DECOUPLING TOPOLOGY AND TECHNOLOGY FOR PHYSICAL-AWARE ANALOG TOOL PARAMETER TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Analog layout generators automate physical implementation, but configuring them for different circuits and process design kits (PDKs) still requires expert knowledge and costly evaluations. Reusing historical tuning experience is difficult because parameter meanings and applicability vary across tasks, while changes in circuit structure and technology affect both layout responses and prediction reliability. We present **ADAPT**, which decouples circuit topology, PDK information, and parameter semantics into separate representations and fuses them to learn a global neural response prior from historical evaluations: technology information conditions the circuit graph network through feature-wise linear modulation (FiLM), and parameter choices then interact with the joint circuit–PDK context to predict layout metrics. A context-calibrated residual Gaussian process models deviations from this prior and quantifies predictive uncertainty. Target observations update the residual posterior while the neural prior remains fixed, and multi-objective Bayesian optimization selects subsequent tool parameter configurations. We construct a dataset covering 40 transistor-level circuits and three technology setups, comprising 120 circuit–PDK tasks with recorded parameter requests, execution outcomes, and available layout metrics. Experiments show that ADAPT achieves the best hypervolume (HV) any baseline attains in 40 evaluations within an average of 12.5 evaluations, a more than 3× reduction in layout tool executions. Our dataset and code are available online[1].

1 INTRODUCTION

Analog circuits underpin sensing, communication, and power management, but manual layout remains costly (Basso et al., 2024) (Figure 1). Generators such as ALIGN (Kunal et al., 2019) and MAGICAL (Xu et al., 2019) automate device generation, placement, and routing. High-quality results still require configuring device implementation, spacing, and signal and power routing through expert knowledge and repeated tool evaluations (Chen et al., 2024). These choices interact with

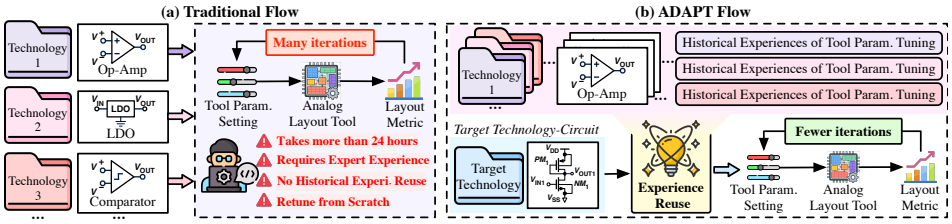

Figure 1: Traditional analog tool parameter tuning workflow vs. ADAPT workflow. (a) Traditional methods independently optimize each (circuit, technology) pair, leading to $N \times M$ repeated tuning efforts. (b) ADAPT learns transferable representations to enable historical experience reuse, dramatically reducing tuning costs.

[1] https://anonymous.4open.science/r/ICLR-ADAPT-1CA0/README.md

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.

circuit constraints and process design kit (PDK) rules, so effective settings may need retuning for another circuit or technology.

Learning-based tool tuning has advanced digital EDA through feature-guided search (Xie et al., 2020), multi-objective Bayesian optimization (MOBO) (Geng et al., 2022a; Zheng et al., 2023), and transfer learning (Geng et al., 2022b), with subsequent work improving parameter modeling and search strategies (Luo et al., 2024; 2025; Xu et al., 2024; Yu et al., 2026; Jeon & Kim, 2026; Xu et al., 2026b). Representative transfer methods learn relationships between design-specific objective responses using corresponding tool-parameter coordinates (Geng et al., 2022b). However, analog circuits involve far greater device-level degrees of freedom, together with symmetry, matching, and hierarchy constraints, so learning-based tuning has received much less attention in the analog domain, and analog layout tool parameter tuning is correspondingly harder. In analog layout, BO (Liu et al., 2020) and language-model-assisted search (Chen et al., 2024) have been explored for individual designs, but do not establish transfer across circuits and PDKs.

Reusing historical tuning experience poses three challenges. First, representations must distinguish circuit topology from PDK information while capturing their joint influence on layout. Connectivity specifies the structure to implement, whereas the PDK constrains its physical realization through available devices, metal layers, vias, and design rules. Second, parameter choices must have comparable meanings across tasks. Changing the circuit or PDK may alter the objects they control, render parameters inactive, or cause previously effective configurations to fail. Third, patterns learned from historical evaluations may not hold for a new circuit or PDK. Effective tuning therefore requires assessing both predicted layout quality and prediction reliability to determine how much to trust historical experience and where further exploration is needed.

We recognize that the circuit, the PDK, and the tool parameters are tightly interrelated yet play distinct roles: the circuit defines the object to be implemented, the PDK supplies the global constraints and technology information, and the parameters act on both through the layout tool to produce the final layout and its metrics. Three insights follow from this observation. First, circuit structure should be encoded before incorporating technology. Second, parameters should encode the operations they control, the objects they address, the meanings of their choices, and their applicability. Grouping modules by hierarchy, nets by port status and branching, and MOS primitives by polarity and composition establishes consistent parameter roles across tasks. Third, historical parameter tuning experience should provide a prior that target observations can refine, since training accuracy alone does not establish predictive reliability on new tasks. Shared and task-specific error components allow related experience to guide adaptation while accounting for errors specific to the target.

Building on these insights, we propose **ADAPT** for analog layout tool parameter tuning across circuits and PDKs. ADAPT learns a neural response prior from historical evaluations and augments it with a context-calibrated residual Gaussian process (GP); for a new circuit–PDK pair, target observations update only the residual posterior, and MOBO selects subsequent executable configurations.

Our contributions are:

- A dataset of 40 transistor-level circuits across three technology setups: 120 circuit–PDK tasks recording parameter requests, execution outcomes, and available layout metrics.
- Separate topology, technology, and parameter-semantics representations, fused through late FiLM and parameter–context interactions to learn a transferable neural response prior.
- A context-calibrated residual GP fitted to out-of-fold errors, with shared and task-specific components supporting uncertainty-aware adaptation from limited target observations.
- On five 14 nm FinFET benchmarks, ADAPT reaches the best baseline HV@40 in 12.5 evaluations on average, reducing evaluations by more than $3\times$. Its HV@20 exceeds every baseline's HV@40, and its HV@40 improves on the strongest baseline by 1.9%–4.5% per task.

2 Related Works

Analog Layout Generation. ALIGN Kunal et al. (2019) and MAGICAL Xu et al. (2019) translate netlists into physical layouts through constraint extraction, device generation, placement, and routing under PDK-specific rules. Exposed tool parameters steer each stage and can markedly alter

layout metrics even for a fixed circuit and PDK. Because parameter applicability and effects depend on both circuit structure and technology, configuration remains expertise-driven, motivating learned tuning across circuit–PDK pairs.

Circuit Representation. Graph-based representations provide structural inductive biases for analog design automation. GCN-RL leverages circuit graphs for transistor sizing and transfers across topologies and technology nodes Wang et al. (2020), CktGNN adopts nested graph encoders for joint topology and sizing optimization Dong et al. (2023), and KCLNet incorporates Kirchhoff-law-inspired constraints to obtain electrically informed embeddings Xu et al. (2026a). These works confirm the value of learning circuit structure, yet they target circuit synthesis or characterization rather than modeling how tool parameters shape layout metrics across circuit–PDK pairs. Our representation instead couples heterogeneous device–net connectivity with explicit technology conditioning to serve this task.

EDA Tool Parameter Tuning. Learning-based tool tuning has advanced rapidly in digital EDA, evolving from multi-objective Bayesian optimization to transfer learning, high-dimensional trust-region search, hybrid-space modeling, preference learning, and causal or learned search representations Xie et al. (2020); Geng et al. (2022b;a); Zheng et al. (2023); Luo et al. (2024; 2025); Xu et al. (2024); Yu et al. (2026); Jeon & Kim (2026); Xu et al. (2026b). Analog layout tuning has received far less attention, and its difficulty extends beyond expensive evaluation: circuit topology, symmetry and matching constraints, hierarchy, routing intent, and PDK-specific legality jointly determine both the feasible layouts and the effective tuning variables, so the parameter space itself varies across circuit–PDK pairs. The closest prior work embeds multi-objective Bayesian optimization into analog layout synthesis with layout metric feedback, but optimizes design-specific critical-net weights independently for each circuit Liu et al. (2020). ADAPT instead casts heterogeneous tuning tasks into a shared conditional parameter space and explicitly models circuit and technology context, allowing tuning experience to transfer across circuit–PDK pairs.

3 Problem Formulation

A tuning task is specified by a context $\mathbf{c} = (C, P)$, where C is the circuit and P the PDK. The analog layout tool is an expensive black box: given $\mathbf{c}$ and a parameter configuration $\mathbf{x}$, a successful execution returns M layout metrics $\mathbf{y} = y(\mathbf{c}, \mathbf{x}) \in \mathbb{R}^M$ to be minimized, while an execution may also fail and return no metrics. To make experience comparable across tasks, all tasks share a d-dimensional conditional parameter space with fixed coordinate semantics: each context induces a feasible set $\mathcal{X}(\mathbf{c})$ via activity masks, and a task-specific decoder maps $\mathbf{x} \in \mathcal{X}(\mathbf{c})$ to native tool inputs (Section B).

We are given historical observations $\mathcal{D}_{\text{hist}} = \{(\mathbf{c}_i, \mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N_{\text{hist}}}$ from source tasks. For a target context $\mathbf{c}_*$, whose circuit, PDK, or their combination may be unseen in $\mathcal{D}_{\text{hist}}$, a budget of $T_{\max} \ll N_{\text{hist}}$ tool executions is available. A failed execution consumes one evaluation slot but provides no regression target and therefore does not update the response-model posterior. The goal is to sequentially select $\mathbf{x}_1, \ldots, \mathbf{x}_{T_{\max}} \in \mathcal{X}(\mathbf{c}_*)$ such that the successfully evaluated configurations approximate the Pareto-optimal set

$$\mathcal{X}^* = \big\{\mathbf{x} \in \mathcal{X}(\mathbf{c}_*) \,\big|\, \nexists\, \mathbf{x}' \in \mathcal{X}(\mathbf{c}_*) : \, y(\mathbf{c}_*, \mathbf{x}') \prec y(\mathbf{c}_*, \mathbf{x})\big\}, \tag{1}$$

where $\prec$ denotes Pareto dominance; solution quality is measured by the HV of the returned non-dominated set. Unlike per-design tuning, which optimizes each circuit–PDK pair in isolation, both the response $y(\mathbf{c}, \cdot)$ and the feasible set $\mathcal{X}(\mathbf{c})$ here vary with the context, and $\mathcal{D}_{\text{hist}}$ carries no guarantee about $y(\mathbf{c}_*, \cdot)$. A solution must therefore transfer what historical evaluations reveal about the context-dependent response while assessing how far that experience remains reliable on the target.

4 Method

4.1 Overall Flow

In Figure 2, for a task context $\mathbf{c} = (C, P)$, decoupled representations encode circuit topology, PDK technology, and parameter semantics. Technology conditions the circuit graph network through FiLM, yielding $z_{\text{ctx}} = E_\theta(\mathbf{c})$ independently of $\mathbf{x}$. Parameter semantics interact with this context to

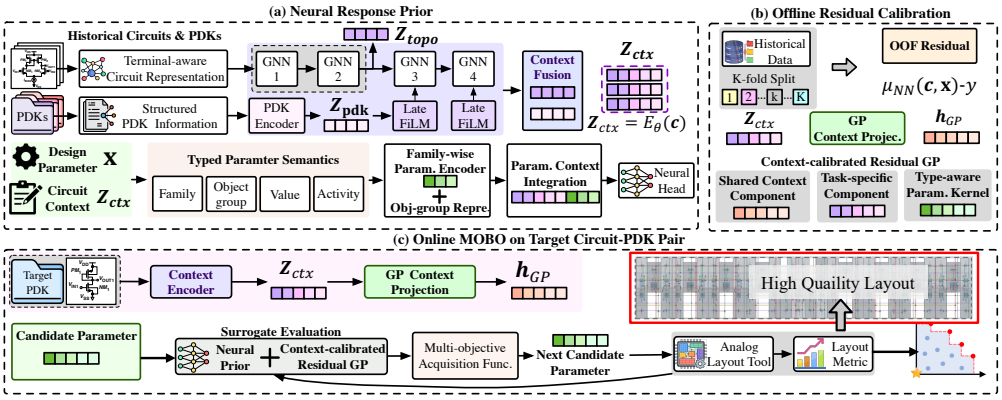

Figure 2: Overview of ADAPT. (a) Decoupled representations of circuit topology, PDK information, and parameter semantics are fused into a neural response prior. (b) A context-calibrated residual GP with shared and task-specific components is fitted to out-of-fold prediction errors. (c) Target observations update only the residual posterior, while the neural prior remains fixed, guiding multi-objective Bayesian optimization of tool parameters.

predict the M layout metrics. The neural response prior is learned from $\mathcal{D}_{\text{hist}}$; a context-calibrated residual GP models its deviations and predictive uncertainty. During target optimization, the neural prior remains fixed, target observations update only the residual posterior, and MOBO selects configurations in $\mathcal{X}(\mathbf{c}_*)$.

4.2 Decouple Topology, Technology, and Parameter Semantics

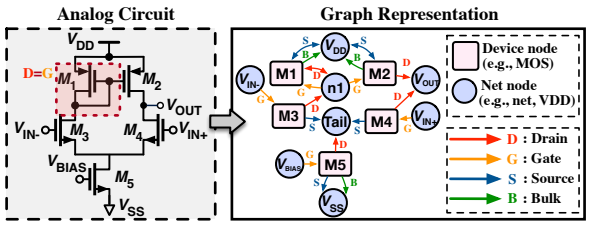

Figure 3: Terminal-aware circuit representation. Device and net nodes are connected by terminal-specific incidences. Drain and gate connections remain distinct when they share a net.

Terminal-Aware Circuit Representation. We represent C as a heterogeneous bipartite multigraph $G_C = (V_d \cup V_n, E_t)$, where V_d and V_n contain device and net nodes and each incidence $(v_d, \tau, v_n) \in E_t$ records the terminal role τ, such as drain, gate, source, or bulk. Distinct terminal incidences are retained even when they connect the same device and net; a diode-connected transistor thus keeps separate drain and gate incidences to one net (Figure 3).

Since circuit structure should be encoded before incorporating technology, the early graph blocks use purely structural descriptors, including device type, connectivity, multiplicity, hierarchy, and explicit constraint and group membership; technology-dependent descriptors are reserved for the later blocks. A device-to-net message at block ℓ is

$$m^{(\ell)}_{v_d \to v_n} = \phi^{d\to n}_{\ell}\left([h^{(\ell-1)}_{v_d} \| h^{(\ell-1)}_{v_n} \| e_\tau]\right), \tag{2}$$

where $h^{(\ell-1)}_v$ is the node state, e_τ encodes the terminal role, and $\|$ denotes concatenation; reverse messages use a separate function. Nodes aggregate incident messages and update their states with residual connections and layer normalization. Pooling after the two structural blocks yields $z_{\text{topo}} = \text{Pool}_{\text{topo}}(H^{(2)})$, where $H^{(\ell)}$ collects the node states.

Structured PDK Encoding. We encode the technology information of P as separate collections of metal-layer, via, and process/generator tokens describing layer order and routing rules, via geometry and enclosure, and device families and generator capabilities, respectively (Figure 4). Missing

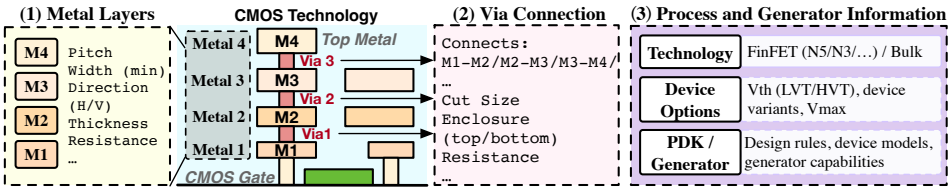

Figure 4: Structured technology information supplied to the PDK encoder: metal-layer resources, via connections, and process and generator capabilities.

attributes carry presence indicators rather than being interpreted as physical zeros. Type-specific encoders and mean/max pooling give

$$p_a = \mathrm{Pool}_{\text{mean,max}}\big(\{\phi_a(t) : t \in \mathcal{T}_a(P)\}\big), \quad a \in \{\mathrm{M}, \mathrm{V}, \mathrm{G}\}, \quad z_{\text{PDK}} = \phi_{\text{PDK}}([p_{\mathrm{M}} \| p_{\mathrm{V}} \| p_{\mathrm{G}}]), \tag{3}$$

where $\mathcal{T}_a(P)$ is the corresponding token collection; layer order and via endpoints are retained as token attributes.

Typed Parameter-Semantics View. A shared parameter coordinate need not correspond to the same physical choice in every context. We therefore construct a deterministic typed view $\chi(\mathbf{x}; \mathbf{c})$ recording the parameter families, addressed object groups, candidate identities, normalized ordered values, and activity and capability masks of $\mathbf{x}$ under $\mathbf{c}$. Following the deterministic grouping of Section B, modules are grouped by hierarchy, nets by port status and branching, and MOS primitives by polarity and composition, so each coordinate retains a consistent role across tasks. These semantics derive from the fixed parameter schema and task inputs, not from evaluation outcomes.

4.3 Fusion into a Neural Response Prior

Technology Conditioning with Late FiLM. Technology conditions only the later graph blocks through feature-wise linear modulation (FiLM) (Perez et al., 2018):

$$H^{(\ell)} = \gamma_\ell(z_{\text{PDK}}) \odot \mathrm{GNN}_\ell\big(H^{(\ell-1)}, G_C; \Xi(C, P)\big) + \beta_\ell(z_{\text{PDK}}), \quad \ell \in \{3, 4\}, \tag{4}$$

where $\odot$ denotes elementwise multiplication and $\Xi(C, P)$ collects fixed, implementation-specific device descriptors. The final context representation is

$$z_{\text{ctx}} = E_\theta(\mathbf{c}) = \mathrm{Fuse}_\theta\Big([z_{\text{topo}} \| z_{\text{PDK}} \| \mathrm{Pool}(H^{(4)})]\Big), \tag{5}$$

and neither $\mathbf{x}$ nor candidate-dependent implementation changes enter E_θ.

Parameter–Context Interaction. A family-wise parameter encoder maps the typed view to $e_{\mathbf{x}} = A_\psi\big(\chi(\mathbf{x}; \mathbf{c}), g_{\text{target}}(\mathbf{c})\big)$, where $g_{\text{target}}(\mathbf{c})$ pools representations of the object groups addressed by the parameters. The mean of the neural response prior is

$$\mu_{\text{NN}}(\mathbf{c}, \mathbf{x}) = \mathrm{MLP}_\omega([z_{\text{ctx}} \| e_{\mathbf{x}} \| (W_{\text{ctx}} z_{\text{ctx}}) \odot (W_{\text{par}} e_{\mathbf{x}})]) \in \mathbb{R}^M, \tag{6}$$

with W_{ctx} and W_{par} projecting to a common width, so the response to a parameter choice depends on its joint circuit–PDK context. The embedding $e_{\mathbf{x}}$ is used only for neural prediction; optimization never searches this latent space.

Training. The M nonnegative layout metrics are transformed as

$$\tilde{y}_{i,m} = \big(\log(1 + y_{i,m}) - \mu_m^{\text{train}}\big)/s_m^{\text{train}}, \tag{7}$$

with normalization statistics estimated from source-training data, $s_m^{\text{train}} > 0$, and fixed during target optimization. The neural response prior is trained end to end using

$$\mathcal{L}_{\text{NN}} = \frac{1}{|\mathcal{I}|} \sum_{(i,m) \in \mathcal{I}} \mathrm{SmoothL1}(\mu_{\text{NN},m}(\mathbf{c}_i, \mathbf{x}_i) - \tilde{y}_{i,m}) + \lambda_{\text{mask}} \mathcal{L}_{\text{mask}} + \lambda_{\text{topo}} \mathcal{L}_{\text{topo}}, \tag{8}$$

where $\mathcal{I}$ indexes available layout metric observations. The auxiliary terms, defined in the appendix, provide masked-feature reconstruction and weak alignment of corresponding topology representations across PDKs; alignment acts on z_{topo} rather than z_{ctx}, so metric supervision shapes the context encoder without discarding technology information.

4.4 Context-Calibrated Residual Gaussian Process

Training accuracy alone does not establish predictive reliability on new tasks, and a representation trained for prediction need not provide a suitable distance for GP covariance. We therefore freeze the neural response prior and learn a separate low-dimensional linear context projection $h_{\mathrm{GP}}(\mathbf{c}) = T_\eta(z_{\mathrm{ctx}})$ using a GP evidence objective. The GP receives $(h_{\mathrm{GP}}(\mathbf{c}), \chi(\mathbf{x};\mathbf{c}))$ rather than $e_{\mathbf{x}}$, and task identity is retained for the private component below. For objective m, the latent response on the transformed scale is modeled as

$$f_m(\mathbf{c},\mathbf{x}) = \mu_{\mathrm{NN},m}(\mathbf{c},\mathbf{x}) + r_m(\mathbf{c},\mathbf{x}), \quad r_m \sim \mathcal{GP}(0,k_m), \quad \tilde{y}_m = f_m + \epsilon_m,\ \epsilon_m \sim \mathcal{N}(0,\sigma^2_{\mathrm{obs},m}), \tag{9}$$

with an independent residual GP per objective: the neural prior specifies the mean, while context and parameter semantics determine residual covariance.

Parameter families are organized into semantic groups $\mathcal{G}$ by the tool stages they control, with kernel

$$k_{\mathrm{par}}(\chi,\chi') = \sum_{g\in\mathcal{G}} \alpha_g k_g(\chi_g,\chi'_g), \qquad \alpha_g > 0, \tag{10}$$

where χ_g restricts the typed view to group g and k_g is a type-aware Matérn-$5/2$ component: ordered values retain their ordering, categorical identities carry no numerical order, and inactive values are canonicalized so that changing only an inactive coordinate creates no artificial distance. The positive-semidefinite construction is specified in the appendix. Writing $\chi = \chi(\mathbf{x};\mathbf{c})$ and $\chi' = \chi(\mathbf{x}';\mathbf{c}')$, we define

$$k_m\big((\mathbf{c},\mathbf{x}),(\mathbf{c}',\mathbf{x}')\big) = \big[\sigma^2_{s,m}\, k_{\mathrm{ctx}}\big(h_{\mathrm{GP}}(\mathbf{c}), h_{\mathrm{GP}}(\mathbf{c}')\big) + \sigma^2_{p,m}\, \delta_{\mathrm{ctx}}(\mathbf{c},\mathbf{c}')\big]\, k_{\mathrm{par}}(\chi,\chi'), \tag{11}$$

where k_{ctx} is a Matérn-$5/2$ kernel and δ_{ctx} equals one exactly for the same task context. The shared component transfers residual structure between contexts, so related historical experience guides adaptation, while the private component absorbs task-specific errors and remains at its prior for an unseen context until target observations arrive. With residual posterior mean $\mu_{r,m}$ and variance $\sigma^2_{r,m}$, the predictive mean and observation variance are

$$\mu_{\mathrm{final},m} = \mu_{\mathrm{NN},m} + \mu_{r,m}, \qquad \sigma^2_{\mathrm{final},m} = \sigma^2_{r,m} + \sigma^2_{\mathrm{obs},m}, \tag{12}$$

where observation noise enters only when predicting a future observation; acquisition evaluation uses the latent posterior.

4.5 Offline-to-Online Bayesian Adaptation

Offline Residual Calibration. To expose the GP to neural prediction errors on held-out contexts, historical data are partitioned into K folds by circuit lineage, keeping related circuit variants and their PDK instances together. On the common transformed scale, the residual target of observation i in fold $k(i)$ is

$$r^{\mathrm{OOF}}_{i,m} = \tilde{y}_{i,m} - \mu^{(-k(i))}_{\mathrm{NN},m}(\mathbf{c}_i,\mathbf{x}_i), \tag{13}$$

where $\mu^{(-k)}_{\mathrm{NN}}$ is trained without fold k; these cross-fitted errors act as proxies for errors on unseen contexts. We then refit the neural response prior on all source-training data, freeze it, and fit T_η, kernel and likelihood parameters, and the residual posterior to these errors via sparse variational inference (Burt et al., 2020), maximizing the sum of objective-wise evidence lower bounds without back propagating through the frozen prior. Task supervision thus learns predictive features, and the evidence objective learns how those features inform residual covariance.

Target Adaptation and Parameter Selection. For a target context $\mathbf{c}_* = (C_*, P_*)$, we compute $z_{\mathrm{ctx},*} = E_\theta(\mathbf{c}_*)$ and $h_{\mathrm{GP},*} = T_\eta(z_{\mathrm{ctx},*})$ once; both remain fixed throughout target optimization. Initial evaluations combine a default configuration, transferred configurations, and diverse exploration. Each target observation at step t supplies $r^{\mathrm{obs}}_{t,m} = \tilde{y}_{t,m} - \mu_{\mathrm{NN},m}(\mathbf{c}_*,\mathbf{x}_t)$, which updates both the shared and target-specific residual components wherever the objective is observed.

Writing $f = (f_1,\dots,f_M)^\top$, let π_t denote the joint posterior over $-f(\mathbf{c}_*,\cdot)$ after t target evaluations, where negation converts minimization to the maximization convention of the acquisition function. For a fixed reference point b_{ref} on the transformed scale, the MOBO step approximately solves

$$\max_{\substack{\mathcal{B}\subset\mathcal{X}(\mathbf{c}_*)\\ |\mathcal{B}|=q}} \alpha_t(\mathcal{B}), \qquad \alpha_t(\mathcal{B}) = \mathrm{qLogNEHVI}(\mathcal{B};\pi_t,b_{\mathrm{ref}}), \tag{14}$$

Table 1: The results of multi-objective optimization on five 14 nm benchmarks with a budget of 40 evaluations on a single random seed. Best values are in **bold**.

Circuit	Method	HV@20 ↑	HV@40 ↑	AUC-HV @40↑	IGD$^+$ @40↓	A_{eval} ↓ (μm^2)	$L_{\text{eq,signal}}$ ↓ (μm)	$L_{\text{eq,power}}$ ↓ (μm)	N_{cut} ↓ (count)
CLK-CMP	DAC'20	2.68e-2	2.68e-2	1.008	8.80e-3	200.69	640.69	621.12	1056
	PPATuner	2.24e-2	2.29e-2	0.889	4.04e-2	720.19	738.17	1404.92	1209
	Attention-EDA	2.25e-2	2.47e-2	0.924	2.63e-2	297.79	609.17	770.82	1145
	CausalTuner	2.13e-2	2.13e-2	0.850	4.01e-2	**165.33**	**578.67**	1119.41	1421
	Reshaping BO	2.12e-2	2.30e-2	0.857	4.02e-2	609.04	622.75	1035.35	1121
	Ours	**2.74e-2**	**2.80e-2**	**1.049**	**4.01e-5**	189.29	585.64	**595.61**	**1040**
PT-CMP	DAC'20	5.21e-2	5.26e-2	2.041	2.97e-2	108.05	370.22	365.08	618
	PPATuner	5.07e-2	5.11e-2	2.007	3.44e-2	137.09	348.36	350.13	615
	Attention-EDA	5.06e-2	5.42e-2	2.017	5.03e-3	89.32	**278.53**	**294.70**	603
	CausalTuner	5.09e-2	5.11e-2	2.005	3.33e-2	108.05	370.22	365.08	618
	Reshaping BO	5.35e-2	5.37e-2	2.063	1.03e-2	**89.10**	310.74	295.56	614
	Ours	**5.43e-2**	**5.63e-2**	**2.131**	**5.50e-4**	103.62	340.44	311.39	**582**
CT-CMP	DAC'20	2.70e-2	3.12e-2	0.966	7.10e-3	137.10	**497.19**	476.00	1132
	PPATuner	2.20e-2	2.59e-2	0.864	4.42e-2	229.69	675.90	650.86	1339
	Attention-EDA	2.72e-2	2.89e-2	0.943	1.92e-2	144.43	558.40	505.60	1132
	CausalTuner	2.76e-2	2.89e-2	0.970	2.01e-2	129.95	600.00	534.44	1135
	Reshaping BO	2.62e-2	2.84e-2	0.963	1.75e-2	131.73	500.44	689.22	1236
	Ours	**3.16e-2**	**3.18e-2**	**1.122**	**6.63e-3**	**108.63**	541.13	**448.70**	**1124**
DC-Switch	DAC'20	8.38e-2	8.38e-2	3.289	1.31e-2	71.61	225.13	189.38	372
	PPATuner	8.35e-2	8.38e-2	3.285	1.35e-2	**59.44**	224.06	116.23	377
	Attention-EDA	8.43e-2	8.69e-2	3.344	1.11e-2	61.60	**223.18**	119.51	353
	CausalTuner	8.46e-2	8.46e-2	3.311	1.32e-2	60.20	233.64	119.51	357
	Reshaping BO	8.35e-2	8.35e-2	3.283	1.35e-2	**59.44**	224.06	116.23	377
	Ours	**8.95e-2**	**9.03e-2**	**3.469**	**4.67e-5**	79.49	226.51	**111.72**	**342**
REF-BUF	DAC'20	5.18e-2	5.19e-2	2.026	1.15e-2	95.32	**284.88**	319.03	702
	PPATuner	5.15e-2	5.15e-2	2.022	1.23e-2	92.86	289.59	320.53	703
	Attention-EDA	5.20e-2	5.28e-2	2.039	6.40e-3	83.56	316.56	305.23	691
	CausalTuner	5.11e-2	5.12e-2	2.012	1.11e-2	96.85	334.41	346.04	702
	Reshaping BO	5.33e-2	5.34e-2	2.076	4.51e-3	95.90	339.83	307.95	683
	Ours	**5.45e-2**	**5.54e-2**	**2.131**	**3.10e-3**	**78.76**	319.72	**260.77**	**672**

using the log-domain noisy expected hypervolume improvement (Daulton et al., 2021; Ament et al., 2023) over the joint latent posterior. Candidates combine global mixed-space samples, applicable historical configurations, and local perturbations, all respecting the activity masks and admissible combinations of $\mathcal{X}(\mathbf{c}_*)$, so every proposal remains an executable configuration rather than a latent vector. The batch size q is capped by the remaining budget, and the loop returns the nondominated configurations with complete observed layout metric vectors after at most $T_{\max}$ target evaluations.

5 Experimental Results

5.1 Experimental Settings

Dataset and Benchmarks. Our dataset for layout tool parameter tuning is based on ALIGN tool Kunal et al. (2019), containing 40 transistor-level circuits across SKY130, 65 nm, and 14 nm, totaling 120 circuit–PDK tasks. We split the circuits into 30/5/5 for training, validation, and testing, respectively, keeping all PDK instances of each circuit in the same split; this yields 90 training and 15 validation tasks. The five held-out 14 nm benchmarks are a clocked latch comparator (CLK-CMP, 28 MOS instances), a pass-transistor equality comparator (PT-CMP, 53), a continuous-time rail-to-rail comparator (CT-CMP, 26), a 2:1 switched-capacitor switch core (DC-Switch, 24), and a reference buffer core (REF-BUF, 39); their SKY130 and 65 nm instances are not used.

Parameter Space. We use 35 discrete coordinates spanning 20 placement, primitive-generation, signal-routing, and power-routing parameter families: 11 global or stage-level controls, 10 module-group controls, four net-group controls, and 10 primitive-group controls. Deterministic groups distinguish root and descendant modules, port and internal nets by branching degree, and MOS primitives by polarity and composition. A circuit- and PDK-specific decoder maps choices to native ALIGN inputs; proven inactive coordinates retain their reference policies. Appendix B specifies the groups, domains, masks, and decoding rules.

Evaluation Metrics. We minimize four post-route objectives: evaluated layout area A_{eval}, signal and power equivalent metal lengths $L_{\text{eq,signal}}$ and $L_{\text{eq,power}}$, and intermetal via-cut count N_{cut}. Pareto

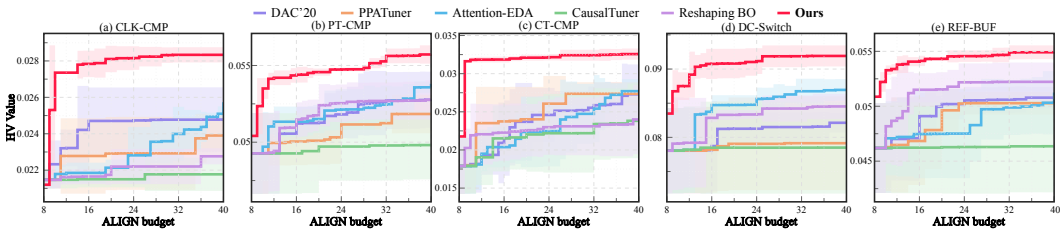

Figure 5: HV trajectories on five 14 nm benchmarks. Lines and shaded regions denote the mean and ± 1 standard deviation over three random seeds.

fronts use complete, connectivity-qualified observations with all four metrics; We report HV at 20/40 evaluations, $\mathrm{AUC}-\mathrm{HV}_{40} = \sum_{t=1}^{40} \mathrm{HV}_t$, and IGD^+ against a fixed empirical nondominated reference front. Failures and exact-cache hits count toward the 40-evaluation budget. Each row's physical metrics describe one observed Pareto layout.

Baselines and Experimental Configuration. We compare ADAPT with DAC'20 Liu et al. (2020), PPATuner Geng et al. (2022b), Attention-EDA Luo et al. (2025), CausalTuner Yu et al. (2026), and Reshaping BO Jeon & Kim (2026). All methods use the same 35D action contract, evaluation pipeline, and 40-query budget, with ALIGN seed fixed to 1. ADAPT uses eight initial evaluations followed by 32 sequential qLogNEHVI proposals with frozen neural and GP hyperparameters.

5.2 Experimental Results and Analysis

Neural Response Prediction. Table 2 reports prediction accuracy on the 14 nm subset. MAPE ranges from 5.65% to 10.97%, with PCC between 0.892 and 0.991. Prediction quality varies across objectives: R^2 reaches 0.977 for N_{cut} and 0.968 for $L_{\text{eq,power}}$, compared with 0.776 for A_{eval} and 0.800 for $L_{\text{eq,signal}}$. These differences motivate assessing residual correction separately rather than inferring optimization performance from neural prediction accuracy alone.

Fixed-Budget Multi-Objective Optimization. Table 1 shows the highest HV@20, HV@40, and AUC-HV@40, and the lowest IGD^+@40, for ADAPT on all five benchmarks. HV@40 exceeds the strongest baseline on each benchmark by approximately 1.9–4.5%. The advantage does not extend uniformly to individual physical objectives. For example, the selected DC-Switch layout has lower $L_{\text{eq,power}}$ and N_{cut} than all baseline layouts, but occupies more area. The results therefore concern multi-objective trade-offs rather than simultaneous improvement in every physical metric.

Table 2: The accuracy of Neural Response Prediction

PDK	Metric	MAPE (%)	PCC	R^2
target14	A_{eval}	10.974	0.892	0.776
target14	$L_{\text{eq,signal}}$	5.653	0.906	0.800
target14	$L_{\text{eq,power}}$	10.137	0.985	0.968
target14	N_{cut}	6.635	0.991	0.977

Computational Efficiency and HV Convergence Analysis. ADAPT improves mean HV early and retains the highest final mean on all five benchmarks (Figure 5). In the individual runs of Table 1, its HV@20 already exceeds every baseline's HV@40. The matched-quality analysis in Appendix C reports an average of 12.5 evaluations to reach the best baseline HV@40. These are evaluation-efficiency results, not wall-clock speedups.

5.3 Ablation Study

Neural Response Prior. Table 4 reports MAPE / R^2 / PCC macro-averaged over the five 14 nm circuits; all variants are retrained with the same protocol. **w/o Param. Semantics** replaces the typed parameter view with a flat vector; **w/o Late FiLM** removes technology conditioning; **w/o Message Passing** encodes nodes independently; **w/o Response Supervision** trains encoders with auxiliary objectives only and fits the head on frozen features. Removing parameter semantics increases MAPE on all four metrics, while removing response supervision most strongly degrades area and signal-wirelength prediction.

Optimization Pipeline. Table 3 reports mean HV at 40 evaluations over three seeds, with shared initial points and reference point. **w/o neural prior mean** uses a zero-mean GP; **w/o offline resid-**

Table 3: Mean HV (e-2) at 40 evaluations over 3 seeds.

Method	CLK-CMP	DC-Switch	REF-BUF
Full (ours)	**2.834**	**9.192**	**5.494**
w/o neural prior mean	2.824	9.178	5.474
w/o offline residual calibration	2.806	9.111	5.460
NN-only greedy	2.749	9.038	5.431

Table 4: Ablation of the neural response prior. Each cell reports MAPE(%)↓ / R^2 ↑ / PCC↑, macro-averaged over the five 14 nm circuits. Best in bold.

Variant	A_{eval}	$L_{\text{eq,signal}}$	$L_{\text{eq,power}}$	N_{cut}
Full (ours)	**10.97 / 0.776 / 0.892**	**5.65 / 0.800 / 0.906**	**10.14 / 0.968 / 0.985**	**6.63 / 0.977 / 0.991**
w/o Param. Semantics	13.64 / 0.704 / 0.850	7.05 / 0.716 / 0.864	12.21 / 0.942 / 0.975	7.45 / 0.962 / 0.985
w/o Late FiLM	12.32 / 0.766 / 0.886	6.17 / 0.777 / 0.895	11.32 / 0.949 / 0.978	6.92 / 0.971 / 0.988
w/o Terminal-Aware Message Passing	12.14 / 0.771 / 0.887	5.87 / 0.794 / 0.907	11.01 / 0.958 / 0.982	6.94 / 0.974 / 0.990
w/o Response Supervision	13.98 / 0.602 / 0.810	8.98 / 0.499 / 0.780	11.64 / 0.944 / 0.974	6.99 / 0.973 / 0.988

ual calibration discards the historical residual posterior; **NN-only greedy** selects by ranking μ_{NN} without a GP. The ordering is identical on all three benchmarks: greedy selection is worst and the offline posterior accounts for most of the remaining gap.

5.4 Case Study

Figure 6 presents routed layouts for three circuit classes: a comparator, a switched-capacitor switch core, and a reference buffer. Table 5 reports all four objectives for each selected layout. DC-Switch and REF-BUF have similar evaluated areas (79.49 and 78.76 μm^2), yet REF-BUF uses 2.33 times the power equivalent metal length and 1.96 times as many intermetal via cuts. This contrast illustrates why area alone is insufficient to characterize the returned layouts.

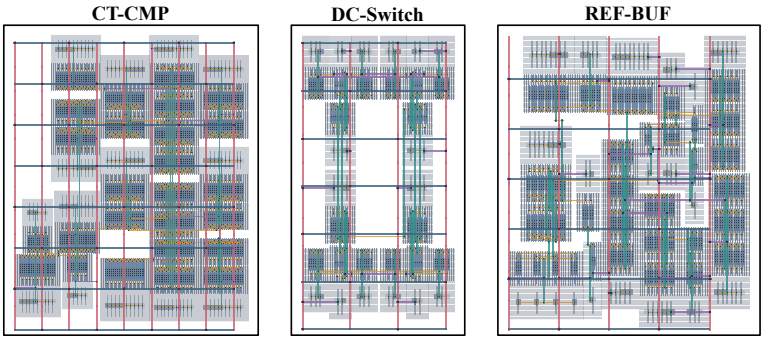

Figure 6: Three routed layouts for CT-CMP, DC-Switch, and REF-BUF, respectively, with zero reported ALIGN DRC violations.

Table 5: Post-route metrics of the layouts in Figure 6. Area is measured in μm^2 and equivalent metal lengths in μm; N_{cut} counts intermetal via cuts.

Metric	CT-CMP	DC-Switch	REF-BUF
A_{eval}	108.63	79.49	78.76
$L_{\text{eq,signal}}$	541.13	226.51	319.72
$L_{\text{eq,power}}$	448.70	111.72	260.77
N_{cut}	1124	342	672

6 Conclusion

ADAPT transfers analog layout tuning experience across circuits and PDKs by decoupling topology, technology, and parameter semantics, learning a neural response prior, and adapting a context-calibrated residual GP to target observations. On five 14 nm benchmarks, it achieves the best reported HV@20, HV@40, AUC-HV, and IGD$^+$, reaching the best baseline HV@40 in 12.5 evaluations on average—a more than 3× reduction in evaluation count.

AI USE STATEMENT

Generative AI tools were used only to edit the manuscript text and assist with LaTeX formatting, including migration to the ICLR 2027 template. They were not used to develop research ideas, implement research code, conduct experiments, or analyze results. The authors take responsibility for the final content of the manuscript.

REFERENCES

Sebastian Ament, Samuel Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. Unexpected improvements to expected improvement for bayesian optimization. *Advances in neural information processing systems*, 36:20577–20612, 2023.

Davide Basso, Luca Bortolussi, Mirjana Videnovic-Misic, and Husni Habal. Fast ml-driven analog circuit layout using reinforcement learning and steiner trees. In *2024 20th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, pp. 1–4. IEEE, 2024.

David R Burt, Carl Edward Rasmussen, and Mark Van Der Wilk. Convergence of sparse variational inference in gaussian processes regression. *Journal of Machine Learning Research*, 21(131): 1–63, 2020.

Guojin Chen, Keren Zhu, Seunggeun Kim, Hanqing Zhu, Yao Lai, Bei Yu, and David Z Pan. Llm-enhanced bayesian optimization for efficient analog layout constraint generation. *arXiv preprint arXiv:2406.05250*, 2024.

Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Parallel bayesian optimization of multiple noisy objectives with expected hypervolume improvement. *Advances in neural information processing systems*, 34:2187–2200, 2021.

Zehao Dong, Weidong Cao, Muhan Zhang, Dacheng Tao, Yixin Chen, and Xuan Zhang. Cktgnn: Circuit graph neural network for electronic design automation. *arXiv preprint arXiv:2308.16406*, 2023.

Hao Geng, Tinghuan Chen, Yuzhe Ma, Binwu Zhu, and Bei Yu. Ptpt: Physical design tool parameter tuning via multi-objective bayesian optimization. *IEEE transactions on computer-aided design of integrated circuits and systems*, 42(1):178–189, 2022a.

Hao Geng, Qi Xu, Tsung-Yi Ho, and Bei Yu. Ppatuner: Pareto-driven tool parameter auto-tuning in physical design via gaussian process transfer learning. In *Proceedings of the 59th ACM/IEEE Design Automation Conference*, pp. 1237–1242, 2022b.

Chanhee Jeon and Taewhan Kim. Reshaping bayesian optimization for design space optimization towards accurate and irredundant evaluation in eda tool parameter exploration. In *2026 Design, Automation & Test in Europe Conference (DATE)*, pp. 1–7. IEEE, 2026.

Kishor Kunal, Meghna Madhusudan, Arvind K. Sharma, Wenbin Xu, Steven M. Burns, Ramesh Harjani, Jiang Hu, Desmond A. Kirkpatrick, and Sachin S. Sapatnekar. Invited: Align - open-source analog layout automation from the ground up. In *2019 56th ACM/IEEE Design Automation Conference (DAC)*, pp. 1–4, 2019.

Mingjie Liu, Keren Zhu, Xiyuan Tang, Biying Xu, Wei Shi, Nan Sun, and David Z Pan. Closing the design loop: Bayesian optimization assisted hierarchical analog layout synthesis. In *2020 57th ACM/IEEE design automation conference (DAC)*, pp. 1–6. IEEE, 2020.

Donger Luo, Qi Sun, Qi Xu, Tinghuan Chen, and Hao Geng. Attention-based eda tool parameter explorer: From hybrid parameters to multi-qor metrics. In *2024 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1–6, 2024. doi: 10.23919/DATE58400.2024.10546562.

Donger Luo, Qi Sun, Peng Xu, Su Zheng, Qi Xu, Tinghuan Chen, Bei Yu, and Hao Geng. Attention-based eda tool parameter explorer: From hybrid parameters to multi-qor metrics. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2025.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Hanrui Wang, Kuan Wang, Jiacheng Yang, Linxiao Shen, Nan Sun, Hae-Seung Lee, and Song Han. Gcn-rl circuit designer: Transferable transistor sizing with graph neural networks and reinforcement learning. In *2020 57th ACM/IEEE Design Automation Conference (DAC)*, pp. 1–6. IEEE, 2020.

Zhiyao Xie, Guan-Qi Fang, Yu-Hung Huang, Haoxing Ren, Yanqing Zhang, Brucek Khailany, Shao-Yun Fang, Jiang Hu, Yiran Chen, and Erick Carvajal Barboza. Fist: A feature-importance sampling and tree-based method for automatic design flow parameter tuning. In *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 19–25. IEEE, 2020.

Biying Xu, Keren Zhu, Mingjie Liu, Yibo Lin, Shaolan Li, Xiyuan Tang, Nan Sun, and David Z. Pan. Magical: Toward fully automated analog ic layout leveraging human and machine intelligence: Invited paper. In *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1–8, 2019. doi: 10.1109/ICCAD45719.2019.8942060.

Peng Xu, Su Zheng, Yuyang Ye, Chen Bai, Siyuan Xu, Hao Geng, Tsung-Yi Ho, and Bei Yu. Ranktuner: When design tool parameter tuning meets preference bayesian optimization. In *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*, pp. 1–7, 2024.

Peng Xu, Yapeng Li, Tinghuan Chen, Tsung-Yi Ho, and Bei Yu. Kclnet: Electrically equivalence-oriented graph representation learning for analog circuits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 1355–1363, 2026a.

Peng Xu, Ziyang Yu, Yuan Pu, Xinyun Zhang, Donger Luo, Hao Geng, Siyuan Xu, Tsung-yi Ho, and Bei Yu. Ratuner: Retrieval-augmented vlsi flow design parameter tuning framework. In *2026 Design, Automation & Test in Europe Conference (DATE)*, pp. 1–7. IEEE, 2026b.

Ziyang Yu, Peng Xu, Su Zheng, Siyuan Xu, Hao Geng, Bei Yu, and Martin Wong. Causaltuner: Will causality help high-dimensional eda tool parameter tuning. In *2026 31st Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 370–376. IEEE, 2026.

Su Zheng, Hao Geng, Chen Bai, Bei Yu, and Martin DF Wong. Boosting vlsi design flow parameter tuning with random embedding and multi-objective trust-region bayesian optimization. *ACM Transactions on Design Automation of Electronic Systems*, 28(5):1–23, 2023.

A Dataset and Experimental Protocol

A.1 Dataset composition and observation units

The released dataset contains 40 circuit entries with registered lineage identifiers, each instantiated in 14 nm, 65 nm, and SKY130. It spans amplifiers, oscillators, comparators, reference buffers, switched-capacitor circuitry, charge pumps, and mixed-signal interface blocks. Fifteen entries originate from ALIGN examples or its composed OTA library; 25 originate from external open-source collections. Each of the 120 tasks retains the same fixed number of requested configurations, stored as one record set per task. A record is a requested tool configuration, including its outcome when execution does not produce an evaluable layout.

We distinguish three nested observation levels. G requires a complete bound full flow and four finite post-route metrics. C additionally requires zero opens, shorts, and unresolved connectivity/checker diagnostics. L further requires zero DRC violations under the frozen checker. Thus $L \subseteq C \subseteq G$; these counts are not additive. Geometric prediction can use metric-valid G observations, whereas the reported optimization fronts use C observations. Unavailable metrics and unperformed checks remain missing rather than zero. Table 6 reports the merged observation counts.

Table 6: Dataset observation counts per PDK. G, C, and L are nested quality levels; counts are not additive across levels.

PDK	Tasks	G	C	L
14 nm	40	1,256	750	706
65 nm	40	1,011	883	419
SKY130	40	1,493	731	344

A.2 Aligned sampling and execution records

Each task's requests combine two equally sized, separately scrambled Sobol blocks in 35 dimensions. The first block is shared across all tasks. The second is specific to the registered circuit lineage and shared across its three PDKs. The global seed is 20260919. A lineage seed is the unsigned integer represented by the first eight hexadecimal digits of SHA256 of the compact UTF-8 JSON list `[method,20260919,"design_specific",lineage_id]`, where `method` identifies the versioned two-block sampling scheme. Each block is drawn with SciPy 1.13.1 `Sobol(d=35,scramble=True,bits=30,seed=seed)` via `random_base2` under NumPy 1.26.4 and Python 3.12.2; the block-size exponent is fixed and recorded in the release manifest. These are two separately initialized blocks, rather than a prefix of one longer scrambled sequence.

For coordinate j, a Sobol component $u_j \in [0, 1)$ selects $x_j = D_j[\lfloor u_j | D_j | \rfloor]$ from the frozen, ordered candidate list. No interpolation is used. All requests are frozen before execution, and ALIGN uses seed 1. Cross-PDK alignment applies to requested actions; object availability and PDK-specific decoding determine the effective native inputs. Failures remain at their original sample indices, without replacement sampling.

The merged view also retains a bounded placement-budget extension: slots whose original executions terminated at the external placement limit of 1,200 s were rerun once with a limit of 3,600 s, with inputs, actions, seeds, and tool versions unchanged. A minority of these reruns produced G observations and supply the complete selected records for those same slots; all other slots retain their original records. Original and retry receipts, budget identifiers, and cumulative execution costs are retained separately. The execution ledger additionally records a small number of early infrastructure-level attempts; it does not increase the per-task sampling budget. The merged release contains no aliased records.

A.3 Circuit inventory and scale

Tables 7 and 8 list all circuit entries in descending MOS count. Source identifiers are retained for unambiguous lookup. MOS counts expand hierarchy but not device multiplicity or finger count; total device counts additionally include passive devices. Net counts use the flattened source graph,

and hierarchy depth is zero for a flat source circuit. The inventory ranges from 20 to 156 MOS instances and 14 to 143 nets. The H column marks the 21-circuit historical fitting cohort used in the reported runs, rather than an additional dataset.

Source codes are A: ALIGN-public examples; U: ALIGN composed OTA library; S: ACST-generated operational amplifiers; M: MAGICAL-CIRCUITS; F: `fredysolis/caravel_analog_fulgor`; B: `Seungmin0825/AnalogToBi`; J: `jialinlu/opensource-analog-circuits`; V: `maherbenhouria/caravel_user_project_analog`; O: `idea-fasoc/OpenFASOC`. The release manifest retains each original netlist identifier and registered lineage identifier.

Table 7: Full circuit inventory (1/2). Src.: source code; Dev.: total devices; h: hierarchy depth; H: historical fitting cohort.

Source circuit identifier	Src.	MOS	Dev.	Nets	h	H
TEST_VGA	A	156	156	143	2	yes
VCO_DTYPE_12_HIERARCHICAL	A	144	144	115	2	yes
res_amp_lin_prog	F	128	128	71	3	yes
RRR1	M	62	62	27	2	yes
1001_full	B	56	60	32	2	–
PFD	F	56	56	32	2	yes
1067_full	B	53	53	40	1	yes
UNITY_GAIN_BUFFERS	A	50	50	49	1	yes
TWO_STAGE_FULLY_DIFFERENTIAL_OP_AMP_40_12	S	46	50	35	0	–
440_full	B	46	47	30	2	–
TWO_STAGE_FULLY_DIFFERENTIAL_OP_AMP_48_6	S	45	49	36	0	–
TWO_STAGE_FULLY_DIFFERENTIAL_OP_AMP_36_12	S	44	48	33	0	–
TWO_STAGE_FULLY_DIFFERENTIAL_OP_AMP_44_6	S	43	47	34	0	–
TWO_STAGE_FULLY_DIFFERENTIAL_OP_AMP_12_12	S	42	46	33	0	–
TWO_STAGE_FULLY_DIFFERENTIAL_OP_AMP_45_12	S	41	45	32	0	–
TWO_STAGE_FULLY_DIFFERENTIAL_OP_AMP_42_6	S	40	44	32	0	–
TWO_STAGE_FULLY_DIFFERENTIAL_OP_AMP_28_6	S	39	43	31	0	–
REFERENCE_BUFFER_CORE	M	39	39	19	1	yes
SONG_DACFC_PIN_3	J	37	39	26	1	–
TELESCOPIC_OTA_WITH_BIAS	A	36	36	35	0	yes

Table 8: Full circuit inventory (2/2). Src.: source code; Dev.: total devices; h: hierarchy depth; H: historical fitting cohort.

Source circuit identifier	Src.	MOS	Dev.	Nets	h	H
div_by_2	F	34	34	21	3	yes
Netlist	A	30	35	23	1	–
COMPLEMENTARY_OP_AMP32	S	28	29	22	0	–
COMPARATOR_1TO7_0P7_LVT	M	28	28	15	0	yes
COMPLEMENTARY_OP_AMP16	S	27	28	21	0	–
COMPARATOR	V	26	26	17	0	yes
single_ended_current_mirror_pmos_LG_npmos_cr4_2	U	25	25	19	1	yes
SINGLE_ENDED_CURRENT_MIRROR_LG_NPMOS_CR4_2	U	25	25	18	1	yes
LEUNG_NMCF_PIN_3	J	24	30	22	1	–
SYMMETRICAL_OP_AMP151	S	24	25	21	0	–
DCDC_CONV2TO1	O	24	24	21	2	yes
SYMMETRICAL_OP_AMP144	S	23	24	20	0	–
single_ended_miller_compensated_pmos_LG_npmos_cr4_2	U	23	24	18	1	–
SINGLE_ENDED_MILLER_COMPENSATED_LG_NPMOS_CR4_2	U	23	24	17	1	–
single_ended_low_voltage_cascode_pmos_LG_npmos_cr4_2	U	23	23	20	1	yes
SINGLE_ENDED_LOW_VOLTAGE_CASCODE_CURRENT_MIRROR_LG_NPMOS_CR4_2	U	23	23	19	1	yes
SINGLE_ENDED_PMOS_LG_NPMOS_CR4_2	U	21	21	17	1	yes
FULLY_DIFFERENTIAL_LG_NPMOS_CR4_2	U	21	21	16	1	yes
CASCODE_CURRENT_MIRROR_OTA	A	20	20	20	0	yes
Netlist_comparator	A	20	20	14	1	yes

B Construction of the Shared Conditional Parameter Space

B.1 Scope and design rationale

The parameter vector $\mathbf{x}$ in our formulation must have a consistent interpretation across circuit graphs G and process design kits (PDKs) p. Using absolute metal-layer numbers or device dimensions as shared coordinates would conflate an optimization decision with technology-specific units and capabilities. Conversely, allocating one coordinate to every net or primitive would make the vector length depend on the circuit. We therefore define a *shared conditional space*: the coordinate order, structural group semantics, and finite action domains are fixed, while a task-specific decoder maps each action to supported native ALIGN inputs.

ALIGN provides hierarchical modules, constraints on named nets, and primitive generators Kunal et al. (2019); it does not natively provide the fixed groups defined here. In particular, `NetPriority.weight` is a scalar for a specified net set, rather than a single global degree of freedom. Broadcasting one weight to all eligible nets is a deliberate restriction on independently addressable objects. The same distinction applies to module constraints and MOS generator parameters. Our construction retains independent decisions *between* fixed structural groups and shares decisions *within* each group. It does not preserve arbitrary per-object control: no fixed finite vector can provide independent values for an unbounded number of objects.

The versioned construction, `adapt-formal-grouped35-v1`, retains **20 parameter families and 35 scalar coordinates** from an audited pool of 32 families. Its dimension is

$$d = \underbrace{11}_{\text{global/stage}} + \underbrace{5 \times 2}_{\text{module}} + \underbrace{1 \times 4}_{\text{net}} + \underbrace{4 + 4 + 2}_{\text{primitive}} = 35. \tag{15}$$

All coordinates have finite numerical domains, including controls whose native meaning is continuous. Writing D_j for the ordered candidate list of coordinate j, the request space is $\mathcal{X} = \prod_{j=1}^{35} D_j$. Before masks and joint constraints, it contains $4{,}333{,}224{,}817{,}852{,}416$ requests. This Cartesian product is a space of *requests*, not a certificate that every request produces a valid layout. Earlier eight-dimensional functional-zone and 32-family shared-value parameterizations are distinct versions; their optimization results cannot be attributed to this 35-coordinate construction.

B.2 Deterministic structural grouping

Let c denote the frozen task context: the source netlist and lineage, selected top module, compiled hierarchy, original constraints and supply intent, PDK resources, and compiler/library/adapter versions. For family ℓ, let $\mathcal{O}_\ell(c)$ be its eligible native objects and $g_\ell(o; c) \in \{1, \ldots, K_\ell\}$ the group of object o. We first assign group requests and then decode the complete request jointly:

$$r_{\ell,o} = x_{\ell,g_\ell(o;c)}, \quad o \in \mathcal{O}_\ell(c), \qquad \boldsymbol{\theta}_c = \Delta_c(\mathbf{r}; c). \tag{16}$$

Here $x_{\ell,k}$ denotes the corresponding coordinate of $\mathbf{x}$, $\mathbf{r}$ collects the per-object requests, and $\boldsymbol{\theta}_c$ is the native configuration. The joint decoder Δ_c can reject a combination; it is not a set of independent, always-feasible scalar maps. Grouping and capability checks are separate: having a group label does not prove that a particular generator consumes every parameter in that group. Membership is established before optimization from structure and original constraints, without inspecting final area, wirelength, quality labels, or optimization gains.

Module-based controls: two groups. The ordered groups are `ROOT` and `DESCENDANT`. `ROOT` contains the specified compiled top-module definition; `DESCENDANT` contains all other reachable, nonempty layout/routing module definitions. Unreachable library definitions are excluded. A flat circuit has an empty descendant group but retains its coordinates. A module master instantiated at several locations receives one group-level policy; independent control of its occurrences would require cloning and a different parameterization. This grouping distinguishes top-level assembly from internal assembly without assuming an identical number of amplifier stages or hierarchy levels across circuits.

Net-based controls: four groups. A net is identified by its module definition and local actual-net identifier. Within each reachable module, define its structural degree as

$$d(n) = \left|\{(i,t) : \text{instance } i \text{ maps formal terminal } t \text{ to actual net } n\}\right|. \tag{17}$$

This counts distinct logical block endpoints, not pin rectangles or repeated occurrences of a module master. A module interface port is not itself an additional endpoint in this count; nets with one block endpoint are retained, since the PnR representation can add a module terminal later.

First form components S by taking the transitive closure of original `SymmetricNets` relations, with unpaired nets as singleton components. For each component, define

$$b(S) = \bigvee_{n \in S} \mathbf{1}[n \text{ is a module interface port}], \qquad d(S) = \max_{n \in S} d(n). \tag{18}$$

The ordered groups are `PORT_LOW`, `PORT_BRANCH`, `INTERNAL_LOW`, and `INTERNAL_BRANCH`: `PORT` means $b(S) = 1$, and `BRANCH` means $d(S) \geq 3$. Using the maximum, rather than the sum, avoids classifying a pair of low-degree symmetric nets as branching merely because they were grouped. All eligible members of a component receive the same group multiplier.

A component containing an explicitly identified supply/ground net or an original `DoNotRoute` net is protected as a whole. Supply roles follow specified intent and formal–actual propagation, rather than name heuristics. Missing supply intent is an input-specification issue. Zero-block-endpoint ports are recorded without inventing an addressable native target. Every remaining eligible net belongs to exactly one of the four groups, and

$$w_n = w_n^{\text{ref}} \alpha_{g(n;c)}, \qquad \alpha_k \in \{1, 2, 4, 8\}. \tag{19}$$

Within each group, ratios of nonzero original weights are preserved, and a zero original weight remains zero. Distinct resulting weights require separate native assignments even within one group. This rule preserves symmetry components; it does not infer signal direction, analog criticality, or a unique input-to-output critical path. Module-local representations of one hierarchical electrical net retain their provenance and are not counted as independent physical nets.

Primitive-based controls: four groups, or two for diffusion sharing. Height and shape use the ordered groups `N_SINGLE`, `P_SINGLE`, `N_MULTI`, and `P_MULTI`. Polarity follows the model's `base` ancestry to NMOS or PMOS. `SINGLE` denotes one electrical MOS element in the compiled MOS primitive, and `MULTI` denotes two or more elements of the same polarity; finger count is not the number of electrical elements. The audited generator protocol covers one- and two-element primitives. Larger groups have a structural label but require separate consumer qualification. Unknown or mixed model polarity is not silently classified as an ordinary inactive coordinate. Our model-ancestry rule must also agree with the actual native N/P dispatch; it is not a claim that ALIGN implements this classification itself.

`shared_diff` uses only two groups, ordered NMOS then PMOS, restricted to supported two-element primitives with identical models, identical full electrical parameter dictionaries, and the same original source net. Explicit original diffusion-sharing requirements remain fixed. No sources are rewired to make a pair eligible, and no dimensions are added for single-device primitives that cannot satisfy this eligibility rule. As with modules, primitive definitions shared by several instances receive one policy, rather than independent occurrence-level values.

Invariance and control granularity. These rules use neither hand-labeled bias/input/output stages nor an outcome-dependent top-k selector. They are invariant to consistent renaming and list reordering of the same graph. They are not invariant to arbitrary changes in hierarchy or compiler template matching. Compiler and library versions must therefore be fixed, and each task's actual group membership must be recorded. Cross-PDK consistency means identical group *definitions*, not identical objects, counts, native values, or effects.

B.3 Coordinate order, domains, and native interpretation

Tables 9 and 10 specify every coordinate. Intervals in the index column are inclusive; each position is one scalar, not a variable-length vector. The listed reference is repeated for each group of that family. All candidate sets are ordered as printed; no unlisted intermediate values are implied. Boolean diffusion-sharing choices are encoded numerically as 0/1. The reference denotes our frozen *experimental policy*, not necessarily ALIGN's original default.

Table 9: Placement and primitive coordinates (1–21). Module2, Net4, MOS4, and Pair2 use the group orders in Section B.2. Generator fields abbreviate `Generator.parameters`.

Index	ID	Native field	Candidates	Ref.	Scope and interpretation
1	P01	`lambda_coeff`	$\{0.1, 1\}$	1	Global placement objective coefficient.
2–3	P04	`HorizontalDistance.abs_distance`	$\{0, 1\}$	0	Module2: extra horizontal spacing tracks.
4–5	P05	`VerticalDistance.abs_distance`	$\{0, 1, 4\}$	0	Module2: extra vertical spacing tracks.
6–7	P07	`Boundary.halo_vertical`	$\{0, 4\}$	0	Module2: extra vertical halo tracks per side.
8–11	P08	`NetPriority.weight`	$\{1, 2, 4, 8\}$	1	Net4: multiplier of each original net weight.
12–15	G01	`height`	$\{0, 1, 2\}$	0	MOS4: legal height increments.
16–19	G02	`legal_sizes`	$\{0, 0.5, 1\}$	0.5	MOS4: input-derived shape quantile.
20–21	G04	`shared_diff`	$\{0, 1\}$	1	Pair2: disable/enable eligible diffusion sharing.

Table 10: Signal-routing and power coordinates (22–35). R01/R02 are Module2 controls; all remaining rows are global or native-stage shared. C++ fields identify actual consumers, not additional public ALIGN CLI flags.

Index	ID	Native field	Candidates	Ref.	Interpretation
22–23	R01	`Route.min_layer`	$\{0, 0.5\}$	0	Conditional lower signal-layer quantile.
24–25	R02	`Route.max_layer`	$\{0, 0.5, 1\}$	1	Conditional upper signal-layer quantile.
26	R05	`ST_number`	$\{3, 5, 7\}$	5	Steiner-tree candidate budget.
27	R06	`UpdateEdgeWeight::times`	$\{1, 2, 4\}$	2	Used graph-edge cost multiplier.
28	R07	`scale_number`	$\{1, 1.5, 3\}$	1.5	Plate-related capacity deduction factor.
29	R09	`sym_factor`	$\{0, 0.5, 1, 2\}$	1	Normalized symmetry-reference A* penalty.
30	W01	`h_skip_factor`	$\{2, 4, 7, 12\}$	7	Horizontal power-grid skip factor.
31	W02	`v_skip_factor`	$\{2, 4, 8, 12\}$	8	Vertical power-grid skip factor.
32	W03	`bottom_power_grid_layer`	$\{0.5, 1\}$	0.5	Conditional lower power-grid quantile.
33	W08	`via_expand_effort`	$\{0.5, 1, 4\}$	1	Normalized power-hookup via penalty.
34	W09	`SetSrcDest::src_index_number`	$\{1, 3, 7\}$	7	Cap on target-distance records.
35	W10	`SetSrcDest::expand_scale`	$\{10, 24\}$	10	Power-hookup search-window margin factor.

Why some consumers remain stage shared. R05, R06, R07, R09, and the power controls operate internally on nets, edges, or pins, but their audited interfaces expose shared algorithmic coefficients rather than separately addressable per-net policies. For example, R05 sets the Steiner candidate budget for applicable calls, while W09 limits records considered by the power-hookup procedure. Grouping these coefficients by selected nets would require another native interface and would define a different action space. P01 likewise remains a run-level placement coefficient. The R05 and R07 consumers are `GcellGlobalRouter` and `GlobalGrid::AdjustPlateEdgeCapacity`, respectively; R09 belongs to the symmetry-reference A* routine and W08 to the power A* routine.

Physical increments and shape choices. For placement spacing/halo control ℓ, a module-specific baseline distance $b_{\ell,o}$ and directional placement pitch $\pi_{\ell,o}$ give the target in native physical units:

$$t_{\ell,o} = \lceil b_{\ell,o}/\pi_{\ell,o} \rceil \, \pi_{\ell,o} + k_{\ell,g_\ell(o)}\pi_{\ell,o}. \tag{20}$$

The adapter then serializes the appropriate constraint units; distance and halo fields do not share the same serialization scale. G01 instead uses $h_o = h_o^{\mathrm{ref}} + k_{g(o)}\Delta h_o$, with legal height step Δh_o from the generator context. Thus a zero increment is not a raw zero height.

Table 11: PDK context used to interpret shared actions. Descriptors are fixed inputs, not additional optimization coordinates.

Context	Representative information	Role in the conditional space
Metal resources	Ordered layers, routing directions, pitch, width, and spacing.	Define track units and supported signal/power layer windows.
Via resources	Adjacent-layer connectivity, cut geometry, enclosure and spacing rules.	Constrain layer transitions and provide resource/cost context.
MOS capabilities	Model ancestry, generator branch, legal height increments, and electrical factorization.	Resolve polarity, height/shape mappings, and family-specific applicability.
Supply interface	Explicit supply/ground intent and actual pin-layer access.	Establish the ordinary power-grid and hookup context; missing evidence remains unresolved.

For G02, let $\mathcal{T}_o = ((a_1, b_1), \ldots, (a_{n_o}, b_{n_o}))$ be the input-derived legal integer factor pairs, filtered by original hard shape constraints and sorted by the exact ratio a/b, then by (a, b). A request q selects

$$r(q, n_o) = \left\lfloor q(n_o - 1) + \tfrac{1}{2} \right\rfloor, \qquad \mathrm{shape}_o(q) = \mathcal{T}_o[r(q, n_o) + 1]. \tag{21}$$

The rank is zero-based; the displayed table indexing is one-based. The selected pair is serialized as a singleton `legal_sizes` list. The factors use the generator convention before its two-device column expansion. This selects a *singleton* legal-size list, a restricted subset of the native list-valued interface, not two independent real-valued shape controls. Height, shape, and diffusion sharing must be composed in one primitive payload. An invalid combination is rejected; the table is not pruned and reranked in response to that action's outcome.

Conditional layer and cost decoding. R01/R02 are decoded jointly in a module-specific set of admissible layer windows, respecting ordered metals, via connectivity, required pin access, and original hard routing constraints. Coordinates are decoded in their fixed order: each quantile selects a rank among sorted native values still compatible with the selected prefix, using the rounding rule in Eq. equation 21. The power decoder similarly selects a compatible window tuple. Hence equal quantiles need not select equal metal names in different PDKs, and two requested quantiles can alias in a small resource set. Layer/pin inventories require explicit provenance from the specified setup or preceding stage; missing evidence is not an unconstrained routing window. Neither group labels nor candidate order is revised using final objective values. Even a window that passes these resource checks can fail during routing.

R09 scales a frozen reference routing-cost unit; W08 scales a frozen pitch-based via-cost unit. Their listed numbers are normalized policy coefficients, not the raw C++ values. W01/W02 and W03 address ordinary power grid fields in run-local `design_info`; the audited power protocol requires one explicitly identified power/ground pair. Multiple independent supply domains require separate qualification, rather than inference from these results.

Technology context versus actions. Table 11 distinguishes fixed technology descriptors from tunable actions. The 35 coordinates select supported policies without changing PDK rules; they do not specify the response model's feature-vector width.

B.4 Masks, equivalence, and execution evidence

For a task with established capability, let $\mathbf{m} \in \{0, 1\}^{35}$ be its structural/capability mask. The numerical contract stores an inactive coordinate as a zero sentinel, $\bar{x}_j = 0$ when $m_j = 0$, and keeps $\bar{x}_j = x_j \in D_j$ otherwise. Decoding restores the reference policy:

$$\widetilde{x}_j = \begin{cases} \bar{x}_j, & m_j = 1, \\ x_j^{\mathrm{ref}}, & m_j = 0. \end{cases} \tag{22}$$

The stored vector $\bar{\mathbf{x}}$ need not itself belong to $\mathcal{X}$, because the inactive sentinel can be outside D_j. Equation equation 16 is evaluated using the restored $\widetilde{\mathbf{x}}$, rather than the sentinel vector or an unmasked

request for an inactive coordinate. A separate mask is essential: zero is also a legitimate active choice for several families. The sentinel is never forwarded as a native field value. Ineligible objects retain their original constraints; an inactive group retains the reference policy, including any singleton-shape reference. A sampling record may additionally preserve the unmasked raw request.

An empty group, a proven missing consumer or hard-fixed field, and a provably single effective payload are distinct inactivity reasons. An unresolved model, missing capability evidence, or unknown observer result must remain explicitly unresolved rather than being silently converted to permanent inactivity. Likewise, a consumer that performs zero operations under the reference placement may operate under another configuration. An unchanged objective does not prove inactivity. We record object existence, decoded alternatives, serialized bindings, observed consumer operations, and metric availability separately; a single Boolean mask is not evidence for all five properties.

Two requests may be equivalent after masking and conditional decoding. Reuse requires equality of the complete native configuration and execution identity—the same task inputs, PDK, runtime, seed, and resource budget—and a completed source receipt. Equal metrics or equal group labels do not prove this equivalence. Aliases are not independent observations for training or for reporting sample counts.

Binding is an execution requirement. P01 maps to `PlacerHyperparameters.LAMBDA`. Module constraints and net weights must be written into the compiled topology before the relevant PnR restart; editing only an unused input JSON does not establish an effect. P08 follows the native chain from `NetPriority` through `CritNet.priority` to `node.Nets.weight` and placement. Primitive controls must reach each eligible definition's `Generator.parameters` before generation. Module routing windows must reach the router, and the shared C++ controls require the versioned native-configuration bridge used by the audited runtime. These internal coefficients are not claimed to be stock command-line options. Explicit-PDN renaming requires a verified module/instance and formal–actual projection when reading bindings back. A reproducible receipt traces each request through its group, native target, serialized value, and consumer.

B.5 Value screening and the limits of validation

Construction-stage evidence. The v1 screening audit reused 899 canonical action records from the earlier shared-family campaigns, together with deduplicated reference runs. Six registered interaction studies supplied 216 referenced physical runs, some of which overlap the single-axis records; these counts must not be added as independent samples. The principal grouping examples are a high-speed comparator (HSC) and a cascode current-mirror OTA (CM), each in the audited 14 nm and 65 nm mock PDKs. These are engineering development/qualification cases, not evidence of foundry signoff or untouched-task generalization.

Screening policy. A tested active value associated with a confirmed failure—such as no placement, rejected constraints, a primitive assertion, or observed connectivity/DRC errors—was excluded from the v1 domain. Unresolved process termination (e.g., an unexplained kill) was recorded separately and its actively tested value withheld, not labeled physically impossible. Untested interior values were not inferred safe from successful endpoints. A family remained tunable only if at least two retained choices could produce different consumer inputs on some eligible objects. This does not require every coordinate to affect every task or to improve an objective.

This conservative engineering filter is not a causal proof that an excluded value always fails. A bad matched reference weakens attribution even when the action itself exhibits an actual error. Unmodified reference coordinates are not all blacklisted because one complete configuration fails. The v1 policy retained some reference values with repeated successful evidence elsewhere, while preserving their unresolved profile/seed records. For a 2×2 interaction, failure only at the joint corner does not establish that both individual values are invalid. For example, the P02/P04 study had a failing P02-only corner and a successful P04-only corner; P04 was retained.

Removed families and fixed background. The twelve omitted families are P02/P03 (aspect-ratio bounds), P06 (horizontal halo), G03 (pattern), G05 (minimum vias), R03 (multiple connections), R04 (coarse routing grid target), R08 (detail routing grid scale), and W04–W07 (three power-layer coordinates and power width). Omission must also define the background: existing hard aspect-ratio

Table 12: Examples of evidence-limited screening. Values refer to the audited normalized policies unless a native field is explicitly named. Exclusion from v1 is not a claim that ALIGN can never support the family.

Control	Observed evidence	v1 consequence
G01 height	Increment $k = 4$: no INV placement in HSC14 across three seeds.	Remove 4; retain $\{0, 1, 2\}$.
G05 minvias	Native counts 2/3: 24 primitive-extraction assertions across four cells.	Remove those values; omit the family with only its default remaining.
R08 grid scale	Native scale 2: 12 runs with opens.	Omit the family; preserve native scale 1.
G03 pattern	No two distinct effective patterns in the audited panel.	Defer the family; do not label all native patterns unusable.
W03 power window	24/24 action runs clean for the two tuples $(q, 1, 0, 1)$, $q \in \{0.5, 1\}$.	Retain the two tuples; not 24 valid causal comparisons or a joint-space guarantee.

and boundary constraints are preserved; no new pattern, minimum-via, multiple-connection, detail-grid-scale, or power-width override is applied. Absent explicit aspect-ratio constraints, the reference bounds are 0.1 and 10. The R04 grid-target override is *unset*; the old value 16 is not ALIGN's native default and must not remain as a hidden intervention. The power tuple is fixed to

$$(q_{\mathrm{W03}}, q_{\mathrm{W04}}, q_{\mathrm{W05}}, q_{\mathrm{W06}}) = (q, 1, 0, 1), \qquad q \in \{0.5, 1\}, \tag{23}$$

with reference $(0.5, 1, 0, 1)$. Successful fields from different power or primitive reference profiles cannot be combined and relabeled as an already validated common baseline.

What the static checks establish. The construction audit verified agreement of the 35-coordinate table and machine schema, reference-vector round trips, rejection of five representative excluded/withheld values, and restoration of the same reference policy under four provisional masks. Module/net renaming and list-permutation checks passed on those four cases. All 22 inspected MOS definitions had resolved model ancestry. For 21 eligible shape definitions, the input-factor enumeration and ordering matched the earlier audited tables. The provisional available-coordinate counts were 34/31/22/21 for HSC14/HSC65/CM14/CM65, respectively; these are not counts of dimensions proven to change post-route metrics. Some historical capability snapshots used primitive artifacts, so these counts are not an independent source-only qualification of every new task.

What remains separate. The 11 September 2026 construction report and its numerical contract did *not* run ALIGN or implement a per-object dispatcher. Subsequent execution adapters and full-flow receipts are separate evidence: the later Dataset2 startup acceptance records complete five-stage, binding, and metric paths on 14 nm, 65 nm, and SKY130 examples. This establishes those observed paths, not universal success over their PDK domains. Neither the static contract nor old shared-value runs certify unequal group assignments or the full joint Cartesian product. In particular, old P08 selected a subset of nets, whereas the new rule partitions all eligible nets; setting all four new multipliers equal does not reproduce that old selector.

The v1 primitive-adapter evidence covered 14 nm and 65 nm. The 45 nm records available to that construction audit included static device checks and runs with physical errors, rather than a qualified grouped35 baseline. This is a boundary of that evidence, not a proof that 45 nm cannot run ALIGN. A new PDK needs its own verified model mapping, generator consumers, resource decoder, and complete runtime receipts. Its technology name or a GDS file alone is insufficient.

B.6 Reproducibility and implications for transfer evaluation

The artifact directory `docs/param/` separates the construction's evidence: `adapt_formal_grouped_space_v1.json` fixes indices, domains, references, and background policies; `grouped_space_contract.py` validates numerical requests; and `_audit_20260911/strict_value_audit.json` records screening decisions and source runs. The module/net and primitive inventories retain membership, exclusions, and shape tables. `contract_validation_20260911.json` reports zero ALIGN runs in that review; later execution evidence is separately archived under `dataset2/data_generation/acceptance/`.

Each observation must retain source/lineage, input/PDK/runtime hashes, requests and masks, background version, object bindings and consumer evidence, seed/budget, failure stage, and metric/checker availability. Process completion, complete routing, geometric observability, connectivity, and DRC acceptance are distinct outcomes. Missing metrics or unperformed checks remain unavailable, not zero. Geometry-only observations are not connection-qualified optimization rewards. Results from different action or objective versions require explicit provenance rather than automatic pooling.

Grouping does not read objective values, but empirical domain screening uses pilot failures. Those pilot lineages and circuit–PDK pairs are therefore development data, not completely unseen tests; lineage-related variants must share a design split. Access to target-PDK rules for decoding must be separated from access to target objectives for method selection. Comparisons must share groups, domains, decoders, mask rules, query budgets, and failure handling. Aligned coordinates alone do not establish transfer gains or independent effects: primitive height/shape/sharing, module spacing/halo, and routing/power policies remain coupled, and global placement and net weights can overlap. These effects require matched runtime evidence.

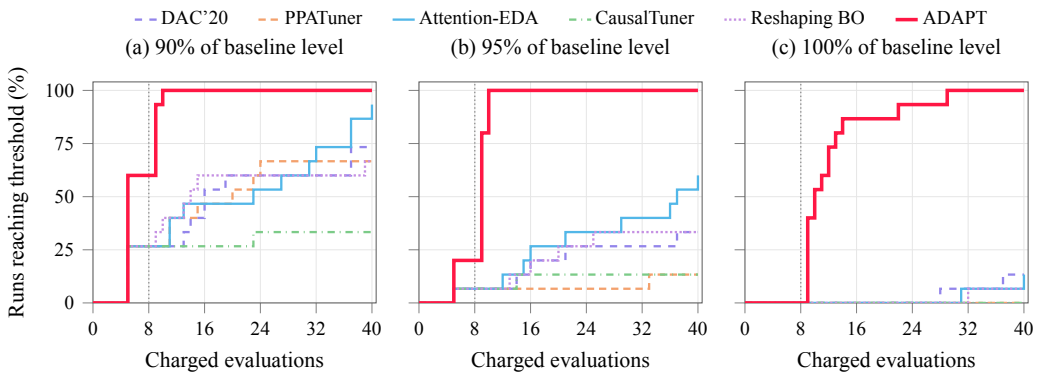

Figure 7: Quality attainment on the five 14 nm benchmarks. Each curve is the fraction of 15 runs (five benchmarks, three optimizer seeds) that have reached the indicated fraction of the best budget-complete baseline HV on the same benchmark. Methods with three recorded seeds are shown; NR runs remain in the denominator. The dotted vertical line marks the end of ADAPT's eight initial evaluations. Curves are empirical proportions, not confidence bands.

C Evaluation Efficiency at Matched Quality

We complement the HV trajectories in Figure 5 with a retrospective analysis of the evaluations required to reach fixed quality levels. The analysis covers the same five 14 nm benchmarks: CLK-CMP, PT-CMP, CT-CMP, DC-Switch, and REF-BUF.

Quality thresholds and accounting. For benchmark c, let $H_{m,c,s}(n)$ denote the cumulative HV of method m with optimizer seed s after n charged evaluations. We use the same source-only normalization, reference point $(1.1, 1.1, 1.1, 1.1)$, and connectivity-qualified Pareto fronts as in the main comparison. Define

$$b_c = \max_{(m,s)\in\mathcal{B}_c^{40}} H_{m,c,s}(40), \qquad \tau_c(\alpha) = \alpha b_c, \quad \alpha \in \{0.90, 0.95, 1.00\}, \tag{24}$$

where $\mathcal{B}_c^{40}$ contains all recorded baseline runs that complete 40 charged evaluations. All three thresholds are reported; none is selected using ADAPT's outcomes. The level b_c is an empirical baseline result, not a global optimum or a prescribed physical specification. First attainment is

$$N_{m,c,s}(\alpha) = \min\{n : H_{m,c,s}(n) \geq \tau_c(\alpha)\}. \tag{25}$$

If a run never reaches the threshold, it is marked as not reached (NR), without assigning an artificial evaluation count. Such runs remain in the attainment-rate denominator. Initialization, unsuccessful evaluations, exact-cache hits, and placement-only evaluations all consume their recorded budget slots; only qualified full-flow results contribute to HV. The analysis measures charged evaluations, not elapsed time or equal-cost full-flow executions.

Attainment across quality levels. Figure 7 summarizes seeds 11, 22, and 33 for ADAPT and the five baselines with three recorded seeds. ADAPT reaches each threshold in all 15 runs, requiring an average of 6.67, 8.40, and 12.47 charged evaluations at the 90%, 95%, and 100% levels, respectively. The means weight the five benchmarks equally. At the 100% level, all ADAPT runs attain the threshold within 29 evaluations. These first-attainment statistics quantify progress toward a common quality level, complementing the fixed-budget HV comparison.