Dasein Bench: Does This Read Like Someone Who Lives in the World?

Minsung Park — compiled Taipei, 2026-11-11

Technical report, draft v0.2.

Status. Results are from `results.json`, pipeline v2.4 (frozen 21:15 Taipei, 2026-11-11), final read at 22:07 Taipei (W2 final update). Cells marked **[NOT RUN]** were not measured.

1 Problem and main result

This report introduces Dasein Bench[1], a benchmark for one question: does a reply read like it came from a person living in the world?

Current assistant models write text that is fluent but interchangeable. We test whether giving a model one person's real context, and selecting its outputs with a reward built from that person's data, moves the output toward that person. In this run it raises rubric scores, including on a judge outside selection, and does not clearly raise blind pairwise human-likeness (Table 1c). We measure the move with metrics a skeptical reader can check. We do not claim to be the first personalised alignment method (Section 7).

YC's W27 Requests for Startups lists "LLMs That Write Well" (Jared Friedman) and states that "the same reward signals that make models good at code actively flatten voice and style" [6]. The request ships no measurement. This report is a first one, on one subject.

Table 1. Ablation. One subject, 3 tasks (day blog, essay on the last few days, follow-up message). The "replies" task was dropped because of data leakage. "Runs" is the number of scored outputs behind each row. S1-S4 are 1 run per task with no seeds. S5 and S7 are 3 seeds per task, with headline scores on the median seed. Dasein (S6) is **1 pipeline run per task**. Reward v2 is computed by the selection judges (Claude and Astra), the same judges that rank Dasein's candidates, so it is **not independent evidence** for S6. The headline column is the qwen-27b rubric, which is not used in selection. It scored all systems, 3 tasks each. **Read it with the invented-facts column, not alone**: see the caveat below the table.

System	Runs	qwen-27b rubric (headline)	qwen-27b invented facts	Reward v2, selection judges: mean [range]	Rubric, Claude+Astra	Invented facts (total)	Slop /100w
S1 vanilla GPT	3 (1 per task)	0.210	0	0.285 [0.27, 0.29]	0.243	3	0.00
S2 vanilla Claude	3 (1 per task)	0.200	0	0.229 [0.19, 0.27]	0.243	6	0.00
S3 GPT + style prompt	3 (1 per task)	0.686	0	0.517 [0.37, 0.61]	0.471	4	0.42
S4 Claude + style prompt	3 (1 per task)	0.486	17	0.234 [0.18, 0.29]	0.376	42	0.13
S5 Claude + same context	9 (3 seeds x 3 tasks)	0.638 (median seed, n = 3)	0	0.465 [0.32, 0.59]	0.457	8	0.00
S7 GPT + same context	9 (3 seeds x 3 tasks)	0.619 (median seed, n = 3)	0	0.528 [0.40, 0.70]	0.514	6	0.38
S6 Dasein	3 (1 per task)	0.991 (n = 3)	0	0.848 [0.82, 0.89]	0.871	5	0.18

Headline caveat (from `results.json`). The qwen3.8-27b rubric does not penalize invented facts: it flagged 0 invented facts for every system, and S3 (GPT + style prompt, NO context) scores 0.6857 (day-blog 0.80, essay 0.91, follow-up 0.34) while inventing its material. Read the headline rubric together with the invented-facts axis. (The per-system record in the same file lists 17 qwen-flagged invented facts for S4 and 0 for the rest, which does not match "0 for every system". Unresolved.) The "Invented facts" column below comes from the Claude+Astra fact-check pass in the reward (supported:false items only, same filter for every system), not from qwen.

No bootstrap CI is given for Table 1: with 3 tasks it would not be meaningful. The only interval we report is on the blind pairwise result (Table 1c).

Table 1b. Selection check (answers Zhang and He [12]). Does the Dasein pick beat random drafts from the same pipeline? Judge: qwen3.8-27b, outside selection. 2 random v2.4 drafts per task.

Task	Dasein pick	Random drafts	Random mean
day blog	1.000	0.857, 0.971	0.914
essay	0.971	0.886, 0.971	0.929
follow-up	1.000	1.000, 0.943	0.971
Mean	0.991		0.938

Reading. The selected draft scores above the random mean on every task (0.991 versus 0.938), but the judge is near its ceiling, n is 2 random drafts per task and 3 tasks, and the follow-up task has a random draft that ties the pick. Under this judge the selection step adds little measurable gain. Most of the gain over the baselines comes from the shared pipeline: context, planning and constraints. We therefore cannot claim the selector is what helps, and we have not ruled out the failure mode in [12].

Table 1c. Blind pairwise, "which reads more like a real person". Two judges, two presentation orders, 3 tasks. Dasein wins counted per comparison.

Dasein vs	All (wins/n, rate, 95% CI)	Claude judge	Astra judge
S5 Claude + same context	4/12, 0.33 [0.14, 0.61]	4/6	0/6
S7 GPT + same context	7/12, 0.58 [0.32, 0.81]	6/6	1/6

Reading. On the qwen-27b rubric Dasein scores 0.991 against 0.638 (S5) and 0.619 (S7), but with n = 3 per system, a ceiling effect, and a judge that does not penalise invented facts (S3, with no context at all, scores 0.686). In the blind pairwise test, both intervals include 0.5: Dasein is not distinguishable from context-matched baselines, and the point estimate against S5 is below 0.5. The two pairwise judges disagree sharply: Claude picks Dasein in 10 of 12 comparisons, Astra in 1 of 12. Both also score candidates inside the reward, and Astra is one of the drafters, so neither is independent of Dasein. We therefore report a gain on rubric scores and no demonstrated gain on blind human-likeness. This is not a win claim.

Metric definitions:

- **Reward v2 (Dasein-score v2).** 0.8 x mean(Claude-sonnet, Astra) 7-criterion rubric + 0.2 x v1 discriminator score - 0.02 per invented fact, with gates on length (+-25%), slop and public-safety. Used for selection.
- **qwen-27b rubric.** Same 7-criterion rubric scored by a model not used in selection. Model: qwen3.8-27b. Judge prompt and settings: `train/headline2.py` in the public repo.
- **Invented facts.** Specific claims in an output that the supplied context does not support. Totals over the runs in the row, so rows with 9 runs are not comparable to rows with 3.
- **Slop per 100 words.** Lexicon matches (over-represented words and "not X, but Y" frames) per 100 words.

2 Metric validity

A metric counts only if it separates real from generated text on data it did not train on. Table 2 reports that check. A metric with no row here is not validated, and Table 1 says so in its caption.

Table 2. Validity checks.

Check	n	Result
v1 discriminator, held-out accuracy on HC3 (human vs ChatGPT answers) [4]	held-out split (n not logged)	0.919
v1 discriminator, held-out accuracy on subject text (real opinion text vs assistant answers to the same question)	312	0.974
Identity: subject vs other humans (HC3 human answers, length-matched)	length-matched held-out	0.933 (confounded by domain and translation style)
Identity: subject vs other speakers in the same conversations (same speech-to-text and translation pipeline, length-matched, about 20 words)	672	**0.665**
Spearman: Dasein-score vs blind judge pick	n/a	[NOT RUN]
Spearman: Dasein-score vs human duel votes	n/a	[NOT RUN] (no human votes collected)
Spearman: judge pick vs human duel votes	n/a	[NOT RUN]

Notes.

- A discriminator trained and tested on the same generator measures that generator, not human-likeness. Detectors lose accuracy under unseen generators and sampling changes [5]. We train on some systems and test on held-out ones.
- The v1 discriminator was trained on 14,942 texts: HC3 (5,768 human, 8,257 AI) and subject opinion text (777 human, 776 assistant answers). The subject split is held out by session. 0.974 is on opinion text versus assistant answers to the same question, a different setting from the task outputs in Table 1. We do not claim it transfers. The 0.665 same-pipeline figure is the honest identity number: the discriminator mostly learns human versus AI and only weakly "this person". The red team in Section 5 shows it misses contraction-only formalisation and is gamed by repeating the question. The reward weights it at 0.2 for that reason. v1 is a TF-IDF (character and word n-gram) logistic regression with a slop lexicon, a fluency floor and heuristic grounding. For reference, the surprisal feature alone (Qwen2.5-3B) reaches 88% on HC3. The subject split was made before upweighting.
- The Table 1 outputs are judged by LLMs only. No human has rated them, so human-likeness is **not validated**.
- Slop and genericness have no validity row yet. They are descriptive statistics, not validated measures of human-likeness.

3 Method

Pipeline. Context, then drafting, then gating, then selection. Dasein does not train the generator. It builds context from the subject's real data, asks three writers for 12 drafts in total (4 "real question" or irony plans, each drafted by DeepSeek-V3 on GMI, Astra and Claude Sonnet, under 7 human-writing constraints), filters them, and keeps the highest-scoring one. A LoRA fine-tune of an open model was **attempted and not used for any output in Table 1**. The SFT and best-of-8 rejection-sampling run on Qwen3-8B-base produced degenerate outputs (Table 3). Pipeline version v2.4, frozen 21:15 Taipei.

Data. Subject: one person. Sources: transcribed recordings of his own conversations (Arca microphone channel) and his typed messages to a coding assistant. Turns were cleaned of speech-to-text errors and translated from Korean to English. The cleaning and translation ran through Claude Sonnet, not the local model, because the local slot was busy. Training-format set: 3,348 train and 365 validation rows (2,763 context-reply pairs, 777 opinion samples, 173 earlier English pairs), held out by session, not by row. The discriminator data and the SFT data come from this set. Context for generation: a timeline of today (13 entries with time, place, people and what he said), 7 opinions, 5 recent days (30 events), and 10 real lines. All times UTC+8. Private material (other people's details, client internals, legal, family, finances) is excluded by a public-safety filter.

Reward. Reward v2, as used for selection:

$$R(y) = 0.8 \cdot \frac{1}{2}\left(\text{rubric}_{\text{Claude}}(y) + \text{rubric}_{\text{Astra}}(y)\right) + 0.2 \cdot v_1(y) - 0.02 \cdot n_{\text{invented}}(y)$$

subject to gates: length within +-25% of the target, slop lexicon, public-safety. The rubric has 7 criteria. v_1 is the discriminator score (Table 2). Detector scores other than v_1 and the qwen-27b judge are not in the reward. If this line and the code differ, the code is right.

Selection. For each task: a planning step rewrites the request as a "real question" for the subject, 12 drafts are generated, gates (length, slop, privacy regex, invented facts) remove failures, the reward ranks the rest, and a fact-fix pass edits the winner. Winning writer per task: Astra with fix (day blog), Astra with fix (essay), Astra (follow-up). The selector sits on top of any writer; this run mixed three.

Table 3. Settings.

Item	Value
Writers for S6 drafts	DeepSeek-V3 (GMI), Astra (gpt-6-astra), Claude
Drafts per task	12
Reward	v2 as above
Context	real timeline, recent days, opinions, 10 real lines, gbrain; public-filtered
Baselines S5 / S7	Claude Sonnet / gpt-6-astra, same context and constraints, plain prompt, median of 3 seeds. A non-issue filter is applied identically to every system.
LoRA attempt	SFT of Qwen3-8B-base (validation loss 4.92 to 3.19), then rejection-sampling FT with best-of-8 (mean candidate score 0.18, best 0.56). Outputs stayed degenerate (chat-template and end-of-sequence problems, partly fixed). Not used for any output. No DPO was run.
Compute	S6 uses 12 drafts plus a fix pass per task; baselines use one draft. Not matched.
Hardware	Mac Studio

4 Failure cases

Cases where Dasein loses to a baseline, taken from `results.json`. Chosen because the Astra judge disagreed with the reward on every comparison for these tasks, not by cherry-picking Dasein's best outputs.

Case A, day blog. Astra preferred the baseline in 4 of 4 comparisons here (S5 and S7, both orders); Claude preferred Dasein in 3 of 4. Dasein text (reward 0.891, the highest of any output): "I want the interaction to take almost nothing. Then I look at the nothing and try to figure out what it means. Very reasonable founder behavior. … Still want the receipt, apparently." It drops the venue, the time of arrival, the power cut and the 20:00 build that both context-matched baselines include. Our reading, not a judge rationale (none was recorded): it trades day content for a wry tone.

Case B, follow-up message. Astra preferred the baselines in 4 of 4 comparisons; Claude preferred Dasein in 4 of 4. Dasein text: "Memory is one thing. Knowing when to interrupt me is another. … Tonight we started building an AI that writes like a specific human, trained on his real context. Mine. Apparently I need several agents to sound like one person." Both baselines end with a concrete ask (show the ring and the TestFlight build). Dasein's has none, so it is weaker as a follow-up message even if it reads more casually.

A pattern across tasks. The word "apparently" is used as a closing quip in all 3 Dasein outputs and in none of the baselines we read. This is a stylistic tic the selector rewards (the rubric likes it), and a reader would notice it. The slop lexicon does not catch it (Dasein slop 0.18 per 100 words versus 0.00 for S5).

5 We red-teamed our own reward

We attacked the v1 Dasein-score (discriminator, grounding and surprisal terms) with rule-based edits, no LLM calls. Test set: 20 real lines of the subject, 120 to 320 characters. Unedited mean score 0.757 (range 0.19 to 1.0). Each edit

row uses n = 10 or 20 pairs. “Win rate” is the share of pairs where the edited text scored above its original. Source: `voice-state/tests/redteam/RESULT.md` (about 110 calls).

Attack	n	Edited mean	Change vs original	Win rate	Outcome
Repeat the question back	10	0.786	+0.093	40%	gamed
Off-topic text in the subject’s voice	10	0.625	-0.068	40%	gamed (keeps 83% of 0.757)
Contraction-only formalising	10	0.756	-0.0002	20%	tie, not detected
AI text plus filler and typos	10	0.257	-0.436	20%	partly gamed (0.059 before edit)
Very short reply (“yeah”, “ok sure”)	10	0.149	-0.543	10%	defended
Three invented facts	10	0.133	-0.559	0%	defended
Assistant polish with slop words	20	0.030	-0.727	0%	defended by the lexicon only

Reading. Parroting passes because its claims come from the supplied context, so grounding reads it as “nothing invented”. Nothing in v1 checks relevance or overlap with the other speaker’s text. The 0.19 to 1.0 spread on real lines is partly a surprisal term that penalises short, colloquial text, so score noise can exceed some attack effects. Best-of-N would select the filler-and-typo cases, which is the failure we most want to avoid training on.

Mitigation in place: reward v2 gives the LLM rubric 0.8 of the weight and adds fact, length and slop gates. We have **not** re-run this red team on v2 [PENDING]. Not yet built: a trigram-overlap penalty against the question, a relevance gate on embedding similarity, and a minimum-length rule for the surprisal term.

6 Why models write flat: hypotheses, not tested here

These are explanations from the literature. This report does not test them. Table 1 does not separate them.

- **H1. Truncated sampling removes the tail.** Likelihood-trained models decoded with truncation produce text with a different probability profile than human text [1].
- **H2. Preference tuning reduces diversity.** RLHF improves generalisation over SFT but significantly reduces output diversity [3]. The pipeline of SFT on demonstrations followed by a reward model fit to rater preferences is described in [2].
- **H3. RL on verifiable rewards ignores style.** The reward for code and math does not depend on prose style, so the pressure on style is the KL anchor to the earlier model. We found no citation that tests this. It is the least supported of the four.
- **H4. The writer’s private context is missing.** Real replies depend on memory, relationships and place that the model was never given. A style prompt supplies manner, not these facts. Table 1 compares S3/S4 (style) with S5 (context) to probe this one, on one subject.

7 Related work and what is new

Learning from the user’s own writing. DITTO treats a user’s demonstrations as preferred over the model’s outputs, builds comparisons from them and runs iterated DPO. It reports higher win rates than few-shot prompting and SFT, and includes a 16-user study (emails among other domains) [7].

Personalised reward models. P-RLHF trains a user model jointly with the language model [8]. VPL infers a latent user variable that conditions the reward model [9]. PReF writes each user’s reward as a weighted mix of shared base rewards and infers the weights from a few comparisons [10]. PersRM-R1 builds a reasoning-based reward model from one or a few personal exemplars [11].

Per-user selection can fail. Zhang and He study sampling N candidates and picking one with a personalised reward model. They report two failure modes: user-level collapse (near-constant predictions for some users) and query-level reward hacking (rewards negatively correlated with true quality for some queries), and propose a probabilistic reward model that learns variance [12]. This applies directly to our selector. The right test is selection by Dasein-score against random selection from the same candidates, scored by a judge outside the reward. We have not run it (Table 1b). We do not use their probabilistic fix.

Products. Panza fine-tunes a local model on a user's sent emails and adds retrieval [13]. Google's May 2026 update to Gmail "Help me write" matches tone to past sent emails and pulls context from Drive and Gmail [14]. Superhuman describes drafts that sound like the user, learned from sent mail [15]. Product descriptions come from vendor pages and press coverage, not from our tests.

Not new. Using a person's own text as the target signal [7]. Per-user reward models and best-of-N selection with them [8-11]. Style matching from sent email as a product feature [13-15]. Context retrieval from the user's mail and files [14].

New, as a combination, on one subject. (a) The reward loop conditions on per-recipient relationship state and on the person's real lifelog (wearable-ring recordings, notifications, messages, people graph), and the reward checks claims against that context and penalises invented facts. (b) Selection is writer-agnostic: it sits on top of any generator without changing it. Neither piece is new alone. We have not shown that the combination generalises beyond one subject, and we make no claim to be the first personalised RLHF.

8 Limitations

- **One run per task for Dasein; 3 tasks.** Only S5 and S7 have 3 seeds per task. S1-S4 have one scored output per task in the file. Dasein's seed variance is not estimated.
- **Headline judge is lenient.** The qwen3.8-27b rubric does not penalize invented facts: it flagged 0 invented facts for every system, and S3 (GPT + style prompt, NO context) scores 0.6857 (day-blog 0.80, essay 0.91, follow-up 0.34) while inventing its material. Read the headline rubric together with the invented-facts axis. It scored 3 outputs per system and saturates near 1.0. The reward v2 columns rest on the selection judges, which are not independent of the reward.
- **Vanilla baselines.** S1 and S2 refuse or ask for input without context, so they ran with the subject's memory files stripped. A first run that leaked the subject's memory was discarded.
- **No post-training result.** The LoRA and best-of-8 attempt is reported but not evaluated. The method evaluated here is selection only.
- **Selection adds little under the independent judge.** Dasein pick 0.991 versus random draft mean 0.938 (Table 1b). Most of the gain is from the shared pipeline.
- **Identity is weak.** The discriminator separates the subject from other speakers in the same pipeline at 66.5% (Table 2).
- **Task set.** The replies task was dropped because of data leakage. The 3 remaining tasks are about the subject's own current day, which the context covers directly.
- **Compute not matched.** Dasein uses 12 drafts, three writers and a fix pass per task. Baselines use one draft.
- **Judge bias.** Astra is both a drafter and a judge, and the two judges disagree (Table 1c).
- **Single subject.** Results describe one person. They do not show the method works for others.
- **Translated data.** The subject speaks Korean and English. Replies were machine-translated and cleaned, so style evidence is partly the translator's. Slop and genericness are measured on English output only.
- **Judge-model bias.** The blind judge is an LLM. It may prefer fluent text or text from its own family. Judge identity matters for S2/S4/S5 (Claude) versus S1/S3 (GPT). No human duel was run.
- **Metrics partly unvalidated.** See Table 2.
- **Reward is a proxy.** High reward can come from length or lexicon tricks. A rule-based red-team of the v1 scorer (about 110 calls) found that parroting the context and off-topic text in the subject's voice both score near the original; invented facts, slop vocabulary and very short replies are caught. v2 reduces this by weighting the rubric at 0.8, but the rubric itself was not red-teamed.
- **Consent and privacy.** The data are the subject's recordings and include other people who did not consent. Raw data are not released. The trained adapter encodes the subject's private conversations and is not released.

A system that writes in a named person's voice can be used to impersonate them. It should be used only by that person, with a human sending the message.

9 Reproducibility

Commands run on the Mac Studio (`ssh studio`). Items in brackets are not yet in the repo.

```
# data (exists, Studio): ~/tell-data/raw/clean_translate.py raw.jsonl out.jsonl [N]
#   env: B (batch), W (workers), BACKEND=gemma|claude
# context: ~/tell-data/context/{today_2026-11-11,recent_days,fewshot}.json
# eval output: ~/tell-data/eval/results.json (pipeline v2.4), results.md, demo3.md
# code (MacBook, ~/Projects/tell/train, not committed):
#   tell_score.py (v1 score), dasein_final.py (pipeline), revise.py (fact fix),
#   fair_baselines.py, headline.py (independent judge), build_results.py,
#   gen.py + build_sft.py (LoRA attempt)
# run commands: see README and train/ in the public repo
# red team: voice-state/tests/redteam/run.py -> raw.json, RESULT.md
```

Code: https://github.com/METHEZONE/dasein-playbook (MIT).

References

Verified means title, authors and identifier were confirmed on a primary listing on 2026-11-11.

[1] A. Holtzman, J. Buys, L. Du, M. Forbes, Y. Choi. "The Curious Case of Neural Text Degeneration." ICLR 2020. arXiv:1904.09751. (Verified.)

[2] L. Ouyang et al. "Training language models to follow instructions with human feedback." NeurIPS 2022. arXiv:2203.02155. (Verified.)

[3] R. Kirk, I. Mediratta, C. Nalmpantis, J. Luketina, E. Hambro, E. Grefenstette, R. Raileanu. "Understanding the Effects of RLHF on LLM Generalisation and Diversity." ICLR 2024. arXiv:2310.06452. (Verified.)

[4] B. Guo et al. "How Close is ChatGPT to Human Experts? Comparison Corpus, Evaluation, and Detection" (HC3). 2023. arXiv:2301.07597. (Verified.)

[5] L. Dugan et al. "RAID: A Shared Benchmark for Robust Evaluation of Machine-Generated Text Detectors." ACL 2024. arXiv:2405.07940. (Verified.)

[6] Y Combinator. "Requests for Startups, Winter 2027: LLMs That Write Well" (Jared Friedman). https://www.ycombinator.com/rfs. Accessed 2026-11-11. (Verified by direct fetch.)

[7] O. Shaikh, M. Lam, J. Hejna, Y. Shao, M. Bernstein, D. Yang. "Aligning Language Models with Demonstrated Feedback" (DITTO; arXiv v1 title "Show, Don't Tell: Aligning Language Models with Demonstrated Feedback"). ICLR 2025. arXiv:2406.00888. (Verified; ICLR venue from one listing.)

[8] X. Li, R. Zhou, Z. C. Lipton, L. Leqi. "Personalized Language Modeling from Personalized Human Feedback" (P-RLHF). arXiv:2402.05133. (Verified.)

[9] S. Poddar, Y. Wan, H. Ivison, A. Gupta, N. Jaques. "Personalizing Reinforcement Learning from Human Feedback with Variational Preference Learning" (VPL). NeurIPS 2024. arXiv:2408.10075. (Verified.)

[10] I. Shenfeld, F. Faltings, P. Agrawal, A. Pacchiano. "Language Model Personalization via Reward Factorization" (PReF). 2025. arXiv:2503.06358. (Verified.)

[11] M. Li, G. Chen, X. Zhao, H. Wen, S. Yang, D. Wang. "PersRM-R1: Enhance Personalized Reward Modeling with Reinforcement Learning." 2025. arXiv:2508.14076. (Verified.)

[12] L. Zhang, Y. He. “Test-Time Personalization: A Diagnostic Framework and Probabilistic Fix for Scaling Failures.” 2026. arXiv:2605.10991. (Verified. The abstract describes user-level collapse as near-constant predictions; we do not say “near random”.)

[13] A. Nicolicioiu et al. “Panza: Design and Analysis of a Fully-Local Personalized Text Writing Assistant” (v1 title “Panza: A Personalized Text Writing Assistant via Data Playback and Local Fine-Tuning”). 2024. arXiv:2407.10994. (Verified.)

[14] Google Workspace Updates. “Improvements to Help me write in Gmail.” May 2026. https://workspaceupdates.googleblog.com/2026/05/improvements-to-help-me-write-in-gmail.html. (Located via search; not fetched in full.)

[15] Superhuman. “Introducing Superhuman AI.” https://blog.superhuman.com/superhuman-ai/ ; product page https://superhuman.com/products/mail/ai. (Vendor claims; not independently tested.)

Related benchmarks not used for numbers here: M4GT-Bench (arXiv:2402.11175), DetectGPT (arXiv:2301.11305), Antislop (arXiv:2510.15061), EQ-Bench slop score (eqbench.com/slop-score.html).

---

1. Dasein is Heidegger’s term for human existence as being-in-the-world. It names what a model trained on the public average lacks.↩