

Peer Review

Review of: "Star-Agents: Automatic Data Optimization with LLM Agents for Instruction Tuning"

Evgenii Arsentev¹

1. Independent researcher

Summary

The paper proposes Star-Agents, a pipeline for rewriting an existing instruction-tuning corpus so that the resulting samples sit at a difficulty level suited to a particular target model. Three parts: (1) pairs of different LLMs rewrite the instruction and the response separately (Section 3.1, Eq. 2), producing several candidates per seed sample; (2) candidates are scored by a "dual-model" criterion — the gap between the IFD score of a small model and a large one (Section 3.2, Eq. 4–5) — multiplied by a pairwise LLM-judge score (Eq. 6–7); (3) the sampling probability of each agent-pair is updated online in proportion to that score (Section 3.3, Eq. 8), so productive pairs are called more often. The seed corpus is the 70,000-sample Evol-Instruct set (Section 4.1, Datasets), the target models are Pythia-1B and Llama-2-7B (Section 4.1, Models), and the evaluation is GPT-4-as-judge on Vicuna-Bench, the WizardLM testset, and MT-Bench (Section 4.1, Benchmarks). The headline claims are an average improvement of about 12% and gains above 40% on Fermi problems. The 12% is Table 2, column "Average": Pythia-1B-evol_instruct 3.73 against Pythia-1B-star_instruct 4.17, a difference of 0.44. The 40% is attributed to Figure 4(a), which in the version I was given is a caption with no readable values.

The core idea is reasonable and, as far as I can tell, new in this combination: matching data difficulty to model capacity rather than maximising difficulty is a sensible response to the Evol-Instruct line of work. My concerns are about what the reported numbers can actually carry.

Does the claim hold on the data

Partly, and less firmly than the abstract suggests.

Everything rests on single runs. There is no seed, no repetition, no interval, no test anywhere in the paper. The word "significant" or "significantly" is used in the Introduction (final paragraph), in Section 4.2, and three times in Section 4.3, always about differences established by one training run each. The most important of these is the evolution component: Table 3, rows 1 and 2, average score 4.17 with evolution versus 3.97 without — 0.20 points on a 1–10 GPT-4-judge scale, called "a significant improvement" in the Main Components paragraph of Section 4.3. I cannot tell that gap from run-to-run variance, and neither can the reader, because the variance is never reported. Same for MT-Bench in Table 2: Pythia-1B-evol_instruct 2.56 against Pythia-1B-star_instruct 2.69, a gap of 0.13.

The headline 12% is the better of two scales. Table 2, "Average" column: at 1B the gap is $4.17 - 3.73 = 0.44$, i.e., 11.8%, which rounds to the claimed 12%. At 7B it is $6.95 - 6.35 = 0.60$ on a larger base, i.e., 9.4%. The abstract and the Conclusion both present 12% as the method's average improvement without noting that the second target model gives a smaller relative gain. It is a small thing arithmetically and a large thing rhetorically.

The 40% cannot be checked from the text. Section 4.2 attributes it to Figure 4(a), "nine metrics", with improvements "up to 40%" in math, coding, and Fermi. There is no table of per-subtask scores anywhere in the paper, and the paper never states how many test items the Fermi subset contains. A "up to 40%" taken as the maximum over nine subtask categories of one benchmark, each category holding a handful of items, is close to the definition of a selected point. Promoting it into the abstract as "a 40% improvement in Fermi" overstates what a per-category split of an 80-item benchmark can support. At minimum, the per-subtask counts and raw scores need to be in a table.

Length is an uncontrolled confound, and it is the obvious explanation. The selection signal is an LLM judge (Eq. 6) and an IFD-based ratio (Eq. 4) that normalises by $|R_i|$; the final evaluation is GPT-4-as-judge; and every case study in the appendix shows the star_instruct output being longer and more structured than the evol_instruct output. In Table 6, both answers reach the same correct result, \$12,000 — the difference is that one shows the work. In Table 7, the star_instruct answers are longer and bulleted. The judge prompt in Table 5 contains the line "Do not allow the length of the responses to influence your evaluation", which is an acknowledgement that the bias exists, not a control for it. The paper reports no average response length for the models trained on each dataset, and no length-matched comparison. Until that is reported, "Star Instruct produces better responses" and "Star Instruct produces longer responses that GPT-4 prefers" are not separated.

Best-of-N is not separated from multi-agent diversity. Star-Agents generates several candidates per seed sample and keeps the arg max (Eq. 1); Section 4.1, Implementation Details, says 10 agent-pairs were used. The baseline, Evol-Instruct, is one ChatGPT sample per seed. So the comparison in Table 2 is best-of-many against one, under two labels that suggest a like-for-like data comparison. The ablation does not close this: Table 3, row 3 (diversity only, no selection, no evolution, 3.62) removes the selection but also removes the best-of-N, so there is no configuration isolating the two. The missing control is simple — draw the same number of candidates from a single strong agent, apply the same dual-model selection, and report that row. If that row lands near 4.17, the multi-agent framing is doing much less work than the paper claims.

The motivating observation is demonstrated at a scale the paper does not target, and the paper's own Table 2 contradicts it at the scale it does target. Section 3.2 justifies "tailored complexity" by pointing to Figure 2 and models "with 14M and 70M parameters", for which Evol-Instruct is said to be worse than Alpaca. The targets in Section 4.1 are 1B and 7B — one to two orders of magnitude larger. And at 1B, Table 2, "Average" column, Pythia-1B-evol_instruct (3.73) beats Pythia-1B-alpaca (3.10), so the harder corpus is the better one. The premise motivating the method does not reproduce at the smallest scale the paper actually evaluates. This does not sink the method — the selection criterion may work for other reasons — but the motivation as written is not supported by the paper's own main table.

The multi-agent claim may be thinner than it looks. Section 4.3, Evolution, names three agent-pairs: Mistral-ChatGPT rising to about 30%, Phi2-ChatGPT decaying to near zero, ChatGLM3-ChatGPT roughly flat. All three use ChatGPT as the response agent. If the surviving pairs are predominantly ChatGPT-on-response, then the released corpus is closer to "ChatGPT responses to diversified instructions, filtered by best-of-N" than to a genuinely multi-source corpus — and the diversity argument in Section 3.1 needs revisiting. The paper never reports the final composition of the Star Instruct dataset by agent-pair, which is a one-table fix and would settle the point either way.

The 7B comparison mixes measured and quoted numbers. In Table 2, the five rows zephyr-beta-sft, mpt-7B-chat, XGen-7B-8k-Inst, sRecycled-Wiz-7B-v2, and Llama-2-7B carry values only in the MT-Bench column; Vicuna-Bench and WizardLM testset are "-", and there is no Average. Those MT-Bench values are cited to [26] and [27], that is, lifted from other papers, and are therefore produced by a different GPT-4 judge at a different time. The nearest competitor, sRecycled-Wiz-7B-v2 at 5.56, is 0.18 below Llama-2-7B-star_instruct at 5.74. A cross-paper difference of 0.18 on GPT-4-judged MT-Bench is not a difference I

would be willing to defend, yet Section 4.2 states outright that the star_instruct model "outperforms the sRecycled-Wiz-7B-v2". Either re-run those baselines under the same judge or soften the claim.

The selection-method table has a size confound. Table 4 compares Pythia-1B-IFD (3.26), Pythia-1B-Random (3.62), and Pythia-1B-star_instruct (4.17). IFD and Random are subset-selection methods: they keep a fraction of the corpus. Star-Agents rewrites and keeps one sample per seed, so presumably 70,000. The paper never states how many samples the IFD and Random baselines retained. If those baselines trained on a subset while star_instruct trained on the full 70,000, then Table 4 is partly measuring dataset size, and the claim in Section 4.3 that the dual-model strategy beats IFD by "approaching a 0.9 point" is not clean. This number is quoted in the text as an advantage of the selection criterion, so it matters.

Reproducibility

This is the weakest part of the paper, and several of the gaps are load-bearing rather than cosmetic.

Section 4.1, Implementation Details, gives: three epochs, Adam, initial learning rate 2×10^{-5} , maximum 2048 tokens, batch size 64, 10 agent-pairs. That is the complete list of training and method hyperparameters in the paper. Missing from it:

Method hyperparameters. β , the evolution rate in Eq. 8, is defined and never given a value. n , the number of nearest matches retrieved from the Instruction Memory Bank (Section 3.3), is never given. l , the number of agent-pairs drawn from the retrieved pool, is never given. M , the number of agent-pairs sampled per seed sample in Eq. 3, is never given — and its relation to the "10 agent-pairs" of Section 4.1 is unclear. If there are 10 pairs and $M = 10$, the sampling in Eq. 3 is a no-op, and the cost argument that motivates it collapses; if $M < 10$, then M is a free parameter with no stated value. The embedding model used for the memory-bank similarity search is never named. The "base agent-pairs" introduced at the end of Section 3.1 — the fixed set that produces the reference sample for the judge comparison in Eq. 6 — are never identified. Which 10 of the possible pairs from the six models in Table 1 were used is never stated.

The large model in the dual-model criterion is never named. Eq. 5 requires $\text{IFD}_{\text{large}}$, and Eq. 4 requires token-level log-probabilities, so the large model cannot be ChatGPT as accessed through the API. Section 4.1 lists the target models (Pythia-1B, Llama-2-7B) and the generator agents (Table 1), but nowhere states which model plays $\text{IFD}_{\text{large}}$. The central selection criterion of the paper is therefore not specified. It is also unstated whether $\text{IFD}_{\text{small}}$ is computed with the untuned base target model or with a briefly-tuned checkpoint, which in the Cherry LLM setting the criterion is borrowed from is a consequential choice.

No model snapshots, no decoding parameters. Table 1 lists "ChatGPT" with no version and no date; the footnote points at chatgpt.com. ChatGPT is simultaneously a generator agent (Table 1), the evaluator producing π_{llm} (Section 4.1, Models, final sentence), and the model that produced the Evol-Instruct responses being compared against. GPT-4 is the benchmark judge (Section 4.1, Benchmarks) with no snapshot given either. Temperature, top-p, and any sampling seed are absent for every generation and every judging call. For a paper whose entire contribution is data produced by API models and scored by API models, the snapshot identities are the difference between a reproducible result and an anecdote. Table 1 also lists ChatGLM at 6B while Section 4.3 refers to ChatGLM3; the version actually used should be pinned.

No judge validation. π_{llm} (Eq. 6) decides which candidate enters the corpus, and GPT-4 decides the final scores. Neither judge is validated against human preference on any subset, and there is no inter-rater or self-consistency check. The judge prompt in Table 5 has an internal inconsistency that should be fixed regardless: the system prompt opens with "the responses provided by three AI assistants", then instructs the model to compare "the two responses", and the template supplies only Assistant A and Assistant B. The same prompt contains "bset" twice where "best" is meant. These are typographical in the paper but functional in the prompt — that is the text the judge model actually received, and a prompt that announces three assistants and shows two is a live source of judge noise. If the released prompt differs from Table 5, say so; if not, the selection step needs re-running with a corrected prompt.

Cost reporting. Appendix A.5 gives 4×10^{12} MACs for Qwen-14B and 7×10^{11} MACs for Phi-2-2.7B at 256 tokens, then estimates $4 \times 10^{12} \times 10 \times 70,000 = 2.8 \times 10^{18}$ and $4 \times 10^{12} \times 5 \times 70,000 = 1.4 \times 10^{18}$. The arithmetic checks out, but the estimate has four problems. Units are mixed inside a single line: the quantity defined as MACs is relabelled "FLOPs (Qwen-14B)" in the calculation, and the product is reported back in MACs. The per-model cost of the largest agent is applied to all ten agents, although the appendix's own Phi-2 figure is about 5.7× smaller. The 256-token assumption is far below the response lengths visible in Tables 7 and 8. And the estimate omits the two costs that matter most in practice: the dual-model IFD scoring over every candidate, and the ChatGPT judging call per candidate over 70,000 seeds. No wall-clock time, no token counts, no monetary cost. The "5 of 10 LLMs are used" figure in the second estimate is also an assumption, not a measurement — Section 4.3 describes probabilities drifting continuously, not a hard cut to five.

Code. The abstract says codes will be released "soon" and footnote 1 gives a repository URL. I reviewed only the text supplied to me and did not inspect the repository. Given how many of the method's

parameters above are absent from the paper, the release needs to include the exact prompts, the agent-pair list, β , n , l , M , the IFD model identities, and the generated Star Instruct corpus itself, or the work is not reproducible from the paper alone.

Contamination. No decontamination check is reported between the 70,000 seed instructions (or their rewrites) and Vicuna-Bench, MT-Bench, and the WizardLM test set. The last of these is drawn from the same line of work as the seed corpus, which makes the check worth running and reporting.

Specific points

1. **Table 9, row "Wizardlm": the Average does not match the row.** The four reported scores are ARC 51.60, HellaSwag 77.74, MMLU 42.74, TruthfulQA 45.75; their mean is 54.46, but the Average column reads 54.18. The other two rows are consistent: Llama-2-7B-evol_instruct (51.88, 76.70, 45.76, 46.10) averages to 55.11 as printed, and Llama-2-7B-star_instruct (54.44, 77.64, 46.94, 46.13) averages to 56.29 as printed. So one row is out by 0.28 while the others check. Please recompute or state which column was excluded.
2. **Table 9 is never discussed, and it complicates the story.** Section 4.2 points the reader to "Appendix A.4" for the Open LLM Leaderboard results and says nothing further; in the version I was given there is no A.4 heading at all — Table 9 sits between Section A.3 and Section A.5. The table is worth discussing because the gains are uneven: star_instruct over evol_instruct is +2.56 on ARC and +1.18 on MMLU but +0.03 on TruthfulQA, and on HellaSwag the Wizardlm row (77.74) is above star_instruct (77.64). A method claimed to improve data quality in general should address why one benchmark moves by two and a half points and another by three hundredths.
3. **Table 3 and Table 4 appear to share a run under two different descriptions.** Table 3, row 3 (Diversity ✓, Data selection ✗, Evolution ✗) reads 3.62; Table 4, row "Pythia-1B-Random" also reads 3.62. Section 4.3, Main Components, describes the first as diversified multi-agent data with random sampling; Section 4.1, Baselines, describes "Random select" as a data-selection baseline, which reads as random subset selection from the original corpus. These are different experiments and cannot both be 3.62 by construction. Please state explicitly what Pythia-1B-Random selects from and at what size. The interpretation of both ablations depends on the answer.
4. **Table 3 has no error bars and its two decisive gaps are small.** 4.17 vs 3.97 (evolution) and 3.97 vs 3.62 (selection). The first is 0.20. With one run per row and a GPT-4 judge, neither gap is separable

from noise as reported. Three seeds per row with the spread shown would settle this, and it is cheap at 1B.

5. **Eq. 7 has a degenerate case that is not discussed.** $\pi = \pi_{\text{llm}} \cdot \pi_{\text{dual}}$, and $\pi_{\text{llm}} = 0$ whenever the base sample wins (Eq. 6). If every candidate for a given seed loses to the base, all candidates score exactly 0 and the $\arg \max$ in Eq. 1 becomes an arbitrary tie-break — a low-quality sample enters T by accident. How often does this happen, and what is the fallback? Keeping the base sample in that case would be the obvious fix and should be stated.
6. **Eq. 5 can go negative, and Eq. 8 does not guard against it.** π_{dual} is $\text{IFD}_{\text{small}} - \text{IFD}_{\text{large}}$ divided by the maximum of that difference over the batch. Nothing in Section 3.2 forces the numerator to be non-negative; the assumption that "stronger model yields a smaller IFD score" is stated as an assumption in the paragraph following Eq. 4 and is never measured. A negative π_{dual} gives a negative π , and Eq. 8 then subtracts from p_{jk} , which can drive a probability below zero before normalisation. Please either report the empirical fraction of negative differences, clip the score, or state the clipping that was used.
7. **Eq. 5 uses an undefined index bound.** The maximum is taken over $1 \leq i \leq m$, but m is not defined anywhere — batch, candidate set for one seed, or the whole corpus. The three give materially different normalisations, and the third would make the score depend on a global pass.
8. **Eq. 1 is malformed as written.** $T = \{\arg \max_{d \in D(S_i)} \pi(D(S_i)) \mid i = 1 \dots N\}$: the $\arg \max$ ranges over d , but π is applied to the set $D(S_i)$ rather than to d , so the objective does not depend on the variable being optimised. It should read $\pi(d)$. Minor to fix, but it is the paper's first equation.
9. **The evolution result is confounded with sample ordering.** Section 4.3, Evolution, describes probabilities changing over "about 70,000 iterations", i.e. one iteration per seed sample, a single online pass. This means the early part of the corpus was built with a near-uniform agent mixture and the late part with the evolved one, and the two halves of Star Instruct are not drawn from the same process. Was the seed order shuffled? Does a second pass with the converged probabilities do better than the first? That second-pass number would be the clean measurement of what evolution buys, and it is a much stronger experiment than Table 3, row 2.
10. **Figure 2 carries the paper's central motivating claim and, in the supplied version, no numbers.** Section 3.2 states that for 14M and 70M models Evol-Instruct underperforms Alpaca, citing Figure 2. In the text I was given, Figure 2 is a caption only — no axis values, no table. Given that this observation motivates the entire dual-model criterion, and given that it is reversed at 1B in Table 2

as noted above, the underlying numbers belong in a table with the model sizes, the datasets, and the benchmark used.

11. **Figure 4 and Figure 5 likewise arrive as captions only in this version.** Figure 4(a) is the sole source of the abstract's 40% Fermi claim, and Figure 5 is the sole evidence for the evolution dynamics, including the "approximately 10%", "30%", and "near zero" values quoted in Section 4.3. I can evaluate neither. Both should be accompanied by tables of the plotted values — for Figure 4, per-subtask scores with the number of items in each subtask; for Figure 5, the final converged probabilities for all 10 pairs, which would also answer the question about ChatGPT's share of the response side.
12. **Table 8 does not illustrate what Section 3.1 describes.** The appendix table is titled "Diverse Data Examples" and shows one instruction with an Evol-Instruct response and four agent-pair responses. But Eq. 2 says the agent-pair rewrites the instruction as well as the response, and here the instruction is identical throughout. An example showing the instruction rewrites — which is half the method — would be more informative than a fifth response variant.
13. **The Alpaca baseline is non-standard and differs in size.** Section 4.1, Datasets, states that the Alpaca set used here was built with ChatGPT rather than text-davinci-003, and comprises 52,000 samples against Evol-Instruct's 70,000. Both facts should be repeated in the Table 2 caption, since the row Pythia-1B-alpaca (Average 3.10) is otherwise read as the published Alpaca and is quoted in Section 4.2 as evidence of a 1-point absolute improvement.
14. **Several rows in Table 2 are not comparators.** Pythia-1B (1.40 average), OPT-1.3B (1.75), and Sheared-LLaMA-1.3B (2.06) are base models without instruction tuning; on MT-Bench they score 1.17, 1.12, and 1.59 on a scale whose floor is 1. Including them inflates the apparent spread of the table without supporting the claim. Nothing is wrong with listing them, but Section 4.2's "consistently outperform nearly all aligned counterparts" should not be read as covering them.
15. **The showcased "better" answer contains a wobbly statement.** Table 7, Llama-2-7B-star_instruct on Riemannian geometry: "Euclidean geometry deals with flat spaces where the curvature of the Earth's surface is negligible" conflates a mathematical property with a physical approximation, and it sits in the passage presented as the improvement over the baseline. If the case studies are meant to demonstrate quality rather than format, it is worth picking examples where the winner is also correct throughout.
16. **Language and consistency.** "Finally" in the abstract; "Model Famliy" as the first column header of Table 1; "calcuated" in Section 3.2; "soces" at the end of Section 3.3; "comparsion" in Section 4.1,

Baselines; "Evolutiuon" as the third component header in Table 3; "Imapct" in the caption of Table 4; "star_instrcut" in the caption of Figure 4; "bset" twice in Table 5. The Acknowledgements thank "the anonymous reviewers", which on an unreviewed preprint posting reads as a leftover from a different submission. None of these change the science; together they make the manuscript read as unproofed, and the Table 5 instances are inside a prompt that was actually executed.

Recommendation

Major revisions. The idea — select for a difficulty gap between a weak and a strong scorer rather than for maximum difficulty — is worth publishing, and the pipeline is real work. But the headline numbers as presented cannot carry the claims: a 0.44-point average difference (Table 2, "Average" column) and a 0.20-point ablation gap (Table 3, rows 1 and 2) from single runs with no seeds and no intervals; a 40% figure that lives only in a figure with no reported values; a best-of-N versus best-of-one comparison under two dataset labels; an uncontrolled length confound between an LLM-judged selection step and an LLM-judged evaluation; and a selection criterion whose large model is never named. The revisions I would want are concrete rather than open-ended: repeated runs with spread for Tables 2, 3, and 4; the per-subtask table behind Figure 4 with item counts; average response lengths per trained model, or a length-controlled judge comparison; a single-agent best-of-N control row; the subset sizes for the IFD and Random baselines; the missing hyperparameters (β , n , l , M , the two IFD models, the embedding model, the base agent-pairs, the ChatGPT and GPT-4 snapshots); and a corrected, published judge prompt. Most of these are inexpensive at the 1B scale the paper already works at, and several are table edits rather than experiments.

Declarations

Potential competing interests: No potential competing interests to declare.