

Peer Review

Review of: "Instruction-Driven Game Engine: A Poker Case Study"

Evgenii Arsentev¹

1. Independent researcher

1. What the work claims

The authors propose an Instruction-Driven Game Engine (IDGE): a fine-tuned language model that reads a free-form game script, takes a player action, and returns the next game state. Learning is framed as Next State Prediction (NSP), with a cheaper variant, Differential State Prediction (DSP), in which the model emits only the change between two states as code that an interpreter merges into a dict. Routine operations such as shuffling and hand ranking live in a set of predefined "core functions" that the model may call. Training follows a three-stage curriculum: a 1k-sample warmup on the core set, 10k standard samples from a purpose-built poker simulator covering ten variants, and a final stage on scripts partly rephrased by GPT-3.5 (Segment Rephrasing, SR).

The case study is poker. The paper reports that a LoRA-tuned CodeGemma-7b reaches 100% round-level success on 200 simulator rounds, that a 5-shot GPT-4 baseline completes none of them, and that on eight new games written by human players, the SR model succeeds in most rounds while the model without SR mostly fails.

2. Whether the conclusion holds on the data

I like the core design choice a lot. Splitting the job so that the model writes a small state-update program and a deterministic interpreter does the arithmetic and the randomness is a sensible answer to the known weakness of LLMs at exact bookkeeping, and the shuffle example in Section 2.2 makes the point well. The out-of-domain scripts in Appendix B (Table 5) are also genuinely inventive; "Stardust" and "Magic Dealer" are not trivial reskins of Texas Hold'em, and I appreciated that the authors printed them in full.

The direction of the results is believable. The strength of some specific claims runs ahead of what the tables can show, mostly because the test sets are small.

In-domain, Table 2. Each variant is played for 20 rounds. Summing the non-checkmark cells, NSP fails $2 + 2 + 3 + 2 = 9$ rounds (columns Om. HL, 27 triple, Badeucey, Badacey), i.e., $191/200 = 95.5\%$; DSP fails $2 + 2 = 4$, i.e., $196/200 = 98.0\%$; DSP (CS) fails 1 (A5 triple, 19/20), i.e., $199/200$; DSP (SR) and DSP (CS+SR) are $200/200$. So the ordering is consistent with the text. But the per-cell differences are tiny. For a single cell such as NSP 18/20 against DSP 20/20, a two-sided Fisher exact test gives $p \approx 0.49$ (the probability that both failures land in one arm is $20/40 \times 19/39 = 0.244$, doubled). Even the pooled NSP-versus-DSP gap (9 against 4 failures out of 200 each) is modest. I would present the pooled numbers with confidence intervals and soften "the advantage of predicting the difference" to what 200 rounds can support. With two rows at $200/200$, the table also cannot rank SR against CS+SR; it has hit its ceiling.

State-level, Table 3. This is the most informative table, and the GPT-4 row explains the zero round-level success convincingly: 31.3 on deal, 20.6 on switch, and 0.0 on show. Two things trouble me here. First, the text in Section 5.2 says that "CS is particularly beneficial for show," yet every CodeGemma row in Table 3 is DSP+CS+SR according to the caption. There is no row without CS, so the table does not show a CS effect on show at all. The only CS evidence in the paper is in Table 2, where DSP (CS) lifts Badeucey and Badacey from 18/20 to full success but drops A5 triple to 19/20. That is a round-level, mixed result, not a show-specific one. Second, the number of states behind each accuracy is not given, so I cannot tell whether 88.0 for the 5k model on show rests on dozens of states or hundreds.

The GPT-4 baseline. It is a fair reference point, but it is not a like-for-like comparison. GPT-4 is prompted with five examples, while CodeGemma is fine-tuned on 10k samples plus the core set. I could not find whether GPT-4 was given the core function definitions, the interpreter feedback, or the same DSP output format. Section 5.1 also states that CodeGemma "works better than" LLaMA3, but no LLaMA3 numbers appear anywhere. A second fine-tuned open model would tell the reader how much of the gain comes from the recipe and how much from the choice of backbone.

Out-of-domain, Table 4. Eight scripts from five players, ten rounds each, 80 rounds in total. Reading ✓ as 10/10 and ✕ as 0/10 (the caption does not define ✕, so this is my assumption), the SR model succeeds in $10 \times 6 + 9 + 8 = 77/80$ rounds, and the model without SR in $8 + 1 + 3 = 12/80$. That gap is large, and I have no doubt it is real. The Conclusion ("accurately complete the majority of user-defined games") is worded appropriately. What I could not work out is how correctness was judged for games that the simulator does not implement. Section 5.1 says the players "record all player inputs and game states throughout the

game-play" while playing with the IDGE itself. Who produced the reference state for each step, and who decided that a state was wrong? If the players did it by hand, the paper should describe that protocol and report agreement.

Training data quality. Section 3 says the second part of the data is generated by GPT-3.5, which augments the scripts and is prompted to "generate the corresponding new game states". Given that GPT-4 scores 0.0 on show and 31.3 on deal in Table 3, I would expect GPT-3.5-generated states to contain errors. How many such samples are in the 10k, and how were their states checked?

3. Reproducibility

The fine-tuning setup is described better than in many LLM papers. Section 5.1 gives LoRA $r = 8$ and $\alpha = 32$, 30 warmup steps, total batch size 8, 3 epochs per curriculum stage, and greedy decoding, and footnote 1 points to `google/codegemma-7b-it` on Hugging Face. Several things are still missing.

- No code, simulator, core set, trained adapters, or data release that I could find in the Qeios text. Without the simulator, the in-domain test cannot be rebuilt.
- The learning rate sentence ("the optimal learning rate in $1.5e-4$ and $3e-4$ ") does not say which value was used for which stage or model. "8 chips" does not name the hardware. No training time is reported.
- The GPT-4 snapshot, temperature, and the five in-context examples are not given. The same holds for the GPT-3.5 prompts used for augmentation and for Segment Rephrasing, including the rephrasing probability.
- No seeds and no repeated runs. Every number in Tables 2–4 appears to come from a single training run.
- Randomness in evaluation is not addressed. Section 5.1 compares predicted code to the reference and, if they differ, executes both on the input state. For a call like `shuffle`, two executions give different decks unless the interpreter seed is fixed. I would state how this is handled.
- In the Qeios text I worked from, Eq. 1 is readable, but Eq. 2 and the DSP equation (Eq. 3) in Section 2.3 come through only as fragments; Table 1's header is also broken, though its values (10k, 439.8, 401.1, 404.9, 135.9, 35.3) can be read. Please check that the equations render on the Qeios page.

4. What specifically to fix

1. Section 5.2 and Table 3: either add a DSP+SR row without CS, or remove the sentence that CS is particularly beneficial for show. As it stands, the claim has no supporting row.
2. Table 2: report pooled success per method over the 200 rounds (by my count 191, 196, 199, 200, 200) with 95% confidence intervals. Say plainly that the top two rows cannot be separated.
3. Table 3 needs the count of test states for each column. It would also help to explain the A–D difficulty labels; I was surprised that start is rated A (hard) while shuffle is D.
4. Table 4: define $\times$ in the caption, and state whether the "IDGE" column includes the core-set warmup.
5. Section 5.1, out-of-domain evaluation. Describe how ground-truth states were obtained for games with no simulator, who judged each state, and whether a second person checked. Five players wrote eight scripts (1–2 each); a small table of which player wrote which script would make this transparent.
6. Section 3, second data source: give the number of GPT-3.5-generated samples inside the 10k and the procedure used to verify their states. If they were not verified, an ablation without them would be valuable.
7. Table 1 is ambiguous. It calls the 10k items "state prediction samples" but also gives 35.3 states per game. Is a sample one state transition or one full game? If transitions, roughly how many distinct games does that make?
8. Section 2.3 argues DSP accelerates inference. Table 1 supports a shorter output (135.9 against 404.9 tokens, about a third), but no latency is measured. One wall-clock number per state would make the efficiency claim concrete.
9. Add a fine-tuned baseline on a second 7B model, since Section 5.1 already mentions LLaMA3. Show its numbers rather than just the conclusion.
10. For the GPT-4 baseline, list the snapshot, temperature, and whether it saw the core function definitions and the DSP format. A short few-shot result for CodeGemma without fine-tuning would also help separate the effect of training from that of the backbone.
11. In-domain "unseen variants": describe how far these configurations are from the training ones (which elements were changed, and whether any exact configuration overlaps).
12. A few wording points. "Buckets effect" in Section 5.2 is usually called the cannikin (or bucket) law. The sentence in Broader Impact that Delta-Engine "is largely inspired from our work" refers to [19],

which shares three authors with this paper (Wu, Wang, Zhao); that overlap should be stated for transparency.

5. Recommendation

Minor revisions. The idea is original and the engineering is sound: code-mediated state updates, core functions, and a curriculum that separates stability from diversity make a convincing package, and the out-of-domain gap in Table 4 (77/80 against 12/80 by my reading) is strong evidence that Segment Rephrasing matters. The requested changes are about aligning the text with the tables (the CS claim), adding denominators and intervals to small test sets, and documenting how the human-authored games were scored. None of this needs a new study. I checked every number I quoted against the Qeios version; I could not see the figures, so I have not commented on them.

Declarations

Potential competing interests: No potential competing interests to declare.