Writing copilots that augment, not replace: state of the art (November 2026)

*Deep-research report for **write-on**, claude-researched 2026-11-10 23:00 IST. Project page: inventcures.github.io/write-on. Code: github.com/inventcures/write-on.*
Inspiration and credit. This project starts from Jason Fried's Write_On, shown in his post and demo video on X (30 September 2026). His idea of "alternative control" at the word, sentence and paragraph level, with a way to dim text back and stash it nearby, opened up the question this report asks: what would a writing tool look like that amplifies thinking and acts as a true thought partner, without writing for you? All credit for that interface idea belongs to him.
How this was made. Several research agents searched academic papers (arXiv, CHI, UIST, ICLR, EMNLP, ICML), commercial products, open-source projects and prior art in non-linear writing interfaces. Their notes were then synthesized and fact-checked into the report below. Figures are as reported by the cited sources; where sources disagree, the report says so.

Summary: Point the learned voice at nudges, not drafts

As of November 2026, the market has solved half of the planned product and gotten the other half backwards. Learning a person's voice from samples is now common: Grammarly, Every's Spiral, Sudowrite and Gemini in Docs all do it in some form. **Every commercial tool verified uses that learned voice to generate or rewrite text** ("Rewrite in my voice", 500–1,000-word drafts). None uses it to give feedback or nudges. Meanwhile, the research has grown much stronger that generation is the problem. Writing with LLMs flattens voice and stance while users stay just as satisfied. It shifts opinions without people noticing. It lowers ownership. Even "preserve my voice" prompts remove only about a third of the drift. Jason Fried's Write_On, posted on X on **30 September 2026** (not in November), supplies the interface half: "alternative control" keeps several live versions under each word, sentence and paragraph, dims text instead of deleting it, and stashes leftovers nearby. Open-source clones of it appeared within weeks. The opening for the planned product is to join the two halves. A Write_On-style editor where AI output can only enter as one more alternative, a ghost, or a question in the margin. A layered style model (editable style card, then style embeddings, then an optional local adapter) that is used mainly to *notice* when the writer drifts from their own patterns, and only secondarily to suggest. The main open gaps are these. No longitudinal field study of a learning writing assistant exists. No benchmark predicts one author's next paragraph opening. Nobody has tested a feedback-only assistant against a generative one in the same experiment.

Write_On turns alternatives into a place in the text, not a point in time

Fried's own post describes the tool as "'alternative control' at the word, sentence, and paragraph

level, plus a way to dim stuff back, and stash stuff near by". He says it is "Not version control, but alternative control". He also says he had wanted it for "maybe a decade", that it was not commercially viable as a 37signals product, and that he built it over a weekend with Claude's help (Digg summary; original post at x.com/jasonfried/status/2105403067793584590, which none of the researchers could fetch). The cleanest way to see the idea is a small example.

Concrete example: alternative control vs. version control. Suppose the sentence is *"We shipped it fast."* With **version control**, the unit is the whole document over time: draft-v1.md says "fast", draft-v2.md says "in a weekend", and you compare snapshots afterwards. With **alternative control**, the unit is the span, and its options live in the text right now: We shipped it [fast ▸ in a weekend ▸ before we were ready]. You hover over "fast", press ↓, and the sentence reads "We shipped it in a weekend." Press ↓ again and it reads "before we were ready." Nothing is lost, and nothing is "v2". The options are a *paradigm* (the set of words that could fill one slot), not a history. **Ghosting** ("dim stuff back") is a third state between keep and delete: "before we were ready" can stay on the page, faded. **Overflow** ("stash stuff near by") is a place for a good line that has no slot yet.

What Write_On does in detail comes from people who rebuilt it from Fried's three clips, not from Fried himself. The **write-on-replica** README reports that alternatives "stay under the text" and are flipped through with the arrow keys. Ghosted text fades and can be revived. An Overflow panel stashes selections and can send them back. A "Lab" marks weak or long sentences, filler and typos and offers 10/20/30/50% trims. It "marks and trims but never rewrites", and cut text is ghosted. The file format is Markdown with alternatives kept in HTML comments (joeysim/write-on-replica). **Undertext** (MIT; Tauri plus ProseMirror) gives exact keys: ↑/↓ cycles a hovered word or sentence, but only after the mouse has moved, so nothing cycles by accident. ⌘⇧D forks a paragraph. ⌘/ ghosts. ⌘⇧X sends the selection to Overflow, stored after an <!-- overflow --> marker. In its Lab, Claude may **only delete** when trimming, and typo fixes are stored as alternatives so they can be undone (klipstack/undertext). The same README lists known bugs: partial ghosts are lost on save, and ~ and { get escaped. This is an early warning that putting rich annotations through plain Markdown and back is fragile.

Three claims in the notes need flagging. First, the date. Decoding the post's snowflake ID gives **2026-09-30 21:03 UTC**, which I re-ran and confirmed. That matches Digg's "first seen ~40 days ago" as of 10 November (Digg). The project brief's "November 2026" framing is the date of this research, not of the post. Second, **whether the original uses AI at all is unconfirmed.** Undertext credits the original with an AI lab that "trims without rewriting." The replica says the clips do not describe any AI feature in the original, yet it also describes trim levels taking about two seconds. That timing suggests a model call, but nobody has confirmed it (undertext vs. write-on-replica). The safe reading is that fade-style trimming, ghosting and stashing are in the original, and the AI's role is unknown. Third, availability. Fried says in the first clip that he hopes to release it as free open source. It had not been released as of October 2026 (write-on-replica). No Hacker News, 37signals or HEY World post about it turned up. Everything the spec borrows from "Write_On" is therefore really borrowed from the clones' readings of a video.

The pieces have a long lineage, and that lineage shows what is new. Xanadu put parallel documents side by side with visible links (Appleton on Xanadu patterns). Ink & Switch's Patchwork found Google Docs–style suggestion views "noisy and hard to scan". It argued for cheap retroactive branches, grouped undo (for example, revert every Oxford-comma fix at once) and AI bots that propose edits on labelled branches (Patchwork notebook). Linus Lee's Prism treated meaning and style as continuous, formatting-like controls decoded from sparse-autoencoder features (Prism). Maggie Appleton's "daemons" were marginal AI characters whose ignored suggestions fade away (LM Sketchbook). iA Writer's Focus Mode

dims text by cursor position, which is temporary and about attention (iA Focus Mode). Write_On's ghosting is the writer's lasting decision about meaning. No earlier system combined in-place per-span alternatives, ghosting and a stash. That combination is Write_On's contribution. The planned product's contribution would be the learning layer, which Write_On, as shown, does not have.

Commercial tools learned your voice, then aimed it at ghostwriting

Commercial voice learning falls into roughly five tiers, from cheapest to richest. They are listed below with what each does with the voice it learns.

Tier	Example (verified)	How it learns	What it does with the voice
Stated preferences	Microsoft 365 Copilot Memory	Saves preferences explicitly, or implicitly only when there is "clear intent"; view/edit/delete (Microsoft)	Shapes generated output
Editable passive profile	Grammarly (now under Superhuman)	"Automatically detects" style; separate Documents/Messages profiles; user can **remove individual style elements** (help)	On-demand "Rewrite in my voice"
Few-shot exemplars	Sudowrite Story Bible	Writer stores **3–5 strongest passages** as style anchors (Sudowrite)	Generates prose
In-document style	Gemini in Docs "Match writing style"	Uses the current doc as the reference, rolled out Apr–May 2026 (Google)	Rewrites for consistency
Fingerprint + judge loop	Spiral 4.0 (Every)	Stylometric "writing fingerprint" picks samples; **87%** on a blind-lineup eval where a judge explains misses (Every)	500–1,000-word drafts; "personalize"/"humanize" via MCP, CLI and API

Two of these ideas are worth borrowing, and both are about making the learned voice *visible and fixable*. One is Grammarly's decomposed profile with removable elements. The other is

Spiral's judge that explains *why* a generated draft gave itself away. **No vendor verified describes per-user fine-tuning.** All personalization happens at inference time. Spiral is the most sophisticated, but its eval method beyond the 87% figure is not public. Its scale claim of 5,524 style guides from 168,464 samples comes from the vendor (Every).

The "partner, not replacement" message is now mainstream, but the products' core actions still generate. Spiral v3 interviews you before drafting, to find what you are "really" trying to say (Every, Spiral v3). Then it writes three drafts. Lex's 2023 feedback feature left AI critique as a *comment* with three ideas for an intro, not as edits (Lex). Its 2025 Line Editor can "edit in your voice" but does not say how it learns that voice (Lex V2). The strongest "you own your words" features in the market are about *provenance*. iA Writer's Authorship shows AI or pasted text in rainbow colors, and iA dropped the earlier dim grey because "the dimmed grey tone was too quiet" (iA). Grammarly Authorship records whether text was typed, pasted, suggested or AI-generated, and can replay how a document was written (Grammarly).

iA's finding is a useful counterpoint to Write_On. Dimming is the right treatment for the writer's *own* text that they are unsure about. It is the wrong treatment for machine text, which needs a louder mark. Commercial alternatives are also *short-lived*. A carousel, three drafts, or an "apply" button disappears once you choose. None of the tools checked keeps rejected alternatives for later comparison (Every; Lex; Google). Price cuts (Spiral down to $15/mo) and bundling into Gemini, Copilot and Grammarly/Superhuman suggest that generic "rewrite in my voice" is becoming a commodity (Every). Coverage has gaps. Jasper, Writer.com, Notion AI, Apple Writing Tools, ProWritingAid, Wordtune, and ChatGPT/Claude Projects were not verified for 2026, so this report makes no claims about them.

Generated text homogenizes writers, and they do not notice

The strongest argument for "assist, never replace" is now quantitative, and it shows one mechanism with several effects.

Concrete example: homogenization (illustrative, not from a study). Three people write a first line about public speaking: A: *"I threw up before my first talk."* B: *"My hands shook so much I dropped the clicker."* C: *"I don't remember the talk, only the silence after."* Ask a generic assistant to "improve" each, and you tend to get three cousins of: *"Stepping onto the stage, a wave of anxiety washed over me."* Each rewrite is fluent, and each writer might be satisfied. But the "I" moved to the end, the concrete detail (vomit, clicker, silence) became an abstraction ("anxiety"), and an adjective-heavy stock image appeared. Across all three writers, the texts became more alike. The studies below measure exactly these shifts.

In "How LLMs Distort Our Written Language" (an RCT with N=100, arXiv March 2026), heavy gpt-4o-mini users rated their essays as **less in their own voice** ($t(69)=5.80$, $p<0.001$) and less creative, yet **were just as satisfied**. **Stance-neutral essays rose about 69%** relative to control. Even "grammar-only" LLM edits shifted meaning in a consistent direction, while human revisions varied. Pronouns fell about 50%, adjectives rose about 33%, and positive-emotion words rose 37–54% (arXiv 2603.18161). The note-taker flags internal inconsistencies in the paper's figures, so treat individual numbers as approximate. The direction of the effect is consistent with the work below.

"Voice Under Revision" (preprint, April 2026) had GPT-5.4, Claude Sonnet 4.6 and Gemini 3.1 Pro rewrite 300 personal narratives, for 2,700 rewrites. **All three models moved all 13 stylometric markers in the same direction**: fewer function words and first-person pronouns,

and longer words and more punctuation. Mean effect sizes under "improve" were 0.60, 1.29 and 1.45. **Prompts asking the model to preserve the writer's voice cut the effect by only 32%**, and the direction held for 85% of markers (arXiv 2604.22142). The usual workaround, draft first and then edit it into your own voice, also fails. In a pre-registered study with 81 participants, post-edited text stayed **closer to the LLM draft than to the participants' own unassisted writing**, and was less diverse. Participants still *believed* it sounded like them (arXiv 2604.24444).

The older evidence points the same way. Writing with InstructGPT, but not base GPT-3, made essays across authors more alike. **The drop came from the model's contributions, while the users' own text kept its diversity** (Padmakumar & He, ICLR 2024). That is the most actionable finding in the whole literature for this product: the harm enters with the machine's words. Writers given an opinionated GPT-3 were about **2× as likely to argue its side**. Their later attitudes shifted too, and only 10–30% noticed the bias (Jakesch et al., CHI 2023). Autocomplete pulled Indian and American writers' styles together, mainly at the expense of Indian styles (Agarwal et al., CHI 2025). GPT-4 story ideas made individual stories more novel but made stories more alike as a group (Doshi & Hauser via ScienceDaily). Note that one novelty figure in that press release appears mislabelled. Users did not feel they owned AI-generated text, personalizing it did not help, and ownership grew with how much the writer shaped the text (Draxler et al.). An MIT EEG study found the weakest brain connectivity and the lowest ability to quote one's own essay in the LLM group (Kosmyna et al.). It is a non-peer-reviewed preprint with 54 participants, then 18, and has been criticized, so it is suggestive at best.

Here is the alternative view, stated at its strongest. Doshi & Hauser found AI ideas *helped* the least creative writers most, with novelty up 10.7% and usefulness up 11.5% (ScienceDaily). Buschek et al. found that non-native English writers gained most from extra parallel suggestions (arXiv 2101.09157). So "never generate" is not obviously right for every writer. The evidence supports a narrower rule that serves both groups: **let machine text in only in small pieces, as one option among several, chosen and edited by the writer, and with its origin marked.** The voice-drift results add a second lesson. Users cannot be relied on to notice drift, so the tool has to *measure and show* it.

Style learning works in layers, and few-shot is weakest where essayists write

The notes describe three families of technique. They are easiest to understand side by side. **Concrete example: style card vs. style embedding vs. LoRA.** Imagine a writer whose blog posts read like Fried's: short declaratives, lots of "we", paragraphs that open with a blunt claim. A **style card** is a written brief you could hand a human editor: *"Opens ~40% of paragraphs with a short claim ('Here's the thing.'). Uses 'we' over 'I'. Avoids semicolons. Average sentence ~11 words. Never uses 'leverage'."* You can read it, argue with it and delete a line. It cannot *measure* a new sentence precisely. A **style embedding** (e.g., LUAR's 512-number author vector) works like a sommelier's nose. Given two candidate rewrites, it can say candidate A is closer to this writer than candidate B. It can say nothing about *why*, and it cannot write anything. A **LoRA adapter** is like an apprentice who has absorbed the writer's habits by copying 500 of their paragraphs. It can produce text that sounds like them, but it cannot explain itself, it needs a lot of data, and pushed too hard it gets worse at writing well. The planned product needs all three: the card for **trust**, the embedding for **ranking and drift alarms**, and the adapter (later,

optionally) for **stronger suggestions**.
Retrieval of the user's own past writing is the cheapest and most studied method, but its benchmarks mostly measure content overlap with ROUGE, not voice. LongLaMP's reported gains do not reconcile: about 30% ROUGE-1 average in the conclusion against "5.7% to 128%" in the introduction. More retrieved items sometimes hurt, with some settings peaking at k=1 (arXiv 2407.11016). The more practical recipe for the style card keeps the LLM frozen and **learns a natural-language preference description**. CIPHER infers it from the user's edits and reduced cumulative edit distance by about 31% (summaries) and about 73% (email) with simulated users, at roughly a third of the token cost of baselines. Preferences that ignored context often did *worse* than no learning at all (arXiv 2404.15269). PROSE infers the description from writing samples and adds **consistency verification**: any trait not supported across several samples is dropped. Its per-component accuracy was 1.07 vs 0.09 for CIPHER-5 (arXiv 2505.23815). Its headline "33% better than CIPHER" evidently uses a different measure, so cite the component figure. Both were tested on simulated users. GhostWriter is the closest human study. It learned style implicitly, gave users explicit moments to refine it, and found that more control increased agency (N=18) (arXiv 2402.08855).
On imitation quality, the sources disagree, and the disagreement matters for this product. Jemama & Kumar found few-shot prompting fooled an authorship verifier **91–100%** of the time and concluded that "prompting strategy, not model size" governs imitation (arXiv 2509.24930). Wang et al. tested more than 400 everyday authors and found 5-shot verification at about 95–97% on news and email but **only about 17–21% on blogs** and 50–66% on Reddit (EMNLP 2025 Findings). Popp et al. found few-shot prompting *below* zero-shot on style, 28.1% vs 31.0%, with three abstracts on Qwen3-4B (arXiv 2610.03163). The likely explanation is genre and metric. Formal, templated writing is easy to imitate from examples. Idiosyncratic personal writing is not. **Personal essay and blog writing, the planned product's core genre, is exactly where few-shot is weakest.** Wang et al. add two levers the spec can use. Choosing exemplars by *topic* similarity hurt attribution badly (Enron 69.3 to 36.0). Giving the model the start of the writer's actual text was the strongest lever (blogs "human" rate up from 9.0% to 21.7%). So: retrieve exemplars by *function* (other paragraph openings by this writer), not by topic, and continue from the writer's own partial sentence.
Stronger stylizers trade style against quality. StyleTunedLM's per-author LoRA beat 5-shot prompting on a style classifier, 0.879 vs 0.693. The authors were Gutenberg writers who were likely in the pre-training data, with about 80k tokens each (arXiv 2409.04574). Popp et al. found plain LoRA had the best style (49.8%) but its **quality-preference rate collapsed to 38.1%**. A *hypernetwork* that predicts LoRA weights from three samples was the only method on the Pareto frontier (arXiv 2610.03163). That is one domain and one model, published 2 October 2026, so it is too new to build on. TinyStyler, an 800M model conditioned on authorship embeddings from about 16 samples, beat GPT-4 on automatic authorship-transfer metrics at about a third of the latency. Reranking five candidates by embedding similarity raised its Joint score from 0.40 to 0.43. Human ratings were lower than automatic ones, and the embedding "may miss rare stylistic choices", which are often the distinctive ones (arXiv 2406.15586).
InMyStyle built training pairs from a single user's own writing by having small models "AI-ify" the user's paragraphs, then training a rank-8 QLoRA to map them back. With 487 paragraphs, "AI-ness" fell about 22%. Training took 9.5–68 minutes on one laptop GPU, with no gain above 0.5B parameters (arXiv 2607.29238). It had only one user and no few-shot baseline. Among style embeddings, the 2026 STEB benchmark found **no single model wins across tasks**: LUAR-CRUD, STAR and LUAR-MUD are nearly tied on operational tasks, and StyleDistance leads on definitional ones (STEB).

One more mechanism explains *why* generic suggestions sound generic. Verbalized Sampling attributes mode collapse partly to **typicality bias** in human preference data. Asking the model for "k responses with their probabilities" raised creative diversity 1.6–2.1× and human-rated quality by 25.7% (ICML 2026). AI imitations also have much lower perplexity than human text (about 15–16 vs 29.5), even when they fool verifiers (arXiv 2509.24930). The practical upshot: the "AI voice" is, roughly, the *most probable* continuation. A personal copilot should prefer candidates that are likely under the writer's own style and unlikely under the generic model.

Choosing and reflecting beat prompting and receiving

HCI studies of co-writing agree on the interaction shape, though most are small single sessions. In "Choice Over Control" (N=129), writers accepted **0.74 of suggestions when offered three, vs 0.55 for one**. 75 of the participants preferred multiple suggestions and 23 preferred one. Single suggestions were rated more distracting, and explicit instructions helped some writers and annoyed others (arXiv 2303.03199). ABScribe stored variations of a segment in place in the text, and writers compared them by hovering. Workload dropped (d=1.20) and perceptions of revising improved (d=2.41), though N was only 12 (arXiv 2310.00117). This is the closest peer-reviewed evidence for Write_On's interaction. Luminate generates *dimensions* first (tone, point of view) and then options along them, to counter chat's "rapid convergence on a limited set of ideas" (arXiv 2310.12953). Multiple parallel suggestions helped ideation but slowed input (arXiv 2101.09157). Predictive text produced shorter, more predictable captions (Arnold et al.). This is a direct argument against always-on ghost-text completion.

Non-generative help also has evidence behind it. Continuously updated paragraph summaries in the margin gave writers "an outside view" of their own text and helped them revise scope (UIST 2022, N=4 and N=8) (arXiv 2208.09323). Feedback from personas the writer defined inspired revisions. The main complaint was that the feedback was **wordy and vague** (arXiv 2309.10433). In Rambler, 10 of 12 users preferred it over ChatGPT. The predictable features (keywords, semantic zoom) were used most, while unpredictable LLM merge and split were used less because they lowered trust (arXiv 2401.10838). Sarkar argues AI should be a "provocateur" that critiques and surfaces assumptions, but warns that constant criticism frustrates users, so how often provocations appear must be tuned (CACM). Robin Sloan wanted the machine to make writing "harder," not easier (Sloan). Paul Graham: "If you're thinking without writing, you only think you're thinking" (Graham).

Taken together, the evidence suggests what makes an editor's advice trusted. The Lish–Carver history is a natural experiment on this. On several pages of "Neighbors", "fewer than half of Carver's words were left standing", and Carver credited Lish with his reputation. Stephen King called the same kind of editing "a total re-write" and "a cheat" (Wikipedia, Gordon Lish). The difference between trusted editing and overreach comes down to four things. Changes stay visible and reversible. The editor cuts and questions rather than substituting their own voice. The editor knows the author's body of work. The author keeps the final say. Each maps onto a product rule: alternatives and ghosts, delete-only trims, questions as margin comments, a style model built from the author's own corpus, and grouped revert.

What to borrow, what to avoid, what to build

	Item	Source / evidence
Borrow	Span-anchored alternatives cycled in place; ghosting; Overflow stash; delete-only AI trims	Write_On clones (undertext, replica); ABScribe
Borrow	~3 alternatives on demand, not one ghost completion	Choice Over Control
Borrow	Editable, decomposed voice profile	Grammarly; GhostWriter
Borrow	Judge that explains why a candidate doesn't sound like you	Spiral 4.0
Borrow	Loud provenance marking for machine text	iA Authorship; Grammarly Authorship
Borrow	Grouped revert per AI pass; no accept/reject chrome	Patchwork
Borrow	Margin "reflect back" summaries; short persona questions	Dang et al.; Benharrak et al.
Avoid	Full drafts, then "edit into your voice"	Baumler et al.
Avoid	Relying on "preserve my voice" prompts	van Nuenen
Avoid	Always-on predictive ghost text	Arnold et al.
Avoid	Topic-similar exemplar retrieval	Wang et al.
Avoid	Ungated per-user LoRA	Popp et al.
Avoid	Dim grey for machine text (reserve dimming for the writer's own ghosts)	iA
Avoid	Noisy suggestion-mode UI; vague, wordy critique; too many daemons	Patchwork; Benharrak; Appleton
Build (new)	Voice used as *feedback*: "you usually open with a concrete scene; this opens with an abstraction"	No product found doing this (commercial notes)

	Item	Source / evidence
Build	Drift meter against the writer's own baseline	Motivated by Abdulhai et al., van Nuenen
Build	"Your own past phrasings" offered next to generated ones	Inference from Padmakumar & He
Build	Opening-move picker by discourse function, learned per author	No benchmark exists (style notes)
Build	Persistent rejected alternatives as preference data	No commercial tool keeps them

Several **gaps remain open, and the spec should treat them as risks, not settled facts.** No multi-week field study of a personal co-writing assistant that adapts over time was found. No benchmark exists for predicting one author's paragraph openings or next argumentative move. No experiment compares a feedback-only assistant with a generative one on originality and ownership. No study tells how often a style card should refresh, or how people feel about editing one. No study compares few-shot, retrieval, LoRA and a style card on one essayist's corpus with judges who know the author. And none of the researchers saw Fried's actual clips. Building the product would itself help fill several of these gaps.

Implications for the spec

Design principles

#	Principle	Concrete rule	Evidence
P1	**Machine words enter only as options**	AI output is appended as one more alternative under a span, never swapped in; the writer's version stays active until they switch	Diversity loss comes from model-contributed text (Padmakumar & He); undertext's "without removing your current text"
P2	**AI may delete or flag, not rewrite**	Trim = deletions only, shown as ghosts; Lab marks sentences, doesn't replace them	Write_On clones; Lish–Carver overreach line
P3	**Small scope, several choices**	Default 3 alternatives, word/sentence/opening level; no paragraph-or-larger	Choice Over Control; Arnold et al.

#	Principle	Concrete rule	Evidence
		generation; no ghost-text autocomplete by default	
P4	**Make the learned voice legible and vetoable**	Style card with traits, each linked to supporting quotes from the writer's own samples; delete/edit any trait	PROSE; Grammarly; GhostWriter
P5	**Voice drives feedback first, generation second**	Primary output of the style model is nudges ("this opening is unlike yours") and reranking, not text	Market gap; Baumler et al.
P6	**Show drift; don't trust the user to notice**	Per-document drift meter vs. own baseline	Satisfied-but-less-own -voice (Abdulhai et al.); unnoticed influence (Jakesch)
P7	**Provenance is always visible**	Machine-originated text marked loudly (not dimmed); dimming reserved for the writer's ghosts; focus-dim visually distinct from ghosts	iA; iA Focus
P8	**Summoned, rationed, fading**	AI acts on a keystroke; unsolicited nudges capped per session and adaptive; ignored nudges fade	Sarkar; Appleton; Sloan
P9	**Critique is short and specific**	One quoted span + one question; no paragraphs of advice	Benharrak et al.
P10	**Writer sets off-limits zones**	e.g. "never suggest ideas, only flag clarity"	Gero et al.
P11	**Plain files, user-owned data**	Markdown + comments; style card, vectors and adapters exportable and	Clones' format; LoRA = personal data (open-source notes)

#	Principle	Concrete rule	Evidence
		deletable; local by default	

Style-learning architecture, layered by data available

A frozen, strong general LLM sits under every layer. Each layer adds to the ones before it and does not replace them.

Layer	Data threshold	Components	What the user sees	Evidence
L0 Cold	0 samples	Local grammar/clarity (Harper WASM); rule-based prose lint (Vale-style); constrained LLM ops (trim-delete, mark weak sentences, 3 alternatives via verbalized sampling)	Write_On editor working with no style learning	Harper; Verbalized Sampling
L1 Card	3–5 samples pasted	PROSE-style induction → **style card** (consistency-verified traits with supporting quotes); exemplar retrieval **by function** (paragraph openings, closings) not topic; continuation from writer's own partial text	Editable card; alternatives conditioned on card + own exemplars; "your past openings like this"	PROSE; Wang et al. (≈5 exemplars saturate; topic retrieval hurts)
L2 Vector	~16+ samples / tens of docs, plus logged edits	Authorship embedding centroid (LUAR or StyleDistance; benchmark both since none wins)	Drift meter; "sounds like you" ranking; nudges based on profile features	TinyStyler (16 samples; rerank +0.03); STEB; CIPHER

Layer	Data threshold	Components	What the user sees	Evidence
		for **reranking** candidates and **drift detection**; CIPHER-style context-conditioned preferences learned from edits when edit cost exceeds a threshold; interpretable stylometric profile (function words, pronoun rate, sentence length; e.g. faststylometry)		
L3 Adapter (optional)	~500 paragraphs + hundreds of accept/reject events	On-device 0.5–3B LoRA trained on InMyStyle "AI-shadow → original" pairs, with name/entity masking; periodic DPO/KTO on (rejected suggestion → writer's final text); **quality gate** on every retrain; watch hypernetwork-predicted adapters as a future replacement	Stronger, less "AI" alternatives; opt-in	InMyStyle; StyleTunedLM; Popp et al.

Three cross-cutting rules apply. First, rank candidates with a contrastive score: high likelihood under the writer's style signal, penalized for generic typicality. This targets the low-perplexity "AI voice" (arXiv 2509.24930). Second, version the style card so inferred traits can be compared and rolled back. Third, model paragraph openings as a per-author distribution over discourse moves (anecdote, contrarian claim, question, number). Offer 3–5 moves at a paragraph boundary, and do not predict a single path. This last rule is a design hypothesis with no benchmark behind it yet.

Editor building blocks

Need	Recommendation	Notes / license
Editor core	ProseMirror (directly, or via Tiptap open core / BlockNote MPL-2.0 core)	Undertext already proves ProseMirror works for this; avoid BlockNote XL (GPL-3.0) and Tiptap Tracked Changes (paid) unless licensed (BlockNote; Tiptap)
Alternatives data model	A "variant" mark: span id → N variants, 1 active, each tagged origin = human, ai or lab, timestamp, and accept/reject history; extend the insertion/deletion/modification mark model of prosemirror-suggest-changes	That library's license was not stated, so verify it (README)
Ghosts / overflow	Ghost = mark with persistence; Overflow = document-level stash with back-links to origin span	Define serialization formally to avoid undertext's round-trip bugs
File format	Markdown; ghosts as ~~text~~, variants in <!-- write_on … --> comments, stash after <!-- overflow -->	Degrades gracefully in other tools (undertext)
History & attribution	Yjs 14 changesets/attribution (as BlockNote is building), feeding the preference log; grouped revert per AI pass	FOSDEM 2026
Always-on nudges	Harper (Apache-2.0, Rust/WASM, offline); learned habits exported as Vale-style rules so they can be inspected	Harper; Vale
Local models	Word/phrase alternatives: 2–4B dense (Gemma 4 E4B, Qwen3.5-4B) with cached prefix; paragraph-level nudges: Qwen3.6-35B-A3B MoE; prefer built-in MTP over draft-model speculative decoding for creative text	Classmethod guide; RunAIHome. Gemma 4's Apache-2.0 status comes from a secondary guide, so verify it; avoid Nemotron's proprietary license
Reference code	Wordcraft (operation-centric	Wordcraft; Novel

Need	Recommendation	Notes / license
	co-writing) over Novel (autocomplete-centric)	

The latency target of under 300 ms for word alternatives is an inference from the notes, not a measured figure. The notes found no time-to-first-token benchmarks for small models on consumer hardware, so measure early.

Evaluation metrics

Metric	What it catches	How	Evidence
Voice fidelity	Suggestions that don't sound like the writer	Authorship verification of accepted suggestions vs. held-out own text (LUAR or TF-IDF+embedding verifier), per genre	Jemama & Kumar; Wang et al.
Drift	Homogenization over a document and over months	13-marker stylometric profile (function words, 1st-person pronouns, word length, punctuation) and lexical JSD vs. own baseline; human-revision baseline ≈0.2–0.3 JSD	van Nuenen; Abdulhai et al.
Stance strength	Drift toward neutral, hedged positions	Share of stance-neutral paragraphs vs. writer's baseline	Abdulhai et al. (+69%)
AI-ness	Generic "AI voice"	Perplexity gap; LLM-judge AI-ness; judge-explains-why on misses	InMyStyle; Spiral
Machine share	Ghostwriting creep	% of final characters with origin: ai still active	Provenance tools (Grammarly)
Agency-weighted acceptance	One-click acceptance without thought	Accept rate *and* edit-after-accept rate; ownership grows with influence	Draxler et al.; Choice Over Control

Metric	What it catches	How	Evidence
Ownership & recall	Cognitive offloading	Periodic self-rated ownership; can the writer quote their own piece?	Kosmyna et al. (suggestive)
Nudge quality	Annoying or vague critique	Dismiss rate, act-on rate, nudges/session; target falling dismissals as the system learns	Sarkar; Benharrak
Learning curve	Whether "learns more the more you use it" is true	All of the above on a temporal split (recent writing of a known user)	LongLaMP temporal setting
Cross-writer diversity (cohort)	Product-level homogenization	Pairwise similarity of users' texts over time vs. a no-AI cohort	Padmakumar & He; Doshi & Hauser

Conclusion

The field's center of gravity is moving from *capability* (can the model write like me?) to *custody* (whose words, whose thinking, and can I see the difference?). The commercial answer so far has been to perfect imitation and sell it as a draft. The research answer is that a perfect imitation draft still anchors the writer to the machine, and the writer does not notice. That changes what "learns your style" should mean in this product. The style model's main job is to be an attentive reader of the writer's corpus: noticing, ranking and warning, closer to a long-time editor's memory than to a ghostwriter's pen. Generation stays in a small, marked, optional box. The product treats this as a guardrail. It is also how the product differs from everything on the market: tools that sell imitation compete on cheaper, better imitation, and a tool that keeps the writer the author does not.

Two non-obvious bets follow. First, keeping *rejected* alternatives in the document is a UX feature and also the product's training data. It is the cheapest source of personal preference pairs, and no competitor keeps them. Second, the first proprietary asset is likely to be the drift and opening-move evaluation harness rather than a model. No public benchmark covers either, and they are what would let the product *prove* it leaves people sounding more like themselves after six months, not less.

Check your understanding

1. **Alt vs. version control.** Take the sentence "Meetings are toxic." Write it as (a) two version-control snapshots and (b) one alternative-control span with three options. Which view lets you compare the options *while reading the paragraph*, and why does that matter for a writer settling on "just the right way to say something"?
2. **Ghost vs. focus-dim vs. provenance.** A paragraph contains a sentence you're unsure

about, the sentence your cursor is in, and a phrase the AI suggested that you accepted. Which visual treatment should each get under the principles above, and what does iA's "grey was too quiet" finding imply?

3. **Pick the layer.** A new user pastes four blog posts. Which style-learning layer applies? What would you retrieve as exemplars for a paragraph-opening suggestion, and what would you deliberately *not* retrieve by (cite the Enron result)?
4. **Card, vector, or adapter?** For each task, name the best component and say why: (a) explaining to the user why a suggestion was ranked first; (b) choosing between five candidate rewrites; (c) producing candidates that are less "AI-sounding" after a year of use; (d) letting the user delete a wrong trait such as "uses lots of semicolons".
5. **Spot the homogenization.** Compute, roughly, the first-person pronoun rate and average word length for "I threw up before my first talk" vs. "Stepping onto the stage, a wave of anxiety washed over me." Which markers from van Nuenen's study moved, and in which direction?
6. **Steelman the opposition.** Using Doshi & Hauser and Buschek et al., argue *for* letting less-experienced or non-native writers receive more generated text. Then propose a spec rule that serves them without breaking principle P1.
7. **Reconcile the conflict.** One study says few-shot imitation fools verifiers 91–100% of the time, and another says 17–21% on blogs. Give two reasons they can both be true. Which number should guide a product for essayists?
8. **Design a metric.** The writer accepts 80% of suggestions. Why is that not proof the product works, and which two other metrics from the table would you check before believing it?