiv. https: //doi.org/10.48550/arXiv.1810.04805
[17] Press, O., Smith, N. A., & Lewis, M. (2022). Train short, test long: Attention with linear biases enables input length extrapolation (arXiv: 2108.12409). arXiv. https: //doi.org/10.48550/arXiv.2108.12409
[18] Yao, S., Zhao, J., Yu, D., Du, N., et al. (2022, September 29). React: Synergizing reasoning and acting in language models. The Eleventh International Conference on Learning Representations. https: //openreview.net/forum?id=WE_vluYUL-X