VPEVOLVE: A SELF-EVOLVING VIRTUAL PROCESS ENGINEER FOR COMPUTATIONAL LITHOGRAPHY

Anonymous authors
Paper under double-blind review

ABSTRACT

Optical proximity correction (OPC) recipes grow as engineers add local rules to repair newly discovered lithography hotspots. Each correction can interact with existing rules, while lessons from commercial-tool trials remain scattered across code and logs. VPEVOLVE combines a Virtual Process Engineer (VPE) harness with a Skill Bank of measured engineering experience. The harness equips a frozen language model with process manuals, layout analysis, recipe editing, and commercial-tool evaluation. The actor proposes changes to the global parameters, local targeted rules, or diagnostic trials. After each evaluation, an LLM reflector and curator turn the measured response into evidence-linked judgments. The actor retrieves them before its next trial. Feasible improvements update the retained recipe; every measured trial informs the Skill Bank that guides the next edit. The model weights remain fixed. On a FreePDK45-derived benchmark with ten commercial-tool evaluations per case, VPEVOLVE reduces the mean per-case maximum edge placement error from 18.294 to 5.361 nm on Poly and from 22.052 to 15.692 nm on Metal1. Every final recipe satisfies the predefined quality constraints and improves the maximum error by at least 0.1 nm. Project website: `https://vpevolve.github.io/vpevolve/`.

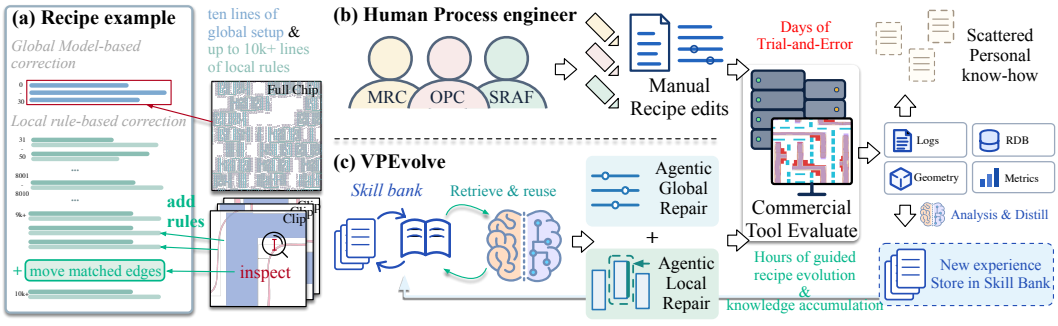

Figure 1: **From recipe development to VPEvolve.** (a) OPC recipes combine global settings with local rules for specific patterns. (b) Engineers diagnose printing errors and revise recipes, accumulating experience across edits and trial records. (c) VPEvolve connects recipe optimization with a Skill Bank that turns measured outcomes into guidance for subsequent experiments.

1 INTRODUCTION

In semiconductor manufacturing, optical proximity effects distort the shapes printed on a wafer (Mack, 2007). Process engineers therefore develop a **recipe** for each process node and target layer to control how an optical proximity correction (OPC) tool corrects the mask. A recipe is more than a set of model parameters. Model-based correction may require only a few dozen lines of settings, while thousands or tens of thousands of lines encode local rules for specific layout patterns, as illustrated in Figure 1(a). When simulation or test-chip measurements reveal large edge placement error (EPE) or mask rule violation, engineers inspect the affected geometry, write rules to identify similar patterns, and specify how to correct them. The tool applies those corrections wherever the patterns recur. As

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

rules accumulate, reading and changing the recipe becomes time-consuming: a fix for one hotspot can interact with existing rules or create another violation. Engineers responsible for OPC, sub-resolution assist features (SRAF) and mask rule checking (MRC) must coordinate edits and repeatedly test the results, as illustrated in Figure 1(b). However, the link between what changed, why, and what happened remains scattered across engineers, code, and logs, making experience hard to reuse.

Traditional parameter-tuning methods cover only part of recipe development. Bayesian optimization can tune model-based settings chosen in advance (Snoek et al., 2012). But process engineers often add local rules after seeing new hotspots, introducing pattern matching conditions and correction values that were not in the original search space. Large language models (LLMs) can propose such edits, but recipe work also requires process knowledge and access to the layout geometry behind a measured error. The lesson of a successful edit can become misleading as the recipe changes. A local rule may fix its target while another hotspot becomes the worst, and later global edits may change the effect of the rule. If the agent only remembers that the rule worked, it can keep refining that correction after the limiting error has moved elsewhere. This motivates a **Virtual Process Engineer (VPE) harness** that gives the agent process knowledge, layout analysis, recipe editing, and commercial-tool feedback. VPEVOLVE combines this harness with an LLM-curated Skill Bank that turns trial outcomes into reusable engineering judgments (Figure 1(c)).

The VPE harness supplies process manuals for recipe commands and editing procedures. The LLM actor reads the current recipe, layout diagnostics, and previous outcomes, then chooses a global edit, a pattern-specific local correction, or a diagnostic trial. Layout tools identify the relevant edges and compile the edit; the commercial tool measures the printing and manufacturing response (Siemens Digital Industries Software, 2026). After each trial, an LLM reflector interprets local and global measurements. An LLM curator revises the Skill Bank with the conditions, supporting trials, counterexamples, and possible next experiments for each judgment. The actor retrieves relevant judgments alongside the latest feedback before deciding what to try next. Measured improvements update the retained recipe; every measured trial supplies evidence for the next edit.

Recipe development is evaluated on a FreePDK45-derived benchmark with ten commercial-tool evaluations per case. VPEVOLVE reduces the mean per-case maximum EPE from 18.294 to 5.361 nm on Poly and from 22.052 to 15.692 nm on Metal1; every endpoint meets the predefined quality constraints and improves by at least 0.1 nm. Mechanism ablations examine how interpreting spatial responses, revising experience, and guiding subsequent trials contribute to these improvements. To distinguish better recipes from more useful experience, a paired continuation study holds the recipe and observed history fixed while varying how experience is curated. Further analyses examine local actions and improvement over longer search horizons. Our contributions:

- **Recipe development as a learning problem.** The task is formulated as sequential search over global settings and growing local rules, with trial evidence retained for later decisions.
- **A Virtual Process Engineer harness.** Process manuals, layout analysis, recipe editing, and commercial-tool evaluation equip a frozen LLM to run global, local, and diagnostic trials.
- **Self-evolving engineering experience.** LLM reflection and curation distill each trial into scoped, evidence-linked skills that guide subsequent experiments.
- **Evaluation and an open testbed.** Commercial-tool experiments show lower maximum EPE than all seven baselines on both layers. Ablations, language-model comparisons, and longer runs test what drives the gain and when it persists. The FreePDK45-derived layouts and lithography simulator are released for open simulation studies.

2 PRELIMINARIES

An edit to a recipe is judged by what it prints. Given a layout and a recipe, the lithography tool generates a mask and predicts the wafer contours at different focus and exposure-dose settings, known as process corners (Mack, 2007; Yang & Ren, 2023) (Figure 2). The recipe controls optical proximity correction (OPC), which reshapes mask features to compensate for proximity effects, and places sub-resolution assist features (SRAFs) to stabilize imaging without printing on the wafer.

Recipe quality is measured from these contours. At a target-edge sample p_i, let n_i be the unit normal and $q_{i,c}$ the corresponding contour point at process corner $c \in \mathcal{C}$. With P_c the printed region and μ

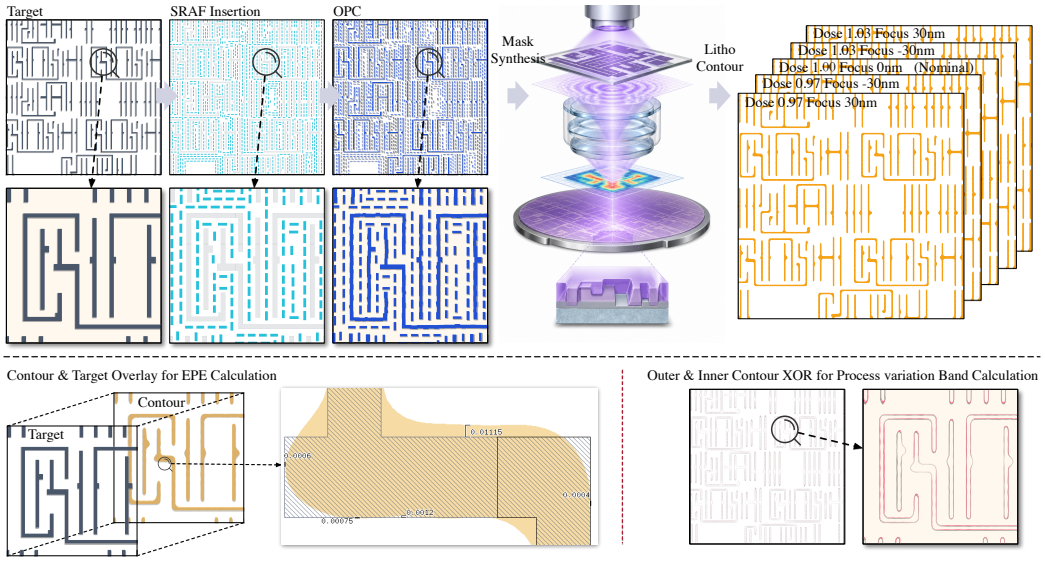

Figure 2: **Computational-lithography evaluation.** A recipe guides SRAF insertion and OPC to produce a mask, whose printed contours are simulated across focus and exposure conditions. EPE compares printed contours with target edges; PVB measures their variation across process conditions. These measurements guide subsequent recipe edits.

denoting area, define edge-placement error (EPE) and the process variation band (PVB) by

$$\mathrm{EPE}_{i,c} = \langle q_{i,c} - p_i, n_i \rangle, \qquad \mathrm{PVB} = \frac{1}{L}\mu\left[\left(\bigcup_{c\in\mathcal{C}} P_c\right) \setminus \left(\bigcap_{c\in\mathcal{C}} P_c\right)\right], \tag{1}$$

The sign of EPE indicates the direction of contour displacement. Reported Max and Mean EPE are the maximum and mean absolute errors at the fixed evaluation gauges under nominal focus and dose. Top-10 EPE averages the absolute errors of the ten highest-error, spatially separated hotspots selected by the evaluator. PVB measures variation across process corners, normalized by the target-edge length L in the scoring region, and has units of μm. Mask rule checks (MRC) assess mask manufacturability, while SRAF printability checks detect assist features that print unintentionally. These measurements determine whether a recipe can be retained and identify residual errors.

Repeated evaluation makes this an optimization problem, but the choices do not stay the same throughout recipe development. Bayesian optimization searches an expensive parameter space specified in advance (Snoek et al., 2012; Gardner et al., 2014). Learning-based lithography methods have produced mask corrections and SRAFs (Yang et al., 2018; Chen et al., 2020; Yang & Ren, 2024; Li et al., 2025). Chen et al. choose EPE measurement positions and edge fragmentation for model-based OPC with reinforcement learning, then distill the results into pattern-conditioned rules using multimodal feature labels and a decision tree (Chen et al., 2025). A recipe also grows through new local rules chosen in response to measured hotspots. Each new rule expands the available edits and can change how previous trial outcomes inform subsequent decisions.

Agents provide a way to work through this changing sequence of tool calls and edits. Methods for tool use interleave reasoning with execution and learn when to call an API (Schick et al., 2023; Yao et al., 2023); agents for science and EDA coordinate experiments and engineering tools (Liu et al., 2025; Huang et al., 2024; Zhang & Sun, 2026; Wu et al., 2024). ChipNeMo adapts language models for chip-design assistance and script generation (Liu et al., 2023).

After a trial, the measured result can inform a later decision even if the candidate recipe is rejected. Reflection, memory, and workflows provide ways to carry such information across attempts (Shinn et al., 2023; Zhao et al., 2024; Wang et al., 2025). Prior agents have stored reusable behavior as executable skills (Wang et al., 2024) or contextual strategies and playbooks (Ouyang et al., 2026; Zhang et al., 2026b); the Darwin Gödel Machine also modifies agent code (Zhang et al., 2026a). In

VPEVOLVE, the recipe and the engineering judgments used to edit it both evolve from measured commercial-tool feedback, while the agent harness stays fixed.

3 METHOD

3.1 RECIPE DEVELOPMENT AS A SEQUENTIAL DECISION PROBLEM

Each recipe trial reveals how a candidate prints an IC layout and informs the next modification. This makes recipe development a sequential decision problem. Given a layout x, an initial recipe r_0, and a budget of B optimization trials, the objective is to reduce the maximum edge-placement error while maintaining acceptable average EPE, process variation, and mask manufacturability. The process model and evaluation settings remain fixed during the search.

Actions and feedback. At step t, the agent uses the retained recipe r_t and its simulation feedback e_t to choose a global edit, a pattern-specific local correction, or a diagnostic trial. Global edits adjust recipe parameters that affect the layout as a whole; local edits introduce correction rules for selected geometric patterns where printing errors remain. A diagnostic trial evaluates a controlled recipe change to test a hypothesis about the measured response. Evaluating a candidate recipe returns both overall quality metrics and the locations of residual errors. A candidate replaces the current recipe when it satisfies the quality constraints and improves the optimization objective. Otherwise, the current recipe is retained, and the observed outcome informs the next decision.

Learning from recipe development. Each evaluation records what changed, under which conditions, and how the printed result responded. A trial record τ_t captures the outcome, and a versioned bank $\mathcal{E}_t$ maintains engineering judgments. Before proposing a modification, the actor retrieves relevant judgments. After evaluation, an LLM reflector interprets the measured response, and an LLM curator proposes a revision to the bank. The resulting loop is

$$\begin{aligned} V_t &= \mathrm{Retrieve}(\mathcal{E}_t, x, r_t, e_t), & a_t &\sim \pi^{\mathrm{act}}_{\theta,H}(\cdot \mid x, r_t, e_t, V_t, \mathcal{T}_t), \\ \widetilde{r}_t &= \mathrm{Apply}(r_t, a_t), & \widetilde{e}_t &= F(x, \widetilde{r}_t), \\ (r_{t+1}, e_{t+1}) &= \mathrm{Select}(r_t, e_t, \widetilde{r}_t, \widetilde{e}_t), & h_t &= \mathrm{Reflect}_{\pi_\theta}(\tau_t), \\ \Delta_t &= \mathrm{Curate}_{\pi_\theta}(\mathcal{E}_t, h_t, \tau_t), & \mathcal{E}_{t+1} &= \mathrm{CheckApply}(\mathcal{E}_t, \Delta_t, \tau_t). \end{aligned} \tag{2}$$

Here, H denotes the fixed agent harness, V_t retrieved experience, a_t the actor's proposed trial, and F the fixed lithography evaluator. Selection returns the candidate recipe and its feedback when it is a feasible improvement, and retains (r_t, e_t) otherwise. The trial record $\tau_t = (r_t, a_t, \widetilde{r}_t, e_t, \widetilde{e}_t)$ is retained even when the candidate recipe is rejected and appended to the trial history $\mathcal{T}_{t+1} = \mathcal{T}_t \cup \{\tau_t\}$, with $\mathcal{T}_0 = \varnothing$. The reflection h_t distinguishes measured responses from hypotheses; the curator's patch Δ_t may add, update, split, or merge judgments and suggest a subsequent experiment. The harness checks evidence references and revision structure before applying that patch. Recipe and experience can change independently: even a rejected candidate can reveal where a rule works or fails. The model weights, process model, and harness remain fixed.

3.2 VIRTUAL PROCESS ENGINEER HARNESS

VPEVOLVE retains the best feasible executable recipe and a versioned bank of engineering judgments. Its VPE harness connects an LLM actor to process guidance, layout analysis, recipe editing, and commercial-tool evaluation. An LLM reflector analyzes each measured trial, and a curator revises the actor's experience. All three roles share one frozen model (Figure 3).

The actor reads the current recipe, spatial and aggregate feedback, previous trial records, process manuals, and retrieved experience. It chooses whether to adjust global settings, apply a local displacement rule, or perform a diagnostic experiment. The process manuals supply recipe commands and engineering strategies. The actor revises its strategy as evidence arrives. Each trial uses one commercial-tool evaluation from the shared budget.

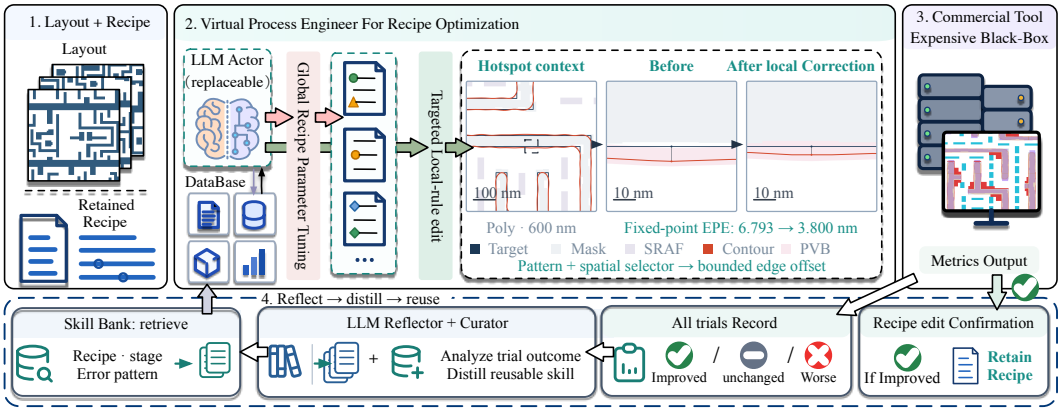

Figure 3: **Recipe and experience evolution in VPEvolve.** The actor uses the current recipe, spatial feedback, and retrieved experience to propose global changes or local corrections for commercial-tool evaluation. Feasible improvements update the retained recipe. The reflector interprets the measured response, and the curator revises the Skill Bank to guide subsequent trials; model weights remain fixed throughout both updates. The Poly example shows a local correction reducing EPE at the fixed measurement point from 6.793 to 3.800 nm.

For a local action, layout tools associate a measured EPE hotspot with its design edge and nearby geometry. Native pattern matching identifies eligible mask-edge fragments, and a spatial selector binds the proposed displacement to the diagnosed occurrence. The EPE sign and edge normal provide a starting direction; the actor tests and revises that hypothesis against the measured response. The LLM chooses the experiment; a deterministic compiler binds every edit to its exact parent recipe and checks allowed commands, geometry, and displacement limits before execution.

The commercial tool returns EPE at the original hotspot and newly worst locations, global Max/Top-10/Mean EPE, PVB, MRC, and SRAF printability checks. A candidate becomes the next recipe only if it satisfies the fixed constraints and improves the prescribed lexicographic quality ordering. Every trial is recorded together with its measurements or execution outcome and the evaluated recipe.

3.3 Self-Evolving Skill Bank

The experience mechanism adapts ACE-style reflection and curation (Zhang et al., 2026b) to spatial process feedback. It adds three connected capabilities. **Domain Response Reflection** compares the intended effect of a trial with the measured response at its original hotspot, the new worst hotspot, and the full scoring region. The LLM distinguishes no response from a repaired target followed by a new global bottleneck, and records possible side effects as hypotheses rather than established facts.

Scoped Experience Revision lets the LLM curator add, update, split, or merge an engineering judgment. A judgment links an intervention and its applicability conditions to observed effects, supporting trials, counterexamples, and unresolved questions. The curator explains changes to evidence roles. Programmatic checks verify that cited trials exist, that an intervention occurred in them, and that the versioned patch respects its declared structure. The LLM assesses applicability and causal interpretation from the measured record. For example, when a rule repairs its target but another hotspot becomes limiting, the curator can preserve evidence of local improvement while qualifying its effect on maximum EPE. A reversed response under changed recipe settings can provide counterevidence and narrow the judgment's applicability.

Experiment Guidance allows the curator to attach a strategy update and up to two next proposed experiments. Each proposal states an action, its expected local and global effects, and an observation that would falsify it. These suggestions enter the actor's next context; the actor chooses the next action. Retrieval favors judgments relevant to the current recipe, the layer, and the measured response. Original trial records remain available alongside the curated bank. The actor uses these records and distilled findings in subsequent decisions without changing model weights or harness code. Appendix D details the implementation.

Table 1: **VPEvolve versus seven baselines on Poly and Metal1.** Layer-wise EPE and PVB are case means; overall success and call counts cover the benchmark. EPE is in nm and PVB in 10^{-3} μm.

	Poly			Metal1			Overall		
Method	Max↓	Avg.↓	PVB↓	Max↓	Avg.↓	PVB↓	Max↓	SR↑	OPC / LLM
Initial recipe	18.294	0.854	7.281	22.052	0.805	5.015	20.173	—	—
Raw Actor	17.271	0.835	7.239	19.609	0.758	4.969	18.440	80%	220 / 455
ReAct	16.501	0.817	7.120	17.775	0.729	4.885	17.138	100%	220 / 524
Textual Memory	17.296	0.781	6.977	18.966	0.742	4.878	18.131	85%	220 / 1030
ReasoningBank	17.523	0.833	7.151	20.538	0.756	4.924	19.031	75%	220 / 988
ACE	16.697	0.783	6.905	19.920	0.758	4.928	18.308	95%	220 / 1833
Static Harness	17.583	0.848	7.257	19.875	0.780	5.001	18.729	75%	220 / 502
Bayesian opt.	17.909	0.827	7.300	21.773	0.778	5.010	19.841	30%	220 / 0
VPEvolve	**5.361**	0.645	7.398	**15.692**	0.690	5.021	**10.527**	100%	220 / 1295

SR: feasible max-EPE reduction ≥ 0.1 nm; all retained endpoints have MRC = 0. OPC counts exclude R0 and include endpoint repeats. Bold marks VPEvolve; Appendix F details evaluation and model-call accounting.

4 Experiments

4.1 Experimental Setup

Benchmark and evaluation. Each commercial-tool evaluation runs recipe correction and lithography simulation under the fixed process model. The benchmark has 20 windows cropped from FreePDK45-generated full-chip layouts (NC State EDA): ten Poly and ten Metal1, balanced across five density strata. Each has a 10×10 μm scoring region and a 2 μm context halo. Methods share R0, the process model, and measurement points. VPEvolve and its mechanism ablations start with an empty Skill Bank and shared engineering guidance. Each case has one seed, a ten-trial optimization budget, and a final recipe repeat. The objective minimizes maximum EPE while mean EPE and PVB remain within 102% of R0, with zero MRC and SRAF-printing violations. Ties are broken by top-ten mean EPE, mean EPE, then PVB. We report maximum and mean EPE, PVB, and success by layer; success requires completion and a feasible maximum-EPE reduction of at least 0.1 nm. Appendices A, B, and F give metric and cost details.

Baselines. We compare with six agent baselines and Bayesian optimization. Raw Actor, ReAct (Yao et al., 2023), and Static Harness (Pi Contributors, 2026) test action selection and fixed guidance. Textual Memory (Shinn et al., 2023; Zhao et al., 2024), ReasoningBank (Ouyang et al., 2026), and ACE (Zhang et al., 2026b) use reflection, retrieval, or context revision. Bayesian optimization (Snoek et al., 2012; Gardner et al., 2014) searches model-based parameters. Agent baselines share recipe-editing actions and trial budgets but vary in context and memory. These baselines cover numerical search, general-purpose agents, and memory mechanisms (Appendix C).

Implementation. VPEvolve uses the Pi agent harness (Pi Contributors, 2026) and extends ACE-style reflection and curation (Zhang et al., 2026b) with domain response reflection, scoped experience revision, and experiment guidance. Unless stated otherwise, all three roles use local Qwen3.6-27B (Qwen Team, 2026) on four NVIDIA H100 GPUs, with a 131,072-token context, 4,096 output tokens, and temperature zero; model weights remain fixed. Cross-model experiments use each evaluated model's API for all three roles; long-horizon experiments use the DeepSeek-V4.1-Flash API. Appendix D gives the inference settings.

4.2 Main Results

Table 1 tests whether VPEvolve improves recipe quality under a limited evaluation budget. It reduces mean window-maximum EPE from 18.294 to 5.361 nm on Poly and from 22.052 to 15.692 nm on Metal1, reductions of 70.7% and 28.8%. All cases achieve a feasible improvement. ReAct is the strongest agent baseline on maximum EPE, at 16.501 and 17.775 nm; VPEvolve improves on it by

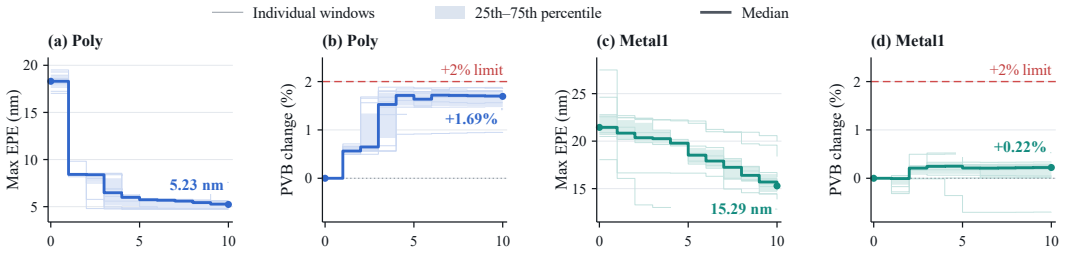

Figure 4: **Recipe evolution over ten optimization trials.** Maximum EPE and PVB change from R0 for retained recipes. Thin lines are cases; thick lines and bands show medians and interquartile ranges. Endpoint labels are medians; Table 1 reports means. The dashed line marks the +2% PVB limit. Trial zero is R0; endpoint repeats are excluded.

Table 2: **Mechanism ablation of experience evolution.** Metrics average the best feasible retained recipe for each case in each layer. EPE is in nm and PVB in 10^{-3} μm. SR is the share of assigned runs completing with a feasible maximum-EPE reduction of at least 0.1 nm.

	Poly				Metal1			
Condition	Max↓	Mean↓	PVB↓	SR↑	Max↓	Mean↓	PVB↓	SR↑
Full VPEvolve	5.361	0.645	7.398	100%	15.692	0.690	5.021	100%
Frozen Experience	8.082	0.640	7.319	90%	16.121	0.717	5.027	100%
w/o Domain Response Reflection	6.213	0.653	7.374	90%	16.280	0.742	5.023	100%
w/o Scoped Experience Revision	5.475	0.632	7.411	100%	16.334	0.717	5.025	90%
w/o Experiment Guidance	5.870	0.624	7.394	100%	16.154	0.725	5.025	90%

67.5% and 11.7%. Mean EPE also falls below every baseline, to 0.645 and 0.690 nm. VPEvolve thus improves worst-case and average accuracy within the fixed quality constraints.

Figure 4 shows how these gains develop. On Poly, median maximum EPE drops to 8.42 nm after one trial and reaches 5.23 nm after ten. Metal1 improves later, from 18.52 nm at trial five to 15.29 nm at trial ten. Refinement continues after the initial edits while PVB stays within the +2% limit.

4.3 Mechanism Ablation

Table 2 examines experience accumulation and its three components. Full VPEvolve and four variants share the model, seeds, R0, empty initial bank, engineering guidance, actions, constraints, and ten-trial budget. All variants select their actions autonomously.

Frozen Experience. This variant uses the current recipe and latest feedback with fixed guidance, without earlier trial history or online experience updates. Its maximum EPE is 8.082 nm on Poly and 16.121 nm on Metal1, versus 5.361 and 15.692 nm for Full. Accumulating trial experience improves the final recipes on both layers, with a larger maximum-EPE improvement on Poly.

Domain Response Reflection. Replacing this component with generic reflection raises maximum EPE to 6.213 and 16.280 nm. Spatial evidence remains available, but explicit guidance on interpreting hotspot-level responses and changes in overall recipe quality is removed.

Scoped Experience Revision. Without this component, factual checks remain but intervention-topic preservation and evidence-reclassification review are removed. Maximum EPE rises from 5.361 to 5.475 nm on Poly and from 15.692 to 16.334 nm on Metal1.

Experiment Guidance. Without this component, judgments remain available but explicit strategy updates and proposed trials are omitted, yielding 5.870 and 16.154 nm. These comparisons support interpreting measured responses, revising their applicability, and using them to guide subsequent experiments. Full has the lowest maximum EPE on both layers and the lowest mean EPE on Metal1. On Poly, removing revision or guidance lowers mean EPE to 0.632 or 0.624 nm from 0.645 nm, showing that worst-hotspot and average accuracy need not improve together.

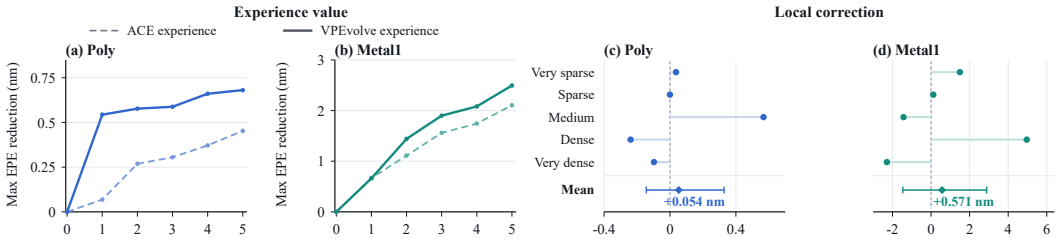

Figure 5: **Value of evolved experience and local correction.** (a–b) Mean additional maximum-EPE reduction using fixed VPEvolve (solid) or ACE (dashed) experience from a shared fifth-trial recipe and history. (c–d) Paired maximum-EPE advantage after four trials; positive values favor local actions. Points are cases, diamonds are means, and bars are 95% paired bootstrap intervals.

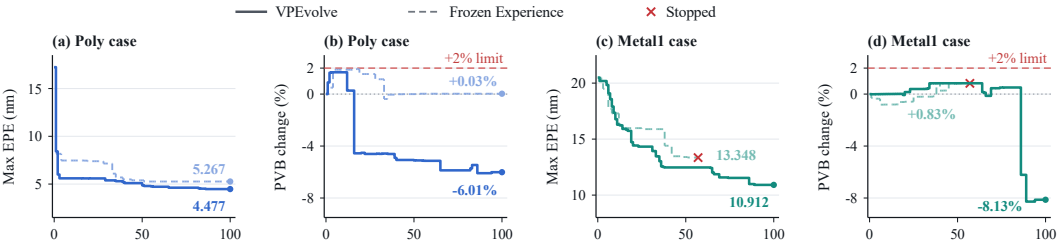

Figure 6: **Long-horizon recipe evolution.** Retained maximum EPE (a, c) and PVB change from R0 (b, d) for Poly (blue) and Metal1 (green). VPEvolve is solid; Frozen Experience is dashed. Red dashes mark the +2% PVB limit; crosses mark Frozen's termination at trial 57 on Metal1. Evaluated candidates count toward the horizon even when not retained. Endpoint repeats are excluded.

4.4 Further Analysis

4.4.1 Value of Evolved Experience

To isolate the value of experience beyond the retained recipe and raw history, each case resumes from the same fifth-trial recipe, feedback, and history, using either generic ACE experience curated from those trials or the recorded VPEvolve experience. Both banks remain fixed through five further trials with the same model, guidance, and R0 constraints. VPEvolve experience yields mean additional maximum-EPE reductions of 0.681 nm on Poly and 2.494 nm on Metal1, versus 0.453 and 2.104 nm for ACE (Figure 5a–b). Final mean EPE is nearly unchanged on Poly and lower on Metal1 (0.693 versus 0.698 nm). With recipe and history held fixed, the larger average gains on both layers show that organizing process feedback into engineering judgments can improve subsequent decisions. Appendix E reports paired differences and intervals.

4.4.2 Local Correction after Global Exploration

To test whether spatial actions complement global tuning, we select one case per density stratum and layer. Paired continuations start immediately before the first local intervention, sharing recipe, feedback, and experience. Both update experience over four trials; one permits global edits alone, the other also permits local correction. Allowing local actions lowers mean maximum EPE from 18.583 to 18.012 nm on Metal1 and from 5.238 to 5.185 nm on Poly (Figure 5c–d). It improves three of five Metal1 cases and two of five Poly cases, with one Poly tie. Mean EPE is higher with local actions (0.651 versus 0.635 nm on Poly; 0.698 versus 0.630 nm on Metal1), as is PVB, while both branches remain feasible. Local corrections provide another way to reduce residual worst errors after global exploration, with their value depending on the remaining hotspots and available global edits.

4.4.3 Long-Horizon Optimization

We extend the budget to 100 trials on one case per layer to test whether experience evolution supports sustained optimization. Full and Frozen Experience start from R0 with empty banks and shared

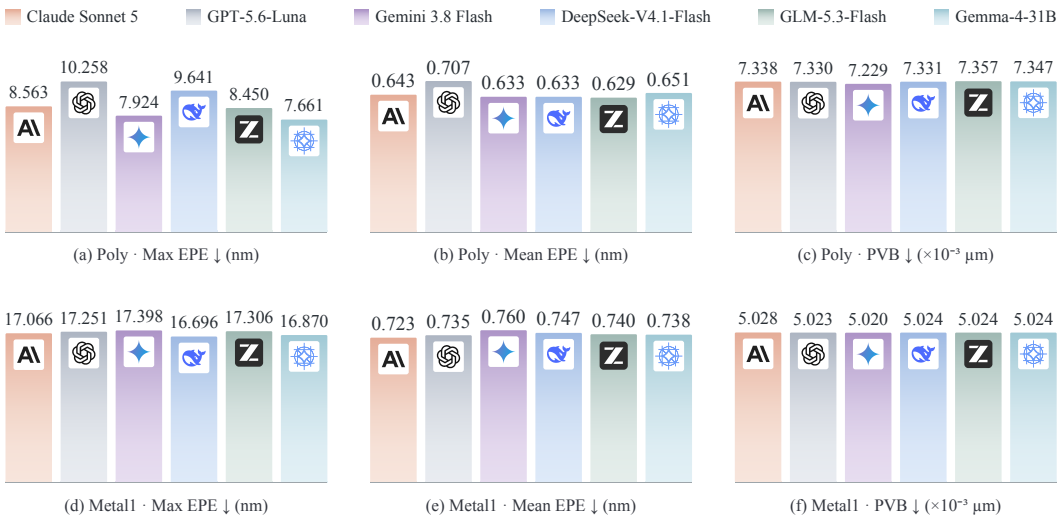

Figure 7: **VPEvolve across language models.** Each model serves as actor, reflector, and curator through its API. Bars show benchmark means for Poly (top) and Metal1 (bottom). EPE is in nm and PVB in 10^{-3} μm; lower is better.

guidance, using DeepSeek-V4.1-Flash through its API under the original quality constraints (Figure 6). On Poly, both complete 100 trials; Full reaches 4.477 nm maximum EPE versus 5.267 nm for Frozen, with mean EPE 0.639 versus 0.813 nm. PVB changes by -6.01% and $+0.03\%$, respectively.

On Metal1, Frozen repeatedly requests information and terminates after 57 trials. At that shared horizon, Full already has lower maximum EPE (12.470 versus 13.348 nm). It continues to 100 trials, reaching 10.912 nm, with mean EPE 0.529 nm and PVB 8.13% below R0. The comparison reveals both better quality at the shared horizon and further gains as Full continues. Together, these trajectories support evolving experience as a means of sustaining useful recipe refinement over longer interactions. Appendix G gives completion and recovery details.

4.4.4 Generalization across Language Models

To test whether VPEvolve depends on a particular language model, we replace all three roles together with each of six models: Claude, GPT, Gemini, DeepSeek, GLM, and Gemma, accessed through their APIs (Figure 7). All six lower mean and maximum EPE from R0. Mean maximum EPE ranges from 7.661 to 10.258 nm on Poly and 16.696 to 17.398 nm on Metal1, versus initial values of 18.294 and 22.052 nm. Gemma and DeepSeek achieve the lowest maximum EPE on Poly and Metal1; GLM and Claude lead on mean EPE. PVB ranges from 7.229 to 7.357 on Poly and 5.020 to 5.028 on Metal1, in 10^{-3} μm. The recipe and experience loop improves quality across model families, with rankings varying by layer and metric. Appendix D gives the inference settings.

5 Conclusion

Each lithography trial can improve a recipe and leave evidence for the next engineering decision. VPEvolve makes both outputs part of the same development loop. Its Virtual Process Engineer harness gives a frozen LLM process guidance, layout analysis, recipe editing, and commercial-tool evaluation; its Skill Bank uses LLM reflection and curation to turn trial feedback into scoped, evidence-linked judgments. The commercial-tool benchmark evaluates the complete workflow, and the released FreePDK45 layouts and simulator support open simulation studies. We hope this work helps process engineers build on the experience earned through repeated trials and makes the demanding work of recipe development a little easier.

Reproducibility Statement

The public package provides the VPEvolve code, experiment configurations, prompts, 20 FreePDK45-derived layouts, and a standalone lithography simulator with examples and tests: `https://github.com/vpevolve/vpevolve_code_and_data`. The simulator supports optical and resist simulation; the reported recipe optimization uses licensed Calibre software and process models. Section 4 and Appendices A–D describe the evaluation settings. Per-window measurements and call ledgers verify the aggregates; endpoint repeats check consistency.

Ethics Statement

VPEvolve is evaluated as a decision-support and automation harness inside an authorized computational environment; it does not control fabrication equipment. Process engineers freeze the editable surface, evaluator, budget, and approval boundary. Typed gates, immutable lineage, and exact-parent edits keep unsupported actions outside physical execution. Licensed software and process models remain in the authorized environment; the public package contains FreePDK45-generated layouts and a simulation runner. Appendix B gives the operational boundary.

AI Use Statement

We used large language models (LLMs) solely for language polishing and grammar checking.

References

Anthropic. Claude Sonnet 5 system card. System card, 2026. URL `https://anthropic.com/claude-sonnet-5-system-card`.

Guojin Chen, Wanli Chen, Yuzhe Ma, Haoyu Yang, and Bei Yu. DAMO: Deep agile mask optimization for full-chip scale. In *Proceedings of the 39th IEEE/ACM International Conference on Computer-Aided Design*, pp. 19:1–19:9, 2020. doi: 10.1145/3400302.3415705.

Guojin Chen, Haoyu Yang, Bei Yu, and Haoxing Ren. Intelligent OPC engineer assistant for semiconductor manufacturing. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39 (22):23144–23151, 2025. doi: 10.1609/aaai.v39i22.34479.

DeepSeek-AI. DeepSeek-V4.1-Flash: Pushing the limits of KV cache compression. *arXiv preprint arXiv:2609.19969*, 2026. URL `https://arxiv.org/abs/2609.19969`.

Jacob R. Gardner, Matt J. Kusner, Zhixiang Xu, Kilian Q. Weinberger, and John P. Cunningham. Bayesian optimization with inequality constraints. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 937–945, 2014. URL `https://proceedings.mlr.press/v32/gardner14.html`.

Gemma Team. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026. URL `https://arxiv.org/abs/2607.02770`.

Google DeepMind. Gemini 3.8 Flash model card. Model card, 2026. URL `https://deepmind.google/models/model-cards/gemini-3-8-flash/`.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. MLAgentBench: Evaluating language agents on machine learning experimentation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 20271–20309, 2024. URL `https://proceedings.mlr.press/v235/huang24y.html`.

Tianyi Li, Zhexin Tang, Tao Wu, Bei Yu, Jingyi Yu, and Hao Geng. LLM-SRAF: Sub-resolution assist feature generation using large language model. In *Proceedings of the 2025 Design, Automation and Test in Europe Conference*, pp. 1–7, 2025. doi: 10.23919/DATE64628.2025.10993108.

Mingjie Liu, Teodor-Dumitru Ene, Robert Kirby, Chris Cheng, Nathaniel Pinckney, Rongjian Liang, Jonah Alben, Himyanshu Anand, Sanmitra Banerjee, Ismet Bayraktaroglu, et al. ChipNeMo: Domain-adapted LLMs for chip design. *arXiv preprint arXiv:2311.00176*, 2023. URL https://arxiv.org/abs/2311.00176.

Siyi Liu, Chen Gao, and Yong Li. AgentHPO: Large language model agent for hyper-parameter optimization. In *Conference on Parsimony and Learning*, volume 280 of *Proceedings of Machine Learning Research*, pp. 1146–1169, 2025. URL https://proceedings.mlr.press/v280/liu25c.html.

Chris A. Mack. *Fundamental Principles of Optical Lithography: The Science of Microfabrication*. John Wiley & Sons, 2007. ISBN 978-0-470-01893-4. doi: 10.1002/9780470723876.

NC State EDA. FreePDK45. https://eda.ncsu.edu/freepdk/freepdk45/. Accessed 2026-09-23.

OpenAI. GPT-5.6 system card. System card, 2026. URL https://deploymentsafety.openai.com/gpt-5-6.

Siru Ouyang, Jun Yan, I-Hung Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long Le, Samira Daruki, Xiangru Tang, Vishy Tirumalashetty, George Lee, Mahsan Rofouei, Hangfei Lin, Jiawei Han, Chen-Yu Lee, and Tomas Pfister. Reasoningbank: Scaling agent self-evolving with reasoning memory. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/980ea04d23d1f6908964eba2a74afe45-Abstract-Conference.html.

Pi Contributors. Pi Agent Harness. https://github.com/earendil-works/pi, 2026. Software repository; accessed September 24, 2026.

Qwen Team. Qwen3.6-27B: Flagship-level coding in a 27B dense model. Official model release, April 2026. URL https://qwen.ai/blog?id=qwen3.6-27b.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/d842425e4bf79ba039352da0f658a906-Abstract-Conference.html.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Siemens Digital Industries Software. Calibre Computational Lithography. https://www.siemens.com/en-gb/products/ic/calibre-manufacturing/computational-lithography/, 2026. Product documentation; accessed September 20, 2026.

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, volume 25, 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/hash/05311655a15b75fab86956663e1819cd-Abstract.html.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=ehfRiF0R3a.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 63897–63911, 2025. URL https://proceedings.mlr.press/v267/wang25bx.html.

Haoyuan Wu, Zhuolun He, Xinyun Zhang, Xufeng Yao, Su Zheng, Haisheng Zheng, and Bei Yu. ChatEDA: A large language model powered autonomous agent for EDA. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 43(10):3184–3197, 2024. doi: 10.1109/TCAD.2024.3383347.

Haoyu Yang and Haoxing Ren. Enabling scalable AI computational lithography with physics-inspired models. In *Proceedings of the 28th Asia and South Pacific Design Automation Conference*, pp. 715–720, 2023. doi: 10.1145/3566097.3568361.

Haoyu Yang and Haoxing Ren. ILILT: Implicit learning of inverse lithography technologies. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 56319–56331, 2024. URL https://proceedings.mlr.press/v235/yang24s.html.

Haoyu Yang, Shuhe Li, Yuzhe Ma, Bei Yu, and Evangeline F. Y. Young. GAN-OPC: Mask optimization with lithography-guided generative adversarial nets. In *Proceedings of the 55th Annual Design Automation Conference*, pp. 1–6, 2018. doi: 10.1145/3195970.3196056.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Z.ai. GLM-5.3-Flash model card. Model card, 2026. URL https://huggingface.co/zai-org/GLM-5.3-Flash.

Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. Darwin gödel machine: Open-ended evolution of self-improving agents. In *International Conference on Learning Representations*, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/aa5f5e6eb6f613ec412f1d948dfa21a5-Abstract-Conference.html.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, Urmish Thakker, James Y. Zou, and Kunle Olukotun. Agentic context engineering: Evolving contexts for self-improving language models. In *International Conference on Learning Representations*, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/8a94ff6f922d995d7d3f4ebf4143e442-Abstract-Conference.html.

Tianshu Zhang and Huan Sun. SciNav: A general agent framework for scientific coding tasks. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/577cd5863ec73be4e6871340be0936ae-Abstract-Conference.html.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. ExpeL: LLM agents are experiential learners. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38 (17):19632–19642, 2024. doi: 10.1609/aaai.v38i17.29936.

A Layouts and Dataset Construction

The benchmark uses windows cropped from full-chip designs generated with the FreePDK45 technology library (NC State EDA), covering Poly (layer 9/0) and Metal1 (layer 11/0). Each layer contains ten windows, with two from each of five density strata ranging from very sparse to very dense. Each window serves as one evaluation unit in the main comparison.

Every window has a fixed $10 \times 10\,\mu$m scoring core and a $2\,\mu$m context halo. Whole polygons intersecting the context are retained in their original coordinates to supply simulation context; only the core is scored. The sampling manifest records the source-layout hash, layer, density group, and core and context coordinates. The local-correction study uses one case per stratum in each layer. The benchmark layouts and their metadata are included in the public release.

B Experimental Platform and Evaluation Protocol

Evaluation platform. Experiments use licensed Calibre software with two threads per job and fix the process model, rule decks, scoring code, available edits, and evaluation budget. Candidate validation follows Section 3: the harness checks the exact parent recipe, command syntax, and allowed edits, with pattern-selection and spatial-motion checks for local corrections. The actor selects trials, and measured responses guide recipe retention and experience updates. The public release includes a standalone lithography simulator for independent layout-level studies.

Measurements and selection. Normal edge-placement error (EPE) is measured at fixed external gauges with 1-nm spacing, excluding the first and last 25 nm of each original edge. Measurements use the $10 \times 10\,\mu$m scoring core. Max and Mean EPE use absolute errors at nominal focus and dose; Top-10 EPE averages the ten spatially separated worst hotspots. PVB is process-variation-band area per target-edge length. A feasible recipe has zero MRC and SRAF-printing violations and keeps mean EPE and PVB within 102% of their R0 values. Selection minimizes maximum EPE, breaking ties by Top-10 EPE, Mean EPE, and PVB, in that order.

For the main and mechanism comparisons, success requires completion with a feasible maximum-EPE reduction of at least 0.1 nm relative to R0. Success rates count all assigned runs. Layer-wise and overall quality means weight assigned windows equally, using each run's last retained feasible recipe; missing measurements are not imputed.

Evaluation budgets. Table A1 lists the trial budgets. Global, local, and diagnostic trials share the same budget, including evaluated candidates that are not retained and attempts with execution or measurement failures. Proposals rejected before evaluation consume no OPC evaluation. R0 initialization and an endpoint repeat are separate from search; the repeat re-evaluates the retained recipe to check its metrics and output geometry. Appendix D specifies model settings, and Appendix F reports evaluation and model-call costs.

Trial records link compiled candidates, parent recipes, edits, measurements, feasibility, and retention decisions. Run metadata records window coordinates and hashes, initial recipe and experience, model settings, prompts, tool manifests, code revisions, evaluator hashes, and call counts.

C Baseline Details

The baselines compare direct action selection, interaction history, accumulated experience, and numerical search under shared initial recipes, evaluation budgets, and quality constraints. Agent baselines use the same recipe-editing actions.

Raw Actor. Proposes recipe edits from the current task state and measurements, using the shared tools and evaluator without consolidating trial outcomes into an explicit experience bank.

ReAct (Yao et al., 2023). Interleaves reasoning, actions, and observations. During recipe optimization, it uses interaction history and tool feedback to revise its working hypothesis and choose subsequent edits, without a separate experience bank.

Table A1: Experimental configurations and per-run trial budgets. Each configuration uses all listed windows; continuations count additional trials. Budgets exclude initialization and one final re-evaluation per run (per completed run for long-horizon evaluation).

Experiment	Windows	Configurations	Trial budget
Main comparison	20	8 methods	10
Mechanism ablation	20	5 variants	10
Cross-model evaluation	20	6 models	10
Experience-value continuation	20	2 experience banks	5
Local-action continuation	10	2 action sets	4
Long-horizon evaluation	2	2 variants	100

Table A2: Generation settings for the cross-model comparison. Output limits include reasoning tokens. Default denotes the provider's sampling setting.

Model	Output tokens	Temperature
Claude Sonnet 5	4,096	Default
GPT-5.6-Luna	4,096	Default
Gemini 3.8 Flash	4,096	0.3
DeepSeek-V4.1-Flash	4,096	0.3
GLM-5.3-Flash	16,384	0.3
Gemma-4-31B	4,096	0.3

Textual Memory (Shinn et al., 2023; Zhao et al., 2024). Accumulates natural-language reflections on previous attempts. Trial outcomes and failures are summarized chronologically, providing textual experience that guides the agent's subsequent recipe edits.

ReasoningBank (Ouyang et al., 2026). Distills previous attempts into reusable lessons with applicability conditions and potential pitfalls. Our adaptation retrieves relevant lessons using lexical overlap with the current case and hotspot feedback to guide recipe edits.

ACE (Zhang et al., 2026b). Maintains an evolving playbook through reflection and incremental curation. The reflector analyzes recipe-trial feedback, and the curator incorporates these observations into playbook entries that guide subsequent actions.

Static Harness (Pi Contributors, 2026). Uses a Pi-style agent harness with fixed engineering guidance. The actor responds to new measurements when choosing recipe edits, while the process guidance remains unchanged throughout optimization.

Bayesian Optimization (Snoek et al., 2012; Gardner et al., 2014). Searches model-based recipe parameters using Gaussian-process surrogates for quality and feasibility. Constrained expected improvement selects the next candidate, and each evaluation updates the surrogates.

D Implementation and Experimental Settings

Models and initialization. Unless otherwise specified, all LLM roles use Qwen3.6-27B served locally on four NVIDIA H100 GPUs, with a 131,072-token context, 4,096 output tokens, and temperature zero. These settings apply to the main comparison, mechanism ablations, and paired continuations. VPEvolve and its mechanism ablations start from R0 with an empty Skill Bank and shared engineering guidance, under the budgets in Table A1.

In the cross-model study, all three roles use the same evaluated model through its API: Claude Sonnet 5 (Anthropic, 2026), GPT-5.6-Luna (OpenAI, 2026), Gemini 3.8 Flash (Google DeepMind, 2026), DeepSeek-V4.1-Flash (DeepSeek-AI, 2026), GLM-5.3-Flash (Z.ai, 2026), or Gemma-4-31B (Gemma Team, 2026). All use a 65,536-token context; Table A2 lists generation settings. Reasoning is disabled for Claude and GPT and set to low for Gemini and GLM. The sampling seed is fixed where supported. Long-horizon runs use DeepSeek-V4.1-Flash for all three roles through its API, with temperature zero, reasoning disabled, a 131,072-token context, and 4,096 output tokens.

Algorithm 1 VPEvolve: Recipe Optimization with Evolving Experience

Input: Layout x, initial recipe r_0 and feedback e_0, trial budget B;
frozen model π_θ, VPE harness H, lithography evaluator F
Output: Retained recipe r_B, Skill Bank $\mathcal{E}_B$, trial history $\mathcal{T}_B$

Initialize $\mathcal{E}_0 \leftarrow \varnothing$, $\mathcal{T}_0 \leftarrow \varnothing$
for $t = 0, \ldots, B-1$ **do**
  ▷ *Actor: choose a global, local, or diagnostic trial*
  $V_t \leftarrow \mathrm{Retrieve}(\mathcal{E}_t, x, r_t, e_t)$
  $a_t \sim \pi^{\mathrm{act}}_{\theta,H}(\cdot \mid x, r_t, e_t, V_t, \mathcal{T}_t)$
  Validate and compile $\widetilde{r}_t \leftarrow \mathrm{Apply}(r_t, a_t)$
  Evaluate $\widetilde{e}_t \leftarrow F(x, \widetilde{r}_t)$; record $\tau_t = (r_t, a_t, \widetilde{r}_t, e_t, \widetilde{e}_t)$
  $\mathcal{T}_{t+1} \leftarrow \mathcal{T}_t \cup \{\tau_t\}$
  ▷ *Recipe update: retain only feasible improvements*
  $(r_{t+1}, e_{t+1}) \leftarrow \mathrm{Select}(r_t, e_t, \widetilde{r}_t, \widetilde{e}_t)$
  ▷ *Reflector: Domain Response Reflection*
  $h_t \leftarrow \mathrm{Reflect}_{\pi_\theta}(\tau_t)$
  ▷ *Curator: Scoped Experience Revision and Experiment Guidance*
  $\Delta_t \leftarrow \mathrm{Curate}_{\pi_\theta}(\mathcal{E}_t, h_t, \tau_t)$
  $\mathcal{E}_{t+1} \leftarrow \mathrm{CheckApply}(\mathcal{E}_t, \Delta_t, \tau_t)$
end for
Re-evaluate r_B to verify the retained endpoint

Experience updates and controls. Algorithm 1 summarizes the optimization loop. Reflection compares the edit and predicted response with before/after measurements at fixed hotspots, new worst locations, and aggregate quality metrics, distinguishing an ineffective correction from a repaired target followed by a new limiting hotspot. Curation adds, updates, splits, or merges judgments linking interventions and applicability to supporting trials and counterexamples. Revisions preserve the intervention topic, review evidence reclassification or removal, and check cited interventions against trial records. Each judgment may propose a strategy update and up to two experiments with predicted effects and observations that would falsify the explanation.

Frozen Experience fixes the initial bank and guidance and uses only the current recipe and latest feedback. The three ablations remove domain reflection guidance, topic-preservation and evidence-reclassification review, or strategy updates and proposed experiments. All experience-update conditions share actor guidance, available actions, physical selection criteria, and factual citation checks.

E Paired Continuation Studies

The paired continuations in Sections 4.4.1 and 4.4.2 hold the starting recipe and history fixed to compare experience quality and the value of local actions. Both studies use the default model settings in Appendix D, the R0 quality constraints, and an endpoint repeat to verify the retained recipe.

Experience value. All 20 cases resume from Full's fifth-trial recipe, feedback, and history. ACE curates the five preceding observations from the same initial knowledge; VPEvolve supplies its recorded experience. Both banks remain fixed during five further trials, sharing the model, engineering guidance, and access to trial history.

Local actions. One case per density stratum and layer is selected before observing outcomes, giving five Poly and five Metal1 cases. Both branches start immediately before the first local intervention with the same recipe, feedback, and experience, and update experience over four trials. One permits global edits alone; the other permits global and local edits. Local actions displace selected mask fragments at a diagnosed edge while keeping target geometry and evaluation gauges fixed. The selected region and displacement are checked before execution under the shared quality constraints.

Table A3 reports endpoint quality; Table A4 gives paired maximum-EPE differences with 95% intervals from 20,000 bootstrap resamples of paired cases. VPEvolve experience and local actions have positive mean advantages on both layers. Local actions also increase mean EPE and PVB, reflecting a tradeoff within the feasibility limits.

Table A3: Endpoint quality in the paired continuation studies. EPE is in nm; PVB is in $10^{-3}\,\mu$m. Experience conditions cover the benchmark; each local-action condition has five cases per layer.

	Poly			Metal1		
Condition	Max↓	Mean↓	PVB↓	Max↓	Mean↓	PVB↓
ACE experience	5.313	0.647	7.404	16.328	0.698	5.022
VPEvolve experience	5.085	0.646	7.398	15.938	0.693	5.021
Global only	5.238	0.635	7.375	18.583	0.630	4.980
Global + local	5.185	0.651	7.379	18.012	0.698	4.996

Table A4: Paired maximum-EPE advantage of VPEvolve experience over ACE experience, or of global and local actions over global actions alone. Positive values favor the first condition. Intervals use 20,000 bootstrap resamples of paired cases.

Comparison	Layer	Pairs	Mean (nm)	95% interval (nm)
Experience curation	Poly	10	0.228	[-0.098, 0.735]
Experience curation	Metal1	10	0.390	[-0.207, 1.022]
Local actions	Poly	5	0.054	[-0.143, 0.329]
Local actions	Metal1	5	0.571	[-1.465, 2.883]

Table A5: Mean calls per assigned window in the main comparison, obtained by dividing Table 1's totals by 20. Commercial-tool calls include endpoint repeats and exclude R0 initialization.

Method	Commercial-tool calls	Model requests
Raw Actor	11	22.75
ReAct	11	26.20
Textual Memory	11	51.50
ReasoningBank	11	49.40
ACE	11	91.65
Static Harness	11	25.10
Bayesian optimization	11	0.00
VPEvolve	11	64.75

F Evaluation and Model-Call Cost

Table A5 expresses the main comparison's call counts per assigned window. All methods use 220 commercial-tool evaluations across 20 windows: ten optimization trials and one endpoint repeat per window. These counts exclude R0 initialization and include failed evaluations (Appendix B).

Model requests cover action selection, tool use, repairs, reflection, and curation. Per window, VPEvolve uses 64.75 requests, compared with 26.20 for ReAct and 91.65 for ACE.

G Long-Horizon Evaluation

The 100-trial comparison (Section 4.4.3) starts both variants from R0 with empty Skill Banks and shared guidance on one case per layer. It uses the DeepSeek-V4.1-Flash API settings in Appendix D and the quality constraints and evaluation accounting in Appendix B.

Both Poly runs and Metal1 Full complete 100 trials. On Metal1, Frozen repeatedly requests evidence excluded by its history restriction after trial 56; recovery completes one more trial before termination at trial 57. Poly Full also repeats evidence requests after trial 97 and completes the run following recovery. Recovery extends model-call allowances, with actual costs retained. The Metal1 comparison measures quality at trial 57 and the further gains as Full continues. Across both layers, Full and Frozen complete 200 and 157 trials (357 total), excluding initialization and endpoint repeats.