



The Adapter Learned the Voice, Not the Typos

Fine-tuning an 8B model to write as one person in a day

Paper: PT-R-2026-038 v1.0

Author: PureTensor Ltd

Date: 6 September 2026

Status: Published

Abstract

Our agents draft a share of the mail that goes out under the operator's name. Recipients notice when a message reads as machine-written, and they ignore it or resent it. Style rules (no em dashes, a list of banned phrases) treated the symptoms. On 6 September 2026 we built pureQUILL instead: a LoRA adapter on Qwen3-8B that rewrites an agent's draft into the operator's own voice, with every fact in the draft locked so the rewrite cannot change it. The corpus, the evaluation stack, four trained adapters and a live service were built in that one day, and every number in this paper was measured that day.

The adapter learned the voice. A frozen DeBERTa authorship classifier, trained before any adapter existed, labelled 34% of the untuned model's rewrites as the operator's, 81% of the first four-GPU adapter's, and 89% of the operator's real messages. Facts survived: 97.8% of drafts kept every locked span on the first pass, rising to 100% after a corpus clean-up, and the live service changed no locked span in 200 held-out drafts.

The adapter did not learn the typos. A 50-step probe showed the operator's misspellings were learnable (10 of 10 reproduced), but the full adapter's typo recall was 0.27 against 0.25 for simply copying the input, and on words it had not seen misspelled it was no better than copying (0.15). The typos now come from a deterministic layer seeded with 685 misspellings mined from the corpus.

Its main failure was copying. Greedy decoding returned 37.4% of held-out drafts unchanged, and 24.9% of the drafts whose original differs from the input. No single decoding setting passed every gate. A targeted ladder (greedy first, then a no-copy profile only when the output copies the input), a rebuilt training set that removes near-identical pairs, and measured register layers brought copying on those drafts to 13.9% in live serving, at a 95th-percentile latency of 9.8 seconds.

After a first run on one GPU, each adapter trained in under 20 minutes on four RTX PRO 6000 Blackwell GPUs across two workstations. The blind test that decides whether the voice is good enough, 30 pairs judged by the operator, had been built but not yet sat when this paper was written. Section 13 states what these numbers do not show.

1. Why a Rewriter, and Why Not a Style Guide



An agent writing in a person's name has two jobs that pull against each other. It must say exactly what the person would commit to (the right date, the right figure, the right name), and it must sound like that person rather than like a language model. Prompt rules address the second job one tell at a time: ban the em dash and the model reaches for a semicolon; ban "I hope this finds you well" and it finds another opener. The person's actual habits (short messages, lowercase sentence starts, particular contractions, particular misspellings) are never written down, so no rule set reaches them.

We split the two jobs. The agent drafts the content as it always has. A second, small model rewrites the draft into the operator's voice and is not allowed to touch the content: every fact is replaced by a sentinel before the rewrite and restored after it, and a rewrite that loses a sentinel is retried, then returned to the human unfiltered and flagged. The rewriter only runs on mail in the operator's voice. Mail the agents send in their own name never passes through it, and nothing it produces is sent without the operator's approval.

We set the success criterion before building anything. In a two-alternative blind test the operator must pick their own real message out of a matched pair no more than 18 times in 30 (chance is 15), and a held-out authorship classifier must label at least 80% of the filtered output as the operator's.

2. The Corpus: Enough Real Text, Barely

The first risk was whether enough of the operator's own typing existed. It did, but not much. After quote-chain stripping, signature removal, secret scrubbing (three rounds, gitleaks clean) and filters for pasted text, tool output and alerts, the corpus held 3,874 training rows: 1,448 emails and 2,426 chat messages, the chat being prompts typed to coding agents and messages to our own agents. A further 683 rows were held out, and 174 dictated messages and 81 non-English messages were excluded because they are a different register. The corpus stays on our own storage and never leaves the fleet.

From the same text we mined a typo matrix: 685 misspellings the operator actually makes. A word counts as a misspelling only if it is rarer in the corpus than its correction, which keeps British spellings and command-line binary names out of the matrix.

Later in the day the operator spotted that mail written by our own agents and sent under the operator's name had entered the corpus. We profiled it: mail from 2025 onwards carried em dashes in 18 to 23% of rows and agent tells in up to 25%, and its typo density had collapsed. A new filter dropped 733 rows (599 by the era in which an agent ran the account, 72 by the frozen classifier, 51 by em dash, 11 by explicit tells). The email register fell from 1,448 rows to 982. Every model in Sections 8 onwards was rebuilt on the cleaned corpus.

3. Locking the Facts

A rewriter that improves the voice and changes a date is worse than no rewriter. Before the rewrite, regular expressions and an entity list of 64 names replace numbers, dates, addresses, links and names with sentinels of the form `[[n]]`; after it, the service checks that the multiset of sentinels is unchanged, retries up to three times, and checks faithfulness (ROUGE-L of at least 0.35 and embedding cosine similarity of at least 0.81, thresholds frozen from real pairs, whose median similarity was 0.96 against 0.40 for shuffled pairs). Anything that fails is returned to the human as the original draft, flagged. We used no constrained decoding.

We probed the sentinel format on the untuned model first. Of 50 drafts, 82% kept every `[[n]]` token and 90% kept every `<<n>>` token, both below our 95% bar. We kept `[[n]]` and moved the gate to the trained adapter, on the expectation that training with sentinels in every example would teach the model to carry them. It did (Section 6).



4. Training Pairs: From Agent Draft Back to the Original

There is no parallel corpus of "agent draft" and "what the operator would have written". We made one in reverse. Our resident agent model rewrote each real message into neutral, normalised prose with the same facts, and the training pair runs from that normalised draft back to the original. It built 3,633 pairs of 3,772 requested at 32 requests in flight; 139 (3.7%) were rejected because the normaliser lost a sentinel. Stock machine-writing phrases on the normalised side were rare (7.8% of pairs carried one of the phrases we screen for), within the bar we set, so the inputs read as plain prose rather than as a caricature of machine writing.

A second probe tested whether typos are learnable at all. The operator's misspellings tokenise into 1.86 times as many tokens as their corrections (2.44 against 1.32 on average), so a model must learn them as unusual token sequences. In a 50-step micro fine-tune on misspelling examples, the model reproduced 10 of 10.

5. A Judge Frozen Before Any Adapter Existed

An adapter tuned against a classifier will learn to satisfy the classifier. We therefore trained the authorship judge first and froze it. It is DeBERTa-v3-base, trained to separate the operator's original messages from their normalised versions on a committed split (5,086 training, 1,090 validation, 1,090 test). Its test AUROC was 0.913 and its F1 0.836, short of our 0.85 F1 bar; three runs plateaued at a validation AUROC of 0.917, so we recorded a dated waiver rather than tune it further. Scrubbing every typo from the test set lowered its AUROC by only 0.0016, so it judges style and does not simply detect misspellings. After the corpus clean-up we retrained it once on the new split (AUROC 0.909, F1 0.844) and froze that version for the later adapters.

Each adapter is scored on held-out pairs by four numbers: the share of rewrites the classifier labels as the operator's, the first-pass sentinel survival, the faithfulness pass rate, and typo precision and recall against the misspellings in the real originals. Two reference points bound the classifier score: the normalised inputs themselves (24% labelled as the operator's) and the real originals (89%).

6. Training: One GPU, Then Four

The recipe was a response-only LoRA (rank 16, alpha 32, dropout 0.05) on all attention and MLP projections of Qwen3-8B in BF16, with thinking disabled, trained with TRL and PEFT in a pinned environment. The prompt carries a register tag (email or chat) and the locked draft; the loss falls only on the locked original. The first run, v1, trained on one GPU: 954 steps over 3 epochs at a learning rate of 1e-5, in 37.8 minutes at a peak of 24.1 GB.

The operator's response to a one-GPU run on a four-GPU fleet was immediate, and the launchers were rebuilt for two nodes of two GPUs each, with torchrun over our 200G fabric (NCCL over sockets, peer-to-peer and InfiniBand transports disabled) and evaluation sharded across all four cards. Two of those cards were also serving our resident agent model, which we resized for the training window and restored afterwards. The first resize attempt took the resident model down for eight minutes: we had not checked that the reduced memory budget, less the weights, still covered the KV cache at the configured context length. The second attempt did that arithmetic first.

Adapter v2 trained on all four GPUs: 1,590 steps over 5 epochs at a learning rate of 2e-5, effective batch 8, maximum sequence 1,024 tokens, in 18.4 minutes at 18.1 GB per rank. We interrupted it at step 50 and resumed it, to prove that recovery works before depending on it.



Validation loss reached its minimum at epoch 3 (0.565) and rose to 0.584 by epoch 5, so every later adapter was selected on validation loss rather than taken at the last epoch.

Held-out, 545 pairs	Untuned model	Adapter v1	Adapter v2	Real originals
Labelled as the operator's	33.9%	72.3%	80.7%	88.6%
First-pass sentinel survival	73.4%	98.0%	97.8%	n/a
Faithfulness pass	71.4%	99.1%	97.6%	n/a
Typo recall (precision)	0.24 (0.78)	0.26 (1.00)	0.27 (0.97)	1.00

Training moved the classifier score from 34% to 81% of rewrites, against 89% for the operator's real messages, and it taught the model to carry sentinels (73% to 98%). The typo-scrubbed score was identical to the plain one (80.7%), so the gain is style, not misspelling.

7. What the Adapter Learned: The Voice, Not the Typos

The typo row in the table above is the finding of the day. Copying the normalised input unchanged keeps 25% of the reference misspellings, because the normaliser leaves some in place. The adapter reached 27%. On misspellings of words it had not seen misspelled in training, its recall was 0.15, exactly the copy baseline. The adapter preserved the typos it was given and added almost none of its own. The micro-probe of Section 4 had shown the typos are learnable in isolation; in a 2,543-example style corpus where most words are spelled correctly, the loss rewards the style and barely notices the misspellings.

A rule-based injector seeded with the typo matrix reached a recall of 0.35 at a precision of 0.47. The decision was to split the voice into ingredients: the adapter carries register, phrasing and rhythm, and a deterministic layer adds misspellings from the operator's own matrix at the measured rate for the register (0.12 per 100 words in email, 0.70 in chat). The blind test decides whether that layer stays on.

8. The Corpus Clean-Up and Adapter v3

After the clean-up of Section 2, the pairs, the classifier and the adapter were rebuilt end to end. Adapter v3 trained on all four GPUs (2,238 training and 480 validation examples, 14 minutes, epoch 3 selected on validation loss). On the new held-out set of 478 pairs it scored 86.4% on the retrained classifier, against 91.8% for the real originals and 26.4% for the normalised inputs; first-pass sentinel survival was 100%, and faithfulness 96.4%. The numbers are not comparable with v2's: the test set and the classifier both changed.

9. The Real Failure: The Adapter Copies

Reading v3's output by hand showed something no aggregate score had: on formal drafts the adapter often returned the input unchanged. A test mail came back with one phrase altered. We wrote a copy-through metric. Under greedy decoding, 37.4% of held-out rewrites were identical to the input and 64.2% were near copies (ROUGE-L above 0.95).

Part of that is legitimate. In 89 of the 478 held-out pairs (18.6%) the operator's real message equals its normalised input, because a short informal message has nothing to normalise; copying is the correct answer there. We therefore report copying on the 389 transformable



pairs, where the original differs from the input. On those, greedy v3 returned 24.9% unchanged and 50.3% as near copies.

The cause sits in the data. Many training pairs had near-identical sides, so the cheapest way to lower the loss was to copy, and greedy decoding takes the cheapest path. The classifier did not catch it, because a copied short message still reads as the operator's; a lone authorship classifier passes near copies.

10. A Decode Sweep With No Winner

We first tried to fix copying at decode time, without retraining, across all four GPUs on the 478 held-out pairs with adapter v3.

Decoding	Identical (transformable)	Near copy (transformable)	Labelled operator	Sentinel first pass	Faithfulness
Greedy	24.9%	50.3%	85.8%	100.0%	96.7%
Contrastive, beta 0.5	6.9%	12.4%	95.8%	81.6%	88.3%
Contrastive, beta 1.0	13.4%	20.6%	91.6%	49.6%	89.7%
Repetition penalty 1.1	10.0%	18.2%	95.8%	99.4%	94.1%
Repetition penalty 1.2	2.8%	5.5%	98.5%	97.7%	78.0%
Repetition penalty 1.3	1.5%	2.7%	98.7%	93.9%	47.5%

Contrastive decoding against the untuned model cut copying sharply and broke the sentinels: survival fell to 82%, then to under 50% at higher strength. A repetition penalty traded copying against faithfulness: at 1.3 almost nothing was copied and fewer than half the rewrites kept the meaning. No single setting passed every gate.

The classifier column is a warning. At a repetition penalty of 1.2 it labelled 98.5% of rewrites as the operator's, more than the 91.8% it gives the operator's real messages, while 22% of those rewrites failed faithfulness. A score above the ceiling means the judge is being satisfied, not that the voice is right.

We kept greedy as the default and added a ladder in the service: if the greedy output copies the input or fails a check, retry with a no-copy profile (repetition penalty 1.1, temperature 0.3, min-p 0.05), then with temperature 0.7; an output still identical after three attempts is returned flagged as copied. The extra cost falls only on the drafts greedy copies.

11. Rebuilding the Pairs: Adapter v4

The data fix followed. Of 3,196 cleaned pairs, 983 whose sides differed by an edit ratio under 0.15 were dropped and 2,213 kept. To restore volume, our resident model paraphrased each kept original and the pair was set to restore it (2,049 kept, 164 rejected), for 4,262 transformation-forcing pairs. Adapter v4 trained on all four GPUs in 18.8 minutes and was selected at epoch 2 (validation loss 0.829).

Held-out, 478 pairs	v3 greedy	v4 greedy	v4 no-copy profile
Identical, transformable	24.9%	21.3%	6.7%
Near copy, transformable	50.3%	38.5%	12.7%



Held-out, 478 pairs	v3 greedy	v4 greedy	v4 no-copy profile
Labelled as the operator's	85.8%	86.6%	98.5%
Sentinel first pass	100.0%	99.8%	98.7%
Faithfulness pass	96.7%	95.4%	88.9%

The rebuild lowered copying under greedy decoding by a modest amount. Most of the remaining gain comes from the no-copy profile, at a cost of seven points of faithfulness, which is why the service applies it only after greedy has copied.

12. The Live Service

Two vLLM replicas of Qwen3-8B with the adapter sit behind a small service that runs the whole chain: lock the facts, rewrite, verify the sentinels, retry up the ladder, check faithfulness, restore the facts, then apply two measured layers. The register layer was measured on the cleaned corpus: the operator's email has a median of 14 words, a median sentence of 4 words, contractions in 27% of the places they could occur, a greeting in 10% of messages, a sign-off in 7%, and lowercase sentence starts in 30%. Chat is similar in length, with longer sentences, more contractions and no greetings. The typo layer applies the matrix at the register's rate.

On 200 held-out drafts at 32 requests in flight, the live service returned 13.9% of transformable drafts unchanged and 16.5% as near copies; 23.5% of all drafts came back flagged as copied (many of them correctly, being drafts with nothing to change). First-pass sentinel survival was 98.5% and no restored output carried a changed fact. In 7.5% of drafts the service failed closed and returned the original draft, flagged, to the human. The 95th-percentile latency was 9.8 seconds. Drafts of 120 words or more start the ladder at the no-copy profile, because greedy near-copied almost every long draft and pushed the 300-word latency to 21.5 seconds.

The rewriter is wired into the send path. Mail in the operator's voice cannot leave without a logged marker from the service, or an explicit bypass with a stated reason, which is itself logged. The agents' own mail never passes through it.

13. What These Numbers Will Not Carry

The voice has not yet been judged by the one person who can judge it. The 30-pair blind sheet (18 chat, 12 email, each pair matched and counterbalanced, built from the live pipeline over held-out messages of at least 30 words) was ready at the end of the day; the operator had not sat it. Until it is sat, the success criterion of Section 1 is unmet and unrefuted.

The classifier is the only automatic judge of voice here, and Section 10 shows its limits: it can be pushed above the score of real messages, and it passes near copies. A style-embedding distance beside it is the next addition. The corpus is small (982 emails after clean-up), so every held-out set is a few hundred pairs, and all of it comes from one person; nothing here shows the method transfers to another writer. The training pairs come from a language model normalising real messages, so the adapter learns to undo that model's normalisation, which may differ from the drafts our agents actually write. Every run used a single seed, and each comparison is one run against one run.

What the day does support is narrower. A small LoRA adapter on an 8B model, trained in under 20 minutes on workstation GPUs, moved an independent authorship judge from 34% to over 80% while carrying locked facts through intact, and it did not learn the person's misspellings, which a deterministic layer from a mined matrix now supplies. Its dominant failure was copying, which came from the data, and no decoding setting alone fixed it without breaking either the facts or the meaning.



14. What We Changed and What Comes Next

The service now separates the ingredients of a voice: the adapter for phrasing and rhythm, a measured register layer for length and form, a mined typo matrix for misspellings, and locked spans for facts. Copy-through is a gate in every evaluation, adapters are selected on validation loss, training and evaluation run on all four GPUs by default, and the send path refuses operator-voice mail that has not passed through the rewriter.

Next come the blind sitting, a length target that compresses long formal drafts toward the operator's median of 14 words, a style-embedding judge beside the classifier, and preference optimisation only if the blind test fails.

PureTensor operates a sovereign AI infrastructure fleet (on-premises GPU compute, storage, and serving) run day-to-day by autonomous agents under human direction. This paper is PT-R-2026-038. All measurements were taken on 6 September 2026. Raw artefacts, including every training summary, evaluation run and decode sweep, are retained; the corpus is private correspondence and is not published.