SIZINGAGENT: INCENTIVIZING PHYSICS-GROUNDED REASONING IN LLM-BASED ANALOG CIRCUIT SIZING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Models (LLMs) are increasingly adapted for analog circuit sizing, yet existing methods often treat them as direct parameter generators and defer all validation to SPICE. We observe that this leads to many risky parameter updates that sound plausible in language but contradict the local physical response of the circuit. In this paper, we introduce **SizingAgent**, a training-free LLM framework that empowers LLM agents with physics-grounded reasoning for sizing optimization in analog design automation. Instead of relying on the LLM to directly generate numerical design parameters, SizingAgent structures optimization as a verifiable sizing loop consisting of three modules: (1) Physics Initializer (PI), which extracts small-signal equations (S-Equations) from the netlist to provide circuit-specific physical priors; (2) Physics-Guided Reasoner (PGR), which analyzes specification gaps and derives targeted sizing updates through a physics-guided reasoning chain; and (3) Sizing Verifier (SV), which checks the physical consistency of the proposed updates using sensitivity matrix. We further release, to the best of our knowledge, the largest open-source benchmark for LLM-based analog circuit sizing. Under a shared Gemini 2.5 Flash backbone, SizingAgent achieves 74.8% Pass@1, surpassing the strongest LLM baseline by 32.8 percentage points and dramatically outperforming classical optimizers, whose Pass@1 reaches only 16.6% for Genetic Algorithms and 26.2% for Bayesian Optimization. Replacing the backbone with GPT-5.5 further improves performance to 93.1% Pass@1. The framework also generalizes across LLM families, with commodity backbones achieving 69.6–76.1% Pass@1 and Qwen3-32B improving from 10.1% to 59.1% over zero-shot prompting. Code and benchmark are available: `https://github.com/artifact-repro/sizingagent`.

1 INTRODUCTION

Analog circuit sizing remains a central bottleneck in integrated circuit design. Given a fixed topology, designers must choose transistor widths, lengths, and bias currents to satisfy multiple competing specifications, such as gain, bandwidth, phase margin, and power, under nonlinear and topology-dependent device physics Razavi (2005). Even a modest 10-transistor block can involve more than 15 coupled design parameters Razavi (2005), and expert designers often spend days to weeks tuning a single block. This complexity has motivated a long line of automated sizing methods. Classical optimizers, including Genetic Algorithms (GA), Bayesian Optimization (BO), and differential evolution, typically treat the circuit as a black-box simulator query problem. Although broadly applicable, they provide limited physical interpretability and often require many SPICE evaluations. Recent LLM-based sizers Liu & Chitnis (2025); Yin et al. (2024); Kochar et al. (2025); Yu et al. (2026); Poddar et al. (2025); Ahmadzadeh et al. (2025) offer a new direction by using large language models to propose sizing decisions. However, existing approaches largely use LLMs as direct parameter generators and defer validation to SPICE, leaving the proposed updates unconstrained by circuit-specific physical equations. As a result, LLMs may produce updates that are linguistically plausible but physically inconsistent with the local response of the circuit, a form of hallucination that is particularly harmful in numerical design optimization Ji et al. (2023).

We claim that LLMs should not be used as unconstrained numerical optimizers for analog sizing. Instead, they are better suited as physics-guided reasoning agents whose proposals are grounded in circuit equations and verified against measured simulator responses before being accepted. This

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.

perspective leads to a different sizing paradigm: rather than asking an LLM to output transistor dimensions, the sizing loop should expose the LLM to per-circuit small-signal equations, measured device behavior, and local sensitivity information. Such a loop can exploit the semantic and diagnostic strengths of LLMs while preventing physically invalid updates from reaching the simulator.

In this work, we introduce **SizingAgent**, a training-free LLM framework for physics-grounded analog circuit sizing. As shown in Fig. 1, existing methods often treat the LLM as a black-box optimizer that directly proposes sizing parameters relying on SPICE to repair physically inconsistent updates through costly trial and error. Instead, SizingAgent reformulates analog sizing as a *verifiable physics-grounded optimization loop* rather than direct LLM-based parameter generation, which consists of 3 modules. (1) **Physics Initializer (PI)** derives circuit-specific small-signal equations from the netlist and grounds transconductance targets using gm/Id lookup tables from measured SKY130 device sweeps. (2) **Physics-Guided Reasoner (PGR)** compares SPICE measurements against target specifications and produces a structured reasoning chain (i.e., observation, equation, constraint, diagnosis, and action) to derive targeted multi-spec sizing updates. (3) **Sizing Verifier (SV)** then validates each proposed update against a sensitivity matrix, rejecting actions whose direction conflicts with the measured local circuit response. Together with multi-start initialization and sensitivity search strategies, SizingAgent provides a training-free framework that converts LLM reasoning into physically constrained, simulator-verifiable sizing decisions. In summary, this paper makes the following contributions:

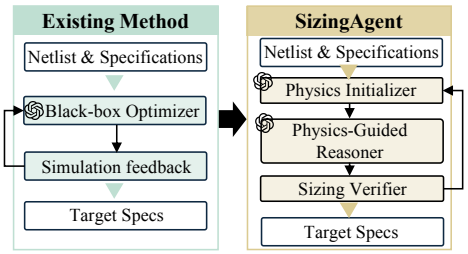


Figure 1: **Comparison of existing methods and SizingAgent.** Existing methods rely on black-box optimization to update design parameters from simulation feedback, whereas SizingAgent incentivizes physics-grounded reasoning through a verifiable sizing loop.

Conceptual. We present an early attempt to incentivize physics-grounded reasoning in LLM-based analog circuit sizing, shifting LLMs from black-box numerical optimizers toward verifiable reasoning agents grounded in circuit equations and local simulator responses.

Algorithmic. SizingAgent structures optimization around three modules: a Physics Initializer (PI) that extracts circuit-specific small-signal equations, a Physics-Guided Reasoner (PGR) that diagnoses specification violations and proposes targeted sizing-variable updates, and a sensitivity-based Sizing Verifier (SV) that rejects direction-inconsistent actions before simulation.

Benchmark. We release a circuit sizing benchmark built on SKY130 PDK for LLM-based analog sizing, covering 4–17 design parameters per circuit, four specification families, and three difficulty tiers. To the best of our knowledge, it is the largest open-source benchmark in this setting and supports end-to-end reproduction on a single workstation.

Empirical. On the proposed benchmark, we evaluate SizingAgent against strong baselines across analog circuit sizing tasks spanning multiple complexity levels. Results show consistent gains and successful adaptation to compact open-weight models (eg., Qwen3-4B).

2 Related Work

2.1 Analog Sizing Automation

Analog circuit sizing has been studied through three paradigms: **knowledge-based methods** Harjani et al. (1989); McConaghy et al. (2008) encode expert rules into synthesis flows; **optimization-based methods** Boyd et al. (2001); Lyu et al. (2018); Eriksson et al. (2019); Chen et al. (2022); Das & Suganthan (2010) cast sizing as black-box numerical search via geometric programming, Bayesian optimization, and evolutionary algorithms; and **learning-based methods** Wang et al. (2020); Settaluri et al. (2020); Budak et al. (2021; 2023); Li et al. (2021); Poddar et al. (2024) train neural surrogates or RL policies to guide the search. The gm/Id methodology Flandre et al. (1996) provides device-level grounding, and recent benchmarks Li et al. (2024); Mehradfar et al. (2024) standardize evaluation. Two limitations recur: knowledge-based methods require topology-specific hand-coding, and black-

box and learning-based optimizers reduce sizing to opaque numerical search with no causal account of why a configuration meets or fails a spec.

2.2 LLM-BASED APPROACHES FOR CIRCUIT SIZING

As LLMs increasingly enter chip design workflows Liu et al. (2023), three lines bring them into analog sizing: **prompt-driven iterative sizing** Liu & Chitnis (2025) uses the LLM as a direct parameter proposer; **LLM-augmented search** Yin et al. (2024); Kochar et al. (2025) couples LLM candidates with Bayesian-optimization surrogates; and **multi-agent or structured reasoning** Yu et al. (2026); Poddar et al. (2025); Ahmadzadeh et al. (2025); Zhang et al. (2025); Liu et al. (2025) imposes structure through specialized agents or hierarchical decomposition. In all three, the LLM acts as a parameter generator or search-space coordinator, structure sits at the workflow level rather than on the per-circuit physical equations, and verification is reduced to running each proposal through SPICE, so a wrong-direction update is caught only after one simulation has been spent on it, compounding with LLM hallucination Ji et al. (2023). SizingAgent addresses both limitations: the Physics Initializer replaces topology-specific manual coding by deriving per-circuit small-signal equations from the netlist, while the Physics-Guided Reasoner and Sizing Verifier ground each update by conditioning a typed five-step reasoning chain on an automatically computed sensitivity matrix and rejecting direction-inconsistent actions against that matrix *before* simulation. Table 1 contrasts these systems along five design axes.

Table 1: **Design-choice map of LLM sizers.** Columns: (a) physics-derived initial sizing from netlist equations and gm/Id; (b) Sizing Verifier checking each action against a sensitivity matrix before simulation; (c) no equations hand-coded for each topology; (d) simulation-derived feedback in the LLM reasoning prompt; (e) cross-LLM evaluation.

Method	(a) Physics init	(b) Sizing Verifier	(c) No hand-coded eqns	(d) Sim-grounded loop	(e) Cross-LLM tested
EEsizer Liu & Chitnis (2025)	✗	✗	✓	Partial	✗
ADO-LLM Yin et al. (2024)	✗	✗	✓	Partial	✗
LEDRO Kochar et al. (2025)	✗	✗	✓	✗	✗
AutoSizer Yu et al. (2026)	✗	✗	✓	✗	✗
HeaRT Poddar et al. (2025)	✗	✗	✗	✗	✗
AnaFlow Ahmadzadeh et al. (2025)	✗	✗	✗	Partial	✗
SizingAgent	✓	✓	✓	✓	✓

3 METHODOLOGY

3.1 PROBLEM FORMULATION

We formulate analog circuit sizing as a constrained search problem with simulation-based verification. A task instance is a tuple $\mathcal{I} = (\mathcal{N}, \mathcal{P}, \mathcal{T}, \mathcal{S})$, where $\mathcal{N}$ is a SPICE netlist with parameter placeholders, $\mathcal{P} = \{(p_i, [p_i^{\min}, p_i^{\max}])\}_{i=1}^{N}$ specifies N tunable variables (transistor widths, lengths, bias currents, compensation capacitors), $\mathcal{T} = \{(s_j, t_j, \bowtie_j)\}_{j=1}^{M}$ is the spec set with directions $\bowtie_j \in \{\geq, \leq\}$, and $\mathcal{S}$ is the SPICE simulator (ngspice). A candidate $\mathbf{p}$ is evaluated through $\mathcal{E}(\mathcal{N}, \mathbf{p}; \mathcal{S}) \to \mathbf{r}$, and the verification predicate $\texttt{PASS}(\mathbf{p})$ holds when every spec is met simultaneously:

$$\texttt{PASS}(\mathbf{p}) = 1 \iff \bigwedge_{j=1}^{M} \left(r_j \bowtie_j t_j\right). \tag{1}$$

The objective is to find $\mathbf{p}^*$ such that $\texttt{PASS}(\mathbf{p}^*) = 1$ while minimizing the simulation count N_{sim} and LLM call count N_{LLM}. Here, $\bowtie_j$ specifies the required direction of each constraint: $r_j \geq t_j$ for a lower bound and $r_j \leq t_j$ for an upper bound. A successful design must have valid measurements that satisfy every constraint. We report both simulation evaluations and LLM calls because they contribute to runtime, while API calls also incur monetary cost.

3.2 FRAMEWORK OVERVIEW

Figure 2 presents how SizingAgent uses circuit physics and simulation feedback across initialization and iterative refinement. PI constructs initial candidates using relations and operating targets inferred

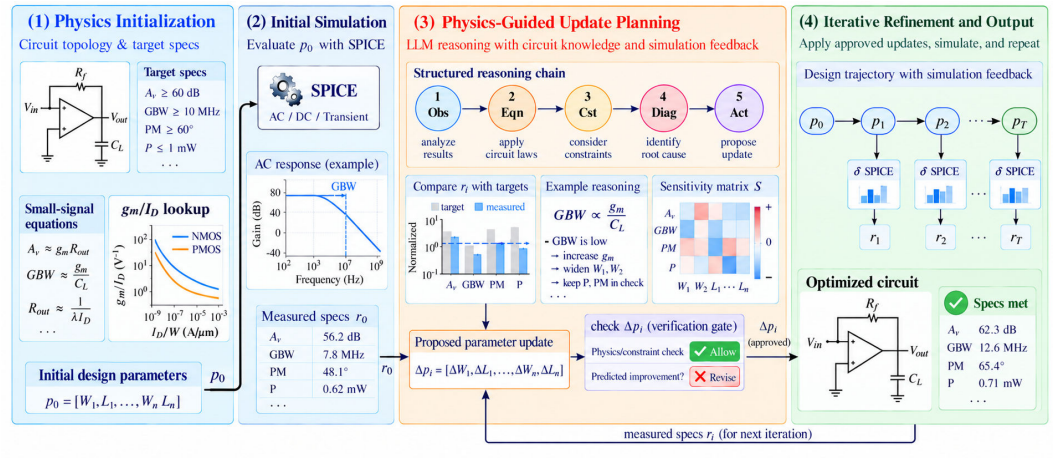


Figure 2: **SizingAgent framework.** PI derives circuit relations and uses gm/Id device data to construct an initial design, which is first evaluated by SPICE. PGR then reasons over the measured specification gaps and local sensitivity estimates to propose one or more parameter updates. SV screens each proposed action for directional consistency. Actions accepted within the same reasoning round are jointly applied to the current design and evaluated by a single SPICE simulation.

by the LLM, followed by gm/Id lookup. PGR proposes one or more parameter updates from the current measurements, specification gaps, and available sensitivity estimates. SV screens the proposed action directions individually. All actions accepted within the same reasoning round are then jointly applied to the current parameter vector, and the resulting candidate is evaluated by a single SPICE simulation. The search considers multiple initial candidates and uses further numerical exploration when refinement stalls. Algorithm 1 summarizes the procedure.

3.3 Physics Initializer (PI)

PI combines topology dependent sizing guidance with device characterization data. Given the netlist $\mathcal{N}$ and specifications $\mathcal{T}$, the LLM proposes relevant circuit relations and operating targets, such as transconductance and current allocation. Let e denote its equation text and $\mathbf{z}$ its numerical targets. The initialization interface is

$$(e, \mathbf{z}) = \mathcal{G}_{\text{eq}}(\mathcal{N}, \mathcal{T}; \text{LLM}), \qquad \mathbf{p}_{\text{init}} = \Phi_{\text{LUT}}(\mathbf{z}, \mathcal{D}_{\text{dev}}). \tag{2}$$

The gm/Id lookup maps these targets to device dimensions using device characterization data. Equation text supplies the design rationale.

PI evaluates up to three LLM generated candidates, subject to the budget and early success. If $\mathcal{C}_{\text{init}}$ indexes the evaluated candidates, the selected design is

$$k^* = \arg\max_{k \in \mathcal{C}_{\text{init}}} \text{Score}(\mathbf{r}^{(k)}), \qquad \mathbf{p}_0 = \mathbf{p}_{\text{init}}^{(k^*)}. \tag{3}$$

Score is the search criterion described in Sec. 4. The candidate count and early stopping rule are specified in the implementation details.

3.4 Physics-Guided Reasoner (PGR)

When sensitivity analysis is enabled, the matrix is estimated at a reference design $\bar{\mathbf{p}}$ and reused during subsequent refinement. For a positive parameter $\bar{p}_i$, the forward perturbation is $\bar{p}_i(1+\eta)$, with $\eta = 0.2$. Let $\bar{r}_j = f_j(\bar{\mathbf{p}})$ and let $r_j^{(i,+)}$ be the measurement after perturbing parameter i. The implemented estimate is

$$S_{ij} = \frac{r_j^{(i,+)} - \bar{r}_j}{\eta |\bar{r}_j|}. \tag{4}$$

Rows correspond to parameters and columns to specifications. The sign indicates the estimated response to increasing a parameter. This estimate is used when the parameter is positive and the

required measurements are valid with $|\bar{r}_j| \geq 10^{-12}$. The current implementation stores zero for unavailable or near zero measurement entries; such entries provide no reliable directional evidence.

The perturbation size is an empirical choice. It produces a measurable response, although it can cross an operating region boundary in circuits with narrow bias windows. The matrix therefore provides an approximate local guide whose reliability depends on the current design and the perturbation size.

Given the current parameters, measured specifications, available sensitivities, and targets, PGR emits

$$\text{chain}_t = \big(\text{obs}, \text{eq}, \text{constraint}, \text{diag}, \mathcal{U}_t\big), \qquad \mathcal{U}_t = \{a_{t,k}\}_{k=1}^{K_t}, \quad K_t \leq 2. \tag{5}$$

The five fields organize the proposal into observations, circuit relations, constraints, a diagnosis, and a set of candidate actions. Each action $a_{t,k}$ specifies one parameter, an increase or decrease direction, and a positive multiplicative factor. Thus, an individual action changes one parameter, while a reasoning round may propose up to two parameter changes that can subsequently be executed together.

PGR receives the current parameters, measured specifications, parameter bounds, available sensitivities, and a short history of search scores. These structured fields make the proposal easier to inspect and support selected numerical and directional checks. The structure follows reasoning prompts with explicit intermediate fields Wei et al. (2022); Perrier (2025). A single-action proposal, which is a special case of Eq. equation 5, may take the form:

$$\begin{aligned}
&\textsc{Obs} : \text{GBW} = 24\text{MHz vs target } 30\text{MHz (gap: } 20\%) \\
&\textsc{Eq} : \text{GBW} = g_{m1}/(2\pi C_L); \quad \textsc{Constraint} : \text{Power} = 509\mu\text{W} > 500\mu\text{W} \\
&\textsc{Diag} : \text{GBW limited, power-constrained}; \quad \textsc{Action} : W_1 \times 1.5.
\end{aligned}$$

Further exploration uses the stored sensitivities and random perturbations. For a violated specification s^* with a positive target, let $q = \text{sign}(t_{s^*} - r_{t,s^*})$ and $g = |t_{s^*} - r_{t,s^*}|/|t_{s^*}|$. The main sensitivity branch selects

$$\mathcal{A} = \{i : qS_{i,s^*} > 0\}, \qquad i^* = \arg\max_{i \in \mathcal{A}} |S_{i,s^*}|. \tag{6}$$

For a sufficiently sensitive entry, it computes a positive factor and applies a bounded update:

$$\begin{aligned}
\alpha &= \text{clip}\left(1 + \frac{g}{|S_{i^*,s^*}|}, 1.05, 3.0\right), \\
p'_{i^*} &= \text{clip}\big(p_{i^*}\alpha\xi, p_{i^*}^{\min}, p_{i^*}^{\max}\big).
\end{aligned} \tag{7}$$

The reference implementation uses $\xi = 10^{\epsilon}$ with $\epsilon \sim \mathcal{N}(0, 0.03^2)$ and selects this branch with probability 0.7 when sensitivities are available. It otherwise uses random perturbations, including when the eligible set is empty or the selected magnitude is at most 0.01. These rules describe the exploration phase; PGR proposals use their own specified action directions.

3.5 Sizing Verifier (SV)

SV uses the stored sensitivity matrix to screen each proposed action direction against the response required for its target specification. For an action $a_{t,k}$ and its associated target specification s_k^*, entries with sufficient sensitivity magnitude are checked by

$$\texttt{Verify}(a_{t,k}, S) = \mathbb{1}\big[\delta_{t,k}\,\text{sign}(S_{p_{t,k},s_k^*})\,\text{sign}(t_{s_k^*} - r_{t,s_k^*}) = 1\big], \tag{8}$$

where $p_{t,k}$ is the parameter modified by the action and $\delta_{t,k}$ is its proposed direction. The indicator is one when the proposed parameter direction agrees with the locally estimated response needed to reduce the specification gap. Entries below the sensitivity threshold provide insufficient directional evidence.

SV evaluates the constituent actions independently. Let

$$\widetilde{\mathcal{U}}_t = \{a_{t,k} \in \mathcal{U}_t : \texttt{Verify}(a_{t,k}, S) = 1\} \tag{9}$$

denote the actions retained after screening. If $\widetilde{\mathcal{U}}_t$ is nonempty, all retained actions are applied to the same current parameter vector before simulation:

$$\mathbf{p}_{t+1}^{\text{cand}} = \text{Apply}\big(\mathbf{p}_t, \widetilde{\mathcal{U}}_t\big), \qquad \mathbf{r}_{t+1} = \mathcal{E}\big(\mathcal{N}, \mathbf{p}_{t+1}^{\text{cand}}; \mathcal{S}\big). \tag{10}$$

Although the individual parameter edits may be applied sequentially in the implementation, no intermediate SPICE evaluation is performed between them. The final combined parameter vector is evaluated once. Thus, verification is performed per action, whereas execution and simulation occur at the level of the joint proposal.

This screening does not attempt to predict the full nonlinear interaction among simultaneously modified parameters. In particular, individually direction-consistent actions may interact after being applied together. SPICE therefore remains the final evaluator of the joint candidate and supplies the measurements used for subsequent search and success assessment.

The sensitivity matrix is reused to limit simulation cost. As the operating point changes, its estimated direction may become inaccurate. A stale entry can screen out a useful proposal or allow an unhelpful one. We therefore treat agreement with the matrix as a local directional-consistency check rather than a guarantee of the joint update's physical correctness. The implementation maintains the current search state and the best recorded candidate separately.

4 EXPERIMENTS

4.1 BENCHMARK

We construct and release a **circuit sizing benchmark** containing 11 Easy, 12 Medium, and 4 Hard circuits with 4 to 17 tunable parameters per circuit. The specifications span four families: small-signal AC (gain, gain-bandwidth product, bandwidth, phase margin), large-signal DC (power, output swing, mirror error), output-stage robustness (output resistance, PSRR, dropout, regulation accuracy), and frequency (ring-oscillator output). Compared with prior suites, AnalogGym Li et al. (2024) and AICircuit Mehradfar et al. (2024) target generic optimization and supervised-ML evaluation rather than controlled comparison across LLM-driven sizers. A 256-point Sobol feasibility check confirms that unguided quasi-random sampling rarely reaches the joint spec-feasible region (median 0%, mean 3.3% across 27 circuits), so the benchmark is not solvable by black-box random search alone. The full circuit listing, design principles, parameter-range protocol, per-circuit Sobol statistics, and the unified ngspice wrapper implementation appear in Appendix F.

4.2 BASELINES

We compare against 6 baselines spanning two families. **Traditional optimization**: **GA** (Gaussian mutation) and **BO** (Gaussian-process surrogate with EI acquisition) treat sizing as a black-box numerical search. **LLM-based solutions**: **EEsizer** Liu & Chitnis (2025) (pure LLM iteration), **ADO-LLM** Yin et al. (2024) (LLM-augmented GP-BO), **LEDRO** Kochar et al. (2025) (LLM range narrowing with TuRBO), and **AutoSizer** Yu et al. (2026) (multi-agent orchestration) cover the four major design choices in current LLM sizers (single-agent iteration, surrogate-coupled, search-space narrowing, multi-agent dispatch). To remove confounds from PDK / simulator / budget differences across prior papers, all LLM methods share the same Gemini 2.5 Flash backbone, SKY130 PDK, ngspice wrapper, and 50-simulation budget; per-method hyperparameters and re-implementation details are in Appendix G.

4.3 METRICS

A run *passes* when every spec is simultaneously satisfied; a circuit with 3/4 specs satisfied still counts as a failure for that run. We report **Pass@**k via the unbiased Chen et al. (2021) estimator $\text{Pass@}k = 1 - \binom{n-c}{k} / \binom{n}{k}$ at n=15 runs per circuit, with k=1 (first-attempt success) and k=5 (best-of-5). When a run does not pass, we report a continuous **Score** = $\exp\left(\frac{1}{M}\sum_j \ln r_j\right)$, the geometric mean of clipped per-spec ratios $r_j \in [0, 2]$, used internally by PGR to rank candidate updates. **Solved** (Table 3) counts circuits that pass in at least one run. SPICE simulation count N_{sim} and LLM API call count N_{LLM} per run serve as efficiency proxies.

4.4 MAIN COMPARISON

Table 2 reports per-circuit Pass@1 / Pass@5 percentages on the unified benchmark. Among the four LLM baselines on Gemini 2.5 Flash, EEsizer attains the strongest aggregate at 42.0/70.0

Table 2: **Main results: SizingAgent outperforms all LLM baselines across five LLM backbones.** Per-circuit Pass@1 / Pass@5 percentages on 27 circuits, 15 runs per cell, 50-sim budget. Pass@k uses the unbiased estimator of Chen et al. Chen et al. (2021). Baselines run on Gemini 2.5 Flash; SizingAgent columns show the same algorithm on five LLM backbones via OpenRouter.

	Traditional		**LLM baselines** (Gemini 2.5 Flash)				**SizingAgent**				
Circuit	GA	BO	ADO-LLM Yin et al. (2024)	LEDRO Kochar et al. (2025)	EEsizer Liu & Chitnis (2025)	AutoSizer Yu et al. (2026)	Gemini	GPT-4o-m	Claude-H	DeepSeek	GPT-5.5
Easy											
Current mirror	66.7/100	40/95.8	57.1/99.7	42.9/97.2	92.9/100	50/99	60/99.8	100/100	100/100	100/100	100/100
Common-source amp	0/0	6.7/33.3	7.1/35.7	0/0	28.6/87.4	0/0	100/100	100/100	100/100	100/100	100/100
Differential pair	0/0	0/0	0/0	0/0	0/0	0/0	100/100	100/100	60/99.8	100/100	46.7/98.1
Cascode mirror	26.7/84.6	20/73.6	100/100	64.3/100	100/100	21.4/76.9	100/100	100/100	93.3/100	100/100	100/100
Diode-load amp	100/100	100/100	78.6/100	92.9/100	42.9/97.2	92.9/100	80/100	100/100	100/100	93.3/100	100/100
Wilson mirror	100/100	100/100	100/100	100/100	100/100	100/100	100/100	100/100	100/100	100/100	100/100
Common-gate amp	0/0	40/95.8	7.1/35.7	0/0	7.1/35.7	7.1/35.7	100/100	0/0	93.3/100	66.7/100	100/100
PMOS mirror	0/0	13.3/57.1	92.9/100	14.3/60.4	100/100	21.4/76.9	100/100	100/100	100/100	100/100	100/100
CMOS inverter	0/0	20/73.6	7.1/35.7	7.1/35.7	21.4/76.9	0/0	46.7/98.1	0/0	0/0	33.3/91.6	100/100
Active-load CS	0/0	46.7/98.1	6.7/33.3	6.7/33.3	0/0	6.7/33.3	33.3/91.6	13.3/57.1	6.7/33.3	13.3/57.1	40/95.8
Source-degen CS	0/0	26.7/84.6	0/0	0/0	53.3/99.3	0/0	100/100	6.7/33.3	20/73.6	66.7/100	100/100
Medium											
Five-T OTA	6.7/33.3	6.7/33.3	35.7/93.7	7.1/35.7	7.1/35.7	28.6/87.4	93.3/100	100/100	100/100	100/100	100/100
Two-stage Miller OTA	0/0	13.3/57.1	21.4/76.9	14.3/60.4	42.9/97.2	0/0	66.7/100	100/100	93.3/100	100/100	100/100
Telescopic OTA	0/0	20/73.6	28.6/87.4	7.1/35.7	35.7/93.7	28.6/87.4	73.3/100	100/100	93.3/100	100/100	100/100
PMOS-input OTA	0/0	6.7/33.3	7.1/35.7	0/0	50/99	0/0	100/100	93.3/100	86.7/100	93.3/100	93.3/100
Source follower	0/0	0/0	0/0	0/0	64.3/100	0/0	100/100	100/100	100/100	100/100	100/100
3-stage ring osc	6.7/33.3	100/100	100/100	92.9/100	100/100	50/99	100/100	100/100	93.3/100	100/100	100/100
Cascode CS amp	0/0	0/0	14.3/60.4	0/0	14.3/60.4	0/0	80/100	0/0	20/73.6	20/73.6	100/100
Symmetric OTA	60/99.8	13.3/57.1	13.3/57.1	53.3/99.3	33.3/91.6	60/99.8	93.3/100	93.3/100	86.7/100	86.7/100	100/100
PMOS cascode amp	0/0	0/0	6.7/33.3	0/0	0/0	0/0	40/95.8	0/0	0/0	0/0	100/100
PMOS-input 2-stage	6.7/33.3	13.3/57.1	26.7/84.6	13.3/57.1	40/95.8	0/0	93.3/100	80/100	66.7/100	86.7/100	93.3/100
Regulated cascode CS	0/0	0/0	0/0	0/0	0/0	0/0	6.7/33.3	0/0	6.7/33.3	0/0	100/100
Folded cascode (PMOS in)	0/0	6.7/33.3	40/95.8	6.7/33.3	93.3/100	0/0	60/99.8	100/100	93.3/100	100/100	100/100
Hard											
Folded cascode OTA	0/0	0/0	14.3/60.4	7.1/35.7	14.3/60.4	0/0	93.3/100	100/100	93.3/100	100/100	100/100
LDO regulator	66.7/100	100/100	100/100	85.7/100	78.6/100	100/100	86.7/100	100/100	100/100	100/100	100/100
Three-stage OTA	6.7/33.3	13.3/57.1	7.1/35.7	7.1/35.7	14.3/60.4	7.1/35.7	6.7/33.3	93.3/100	80/100	86.7/100	93.3/100
Telescopic CS amp	0/0	0/0	0/0	0/0	0/0	0/0	6.7/33.3	0/0	0/0	6.7/33.3	46.7/98.1
Avg Pass@1 / Pass@5	16.6/26.6	26.2/52.4	32.3/57.8	23.1/41.5	42.0/70.0	21.3/38.2	74.8/92.0	69.6/73.7	69.9/82.0	76.1/87.2	**93.1/99.7**

Pass@1/Pass@5; ADO-LLM at 32.3/57.8 and LEDRO at 23.1/41.5 exhibit a large Pass@1-to-Pass@5 gap, indicating that they reach valid sizings only after several independent samples rather than on the first attempt; AutoSizer, with four specialized agents, lags at 21.3/38.2. None of the four exceeds 50% Pass@1 on more than 6 of the 12 Medium-tier circuits. Traditional GA and BO under the same protocol reach 16.6% and 26.2% Pass@1.

On the same Gemini 2.5 Flash backbone as the LLM baselines, SizingAgent reaches 74.8/92.0 Pass@1/Pass@5, 32.8pp above EEsizer's 42.0%, with Pass@1 above 67% on 18 of 27 circuits. Replacing the backbone with GPT-5.5 scales performance to 93.1/99.7 Pass@1/Pass@5, with Pass@1 above 93% on 25 of 27 circuits and unique solutions on three otherwise-unsolved Hard- and Medium-tier circuits, indicating that the framework continues to absorb stronger reasoning capacity. The gain also transfers across LLM families. Across the four commodity backbones, Gemini 2.5 Flash, GPT-4o-mini, Claude Haiku 4.5, and DeepSeek-Chat, SizingAgent lands within a 6.5-point spread at 69.6–76.1% Pass@1, narrower than the 20.7-point baseline spread of 21.3–42.0% on a single LLM, and every SizingAgent backbone outperforms every baseline by at least 27.6pp. These gains stem from the per-circuit sensitivity-matrix grounding rather than the strength of any single LLM, and Appendix B reports the corresponding simulation and LLM-call costs.

Open-weight backbones. We further evaluate five off-the-shelf open-weight LLMs (3B–32B; zero fine-tuning, zero in-context exemplars) under (i) zero-shot direct-parameter prompting and (ii) the full SizingAgent framework (Table 3).

Table 3: **Framework evaluation on open-weight backbones.** Zero-shot prompts for direct parameter values; SizingAgent runs the full pipeline. Mean Pass@1 / Pass@5 on 27 circuits; Solved counts circuits met at least once.

	Zero-shot			**SizingAgent**			
Model	P@1	P@5	Solved	P@1	P@5	Solved	Δ P@1
Qwen3-4B	16.9	30.6	10/27	47.5	52.6	15/27	2.8×
Gemma3-4B	28.4	28.6	8/27	52.2	60.1	18/27	1.8×
Llama3.2-3B	10.5	22.7	10/27	49.4	60.5	18/27	4.7×
Qwen3-14B	19.3	28.3	9/27	54.5	69.0	20/27	2.8×
Qwen3-32B	10.1	21.1	8/27	59.1	71.9	23/27	5.9×

Across the five backbones, the framework yields 1.8–5.9× Pass@1 improvement over zero-shot, with the largest gains on the weakest pure-LLM baselines (Llama3.2-3B 4.7×, Qwen3-32B 5.9×): the **Sizing Verifier** and physics-grounded reasoning compensate for weak instruction-following that hurts

Table 4: **Ablation study.** Pass@1 on the 27-circuit benchmark. The Full Pipeline is a separate experiment from the main comparison.

Condition	Pass@1	Δ
Full Pipeline	79.5	–
w/o SV	71.1	−8.4
w/o Sensitivity Search	60.2	−19.3
w/o Multi-Start	55.1	−24.4
w/o PGR	64.7	−14.8
w/o PI	39.8	−39.8

these models in zero-shot mode. Qwen3-32B is worst at zero-shot (10.1% Pass@1) and best inside SizingAgent (59.1% Pass@1, 23/27 Solved), exceeding the strongest closed-source LLM baseline (EEsizer with Gemini, 42.0% Pass@1) by 17.1pp using a fully open-weight backbone. Per-task per-model results are in Appendix B.1.

Per-difficulty pattern. The advantage concentrates on Medium circuits where structured reasoning matters most: SizingAgent solves 8 of the 12 Medium circuits at Pass@1 of 67% or above, while no baseline reaches that bar on more than 3. On Easy, GP-BO and brute-force search already find a feasible region; on Hard, all methods degrade because failures shift from wrong-direction updates (which the **Sizing Verifier** catches) to the simultaneous coordination of multiple cascode bias voltages (Sec. D).

4.5 Ablation Study

We ablate each of SizingAgent's three core modules and two stabilizing strategies in turn on the full circuit sizing benchmark at 15 runs per cell. Table 4 reports the pass-count for every ⟨condition, circuit⟩ cell, exposing which component is decisive on which subset of circuits; the bottom row aggregates over the full benchmark.

The largest single contributor is the **Physics Initializer**. To isolate the value of LLM-derived physics from the rest of the agent, we replace its LLM-derived small-signal equations and gm/Id LUT with random sampling inside parameter ranges while keeping every other component intact: Pass@1 drops 39.8pp from 79.5% to 39.8%, and avg sims/run rises to 36.3 because the agent burns most of its budget recovering from infeasible starts. This is the strictest no-LLM-initialization ablation we are aware of for an LLM-driven sizer, and it isolates the LLM contribution to the initialization step.

The remaining four components each contribute between 8 and 25 percentage points. **Multi-Start Initialization** adds 24.4pp by rescuing low-parameter circuits where a single PI call lands in an infeasible region. **Sensitivity Search** adds 19.3pp by closing the last gap on high-parameter circuits where PGR identifies the right direction but cannot reach the joint-feasible region within the budget. **PGR** adds 14.8pp through mid-difficulty rescue on circuits with non-monotonic sensitivities. **SV** adds 8.4pp; its smallest aggregate effect masks an indispensable role when PGR is close to satisfying all constraints, where wrong-direction updates would otherwise consume 2–4 simulations of recovery effort. A worked end-to-end example is in Sec. 4.6.

Statistical robustness. Across 15 runs per circuit, SizingAgent achieves 303/405 = 74.8% Pass@1, with 10 of 27 circuits passing all 15 runs and all 27 circuits met in at least one run. The Full Pipeline row in Table 4 reports a separate experiment and is not pooled with these main results. The lowest per-circuit success count is 1/15, as discussed in Sec. D.

4.6 Case Study: A Verified PGR Trace

Figure 3 traces an actual sizing trajectory for the **telescopic OTA**. From a random initialization at I_{bias}=55 μA and base-scale widths producing 35.86 dB gain, 5.89 MHz GBW, and 297 μW, the agent explores three qualitative refinements: shrinking cascode widths, reversing to a low-current path, and aggressive multi-parameter scaling. None closes the GBW gap, and the closest attempt lands at 12.16 MHz against the 15 MHz target. The decisive final step consults the sensitivity entry $\partial\text{GBW}/\partial I_{\text{bias}}$=+0.546 and computes $I_{\text{bias}}\times 1.41$ together with $W_1\times 1.20$, giving

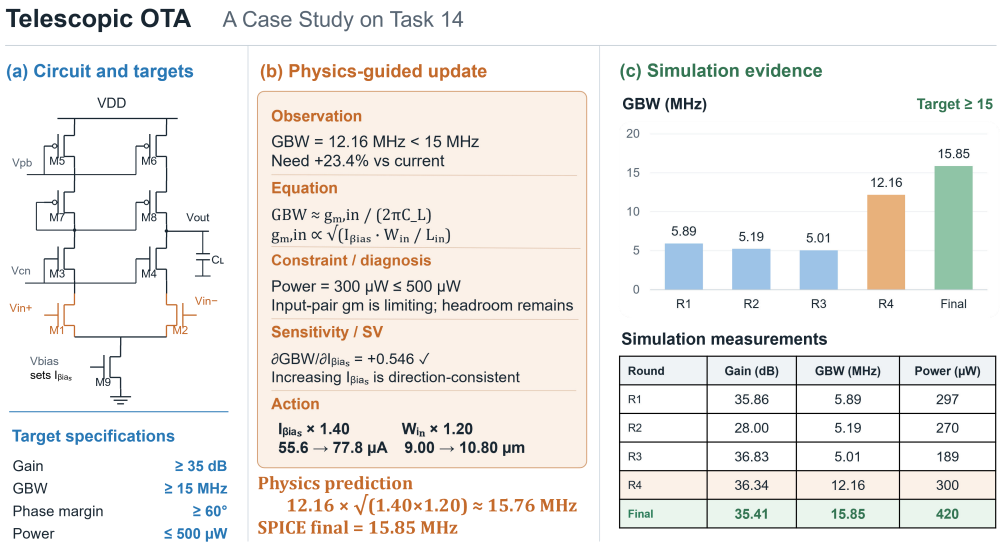


Figure 3: **SizingAgent optimization rounds on a telescopic OTA (Task 14).** Each block illustrates the updates proposed by the Sizing Agent and the resulting simulated specifications during each optimization round. SizingAgent initially explores several candidate updates that fail to meet the GBW target, then uses Physics-Grounded Reasoning to identify an effective increase in I_{bias} and W_1, raising GBW above 15 MHz while maintaining gain, phase margin, and power within specification.

$g_{m1} \times \sqrt{1.4 \cdot 1.2} \approx 1.30$ and meeting all specs in one simulation. The trajectory illustrates the role of physics grounding: heuristic actions identify direction but cannot compute exact scaling factors, while the sensitivity-driven update yields the precise factor needed to close the gap. The typed five-step reasoning chain that emitted the successful action is in Appendix B.2, Table 5; across our full trace collection of 14 SV-checked steps, the sensitivity-grounded actions passed at a 100% rate.

5 Limitations

Our evaluation is scoped to SKY130 with DC, AC, and simple transient analyses; transfer to advanced nodes (28 nm, 7 nm) and to time-domain blocks such as PLLs, comparators, and switched-capacitor filters requires additional measurement primitives. The sensitivity matrix costs N forward-difference simulations per circuit, which becomes a noticeable share of budget when N exceeds 15. Our open-weight study covers five off-the-shelf 3–32B LLM backbones under zero-shot prompting; fine-tuning, in-context exemplars, and larger open-weight backbones are left for future work. Remaining single-run failures concentrate on circuits requiring tight multi-bias coordination, analyzed in Sec. D.

6 Conclusion

SizingAgent addresses an *actionability gap* in LLM-based analog circuit sizing: LLMs can reason qualitatively about circuit behavior, yet translating this reasoning into reliable and executable sizing updates remains difficult. Rather than treating the LLM as a black-box numerical optimizer, SizingAgent grounds initialization and iterative decisions in circuit relations, device data, and simulator-derived local responses, turning LLM reasoning into physically grounded sizing actions that can be checked and evaluated by SPICE. On the proposed 27-circuit SKY130 benchmark, SizingAgent consistently outperforms LLM-based sizers and classical optimizers under a shared Gemini 2.5 Flash backbone, while also generalizing across LLM families and substantially improving open-weight models without fine-tuning. We release the benchmark, baseline implementations, and reference code under Apache 2.0. Future work includes higher-cardinality coordinated updates, interaction-aware verification, cross-PDK transfer, and extension to time-domain analog blocks.

AI USE STATEMENT

We acknowledge the use of large language models primarily for language polishing, rephrasing, improving readability, and refining the presentation of the manuscript, which helped us communicate the work more clearly. These tools were used to assist with stylistic refinement and presentation-level feedback, but did not generate experimental data or determine the reported numerical results. All ideas, experiments, analyses, and conclusions were developed and verified by the authors, who take full responsibility for the final content of this work.

REFERENCES

Mohsen Ahmadzadeh, Kaichang Chen, and Georges Gielen. Anaflow: Agentic llm-based workflow for reasoning-driven explainable and sample-efficient analog circuit sizing. In *2025 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, pp. 1–7. IEEE, 2025.

Stephen P Boyd, Thomas H Lee, et al. Optimal design of a cmos op-amp via geometric programming. *IEEE Transactions on Computer-aided design of integrated circuits and systems*, 20(1):1–21, 2001.

Ahmet F Budak, Prateek Bhansali, Bo Liu, Nan Sun, David Z Pan, and Chandramouli V Kashyap. Dnn-opt: An rl inspired optimization for analog circuit sizing using deep neural networks. *arXiv preprint arXiv:2110.00211*, 2021.

Ahmet F Budak, David Smart, Brian Swahn, and David Z Pan. Apostle: Asynchronously parallel optimization for sizing analog transistors using dnn learning. In *Proceedings of the 28th Asia and South Pacific Design Automation Conference*, pp. 70–75, 2023.

Chen Chen, Hongyi Wang, Xinyue Song, Feng Liang, Kaikai Wu, and Tao Tao. High-dimensional bayesian optimization for analog integrated circuit sizing based on dropout and g m/i d methodology. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 41(11):4808–4820, 2022.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Swagatam Das and Ponnuthurai Nagaratnam Suganthan. Differential evolution: A survey of the state-of-the-art. *IEEE transactions on evolutionary computation*, 15(1):4–31, 2010.

David Eriksson, Michael Pearce, Jacob Gardner, Ryan D Turner, and Matthias Poloczek. Scalable global optimization via local bayesian optimization. *Advances in neural information processing systems*, 32, 2019.

D Flandre, P Jespers, and F Silveira. A gm/id based methodology for the design of cmos analog circuits and its application to the synthesis of a silicon-on-insulator micropower ota. *IEEE J. Solid-State Circuits*, 31(9):1996, 1996.

Ramesh Harjani, Rob A Rutenbar, and L Richard Carley. Oasys: A framework for analog circuit synthesis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 8 (12):1247–1266, 1989.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38, 2023.

Dimple Vijay Kochar, Hanrui Wang, Anantha P Chandrakasan, and Xin Zhang. Ledro: Llm-enhanced design space reduction and optimization for analog circuits. In *2025 IEEE International Conference on LLM-Aided Design (ICLAD)*, pp. 141–148. IEEE, 2025.

Jintao Li, Haochang Zhi, Ruiyu Lyu, Wangzhen Li, Zhaori Bi, Keren Zhu, Yanhan Zeng, Weiwei Shan, Changhao Yan, Fan Yang, et al. Analoggym: An open and practical testing suite for analog circuit synthesis. In *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*, pp. 1–9, 2024.

Yaguang Li, Yishuang Lin, Meghna Madhusudan, Arvind Sharma, Sachin Sapatnekar, Ramesh Harjani, and Jiang Hu. A circuit attention network-based actor-critic learning approach to robust analog transistor sizing. In *2021 ACM/IEEE 3rd Workshop on Machine Learning for CAD (MLCAD)*, pp. 1–6. IEEE, 2021.

Chang Liu and Danial Chitnis. Eesizer: Llm-based ai agent for sizing of analog and mixed signal circuit. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2025.

Chengjie Liu, Weiyu Chen, Huiyao Xu, Yuan Du, Jun Yang, and Li Du. A large language model-based multi-agent framework for analog circuits' sizing relationships extraction. In *2025 International Symposium of Electronics Design Automation (ISEDA)*, pp. 181–187. IEEE, 2025.

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.

Wenlong Lyu, Fan Yang, Changhao Yan, Dian Zhou, and Xuan Zeng. Batch bayesian optimization via multi-objective acquisition ensemble for automated analog circuit design. In *International conference on machine learning*, pp. 3306–3314. PMLR, 2018.

Trent McConaghy, Pieter Palmers, Georges Gielen, and Michiel Steyaert. Automated extraction of expert knowledge in analog topology selection and sizing. In *2008 IEEE/ACM International Conference on Computer-Aided Design*, pp. 392–395. IEEE, 2008.

Asal Mehradfar, Xuzhe Zhao, Yue Niu, Sara Babakniya, Mahdi Alesheikh, Hamidreza Aghasi, and Salman Avestimehr. Aicircuit: A multi-level dataset and benchmark for ai-driven analog integrated circuit design. *arXiv preprint arXiv:2407.18272*, 2024.

Elija Perrier. Typed chain-of-thought: A curry-howard framework for verifying llm reasoning. *arXiv preprint arXiv:2510.01069*, 2025.

Souradip Poddar, Youngmin Oh, Yao Lai, Hanqing Zhu, Bosun Hwang, and David Z Pan. Insight: Universal neural simulator for analog circuits harnessing autoregressive transformers. *arXiv preprint arXiv:2407.07346*, 2024.

Souradip Poddar, Chia-Tung Ho, Ziming Wei, Weidong Cao, Haoxing Ren, and David Z Pan. Heart: A hierarchical circuit reasoning tree-based agentic framework for ams design optimization. *arXiv preprint arXiv:2511.19669*, 2025.

Behzad Razavi. *Design of analog CMOS integrated circuits*. , 2005.

Willy M Sansen. *Analog design essentials*, volume 859. Springer Science & Business Media, 2007.

Keertana Settaluri, Ameer Haj-Ali, Qijing Huang, Kourosh Hakhamaneshi, and Borivoje Nikolic. Autockt: Deep reinforcement learning of analog circuit designs. *arXiv preprint arXiv:2001.01808*, 2020.

Ilya M Sobol. Distribution of points in a cube and approximate evaluation of integrals. *USSR Computational mathematics and mathematical physics*, 7:86–112, 1967.

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.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Yuxuan Yin, Yu Wang, Boxun Xu, and Peng Li. Ado-llm: Analog design bayesian optimization with in-context learning of large language models. In *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*, pp. 1–9, 2024.

Xi Yu, Dmitrii Torbunov, Soumyajit Mandal, and Yihui Ren. Autosizer: Automatic sizing of analog and mixed-signal circuits via large language model (llm) agents. *arXiv preprint arXiv:2602.02849*, 2026.

Haoyi Zhang, Shizhao Sun, Yibo Lin, Runsheng Wang, and Jiang Bian. Analogxpert: Automating analog topology synthesis by incorporating circuit design expertise into large language models. In *2025 International Symposium of Electronics Design Automation (ISEDA)*, pp. 772–777. IEEE, 2025.

A Reproducibility

The full SizingAgent code, benchmark, and the four LLM-baseline re-implementations (EEsizer, ADO-LLM, LEDRO, AutoSizer) are released under Apache 2.0 at `https://github.com/artifact-repro/sizingagent` (also linked from the abstract). The artifact reproduces every reported Pass@k number end-to-end on a single workstation, using ngspice 41+ on the open-source SKY130 PDK and Python 3.10+ with `numpy`, `scipy`, and `scikit-optimize`. Closed-source LLM backbones (Gemini 2.5 Flash, GPT-4o-mini, Claude Haiku 4.5, DeepSeek-Chat, GPT-5.5) are routed through OpenRouter under a single unified API; open-weight backbones (Qwen3-4B/14B/32B, Gemma3-4B, Llama3.2-3B) run locally via Ollama.

Each ⟨method, circuit⟩ cell is repeated 15 times with seeds $\{1, \ldots, 15\}$ for stochastic LLM sampling, GA, and BO; Pass@k follows the unbiased Chen et al. Chen et al. (2021) estimator at n=15. Three top-level scripts reproduce the headline tables: `run_full_comparison.sh` (Table 2), `run_complete_ablation.sh` (Table 4), and `run_open_weight.sh` (Table 3). Checkpoint files in `results/` let a partial run resume without re-simulating completed cells. Per-method hyperparameters, prompt templates, and the unified ngspice wrapper appear in Appendices G, H, and F respectively.

B Algorithm and Compute Footprint

Compute footprint. The full headline benchmark, comprising SizingAgent on five LLM backbones (Gemini 2.5 Flash, GPT-4o-mini, Claude Haiku 4.5, DeepSeek-Chat, and GPT-5.5), four LLM baselines on Gemini 2.5 Flash, and the GA and BO traditional baselines, consumed 28,091 LLM API calls and 146,381 ngspice simulations in total, taking 130 wall-clock hours on a single workstation. Per-run on Gemini 2.5 Flash, SizingAgent averages $N_{\text{sim}} = 20.45$ simulations and $N_{\text{LLM}} = 8.11$ API calls; on GPT-5.5 these drop to 10.9 and 3.6 because the stronger backbone converges in fewer iterations. The full ablation in Sec. 4.5 adds another 59,168 ngspice simulations and 14,857 LLM calls (48 hours) on Gemini 2.5 Flash.

Resource efficiency. SizingAgent averages $N_{\text{LLM}} = 8.11$ calls and $N_{\text{sim}} = 20.45$ simulations per run. Under the combined cost $N_{\text{sim}} + 10N_{\text{LLM}}$, its cost is 101.55 per run, with 74.8% Pass@1.

B.1 Detailed Open-Weight Results

The summary in Table 3 (Sec. 4.4) aggregates over all 27 circuits. Per-circuit Pass@1 / Pass@5 for each of the open-weight backbones is in the released artifact under `small_model_results/`.

B.2 PGR Reasoning Trace (Telescopic OTA Case Study)

Table 5 shows the structured five-field PGR trace that produced the successful final joint proposal in Figure 3. At the time of emission, the design state was 36.34 dB gain, 12.16 MHz GBW (a 21% shortfall from the 15 MHz target), and 300 μW power on the telescopic OTA.

The observation and constraint fields can be checked directly against the simulator measurements and target specifications. The equation and diagnosis fields provide structured circuit reasoning but are not independently machine-verified. For the proposed actions, SV checks each parameter direction individually against the stored sensitivity matrix. For example, the local estimate for the bias current satisfies $\partial \text{GBW}/\partial I_{\text{bias}} = +0.546$, providing positive directional evidence for increasing I_{bias}.

Algorithm 1 SizingAgent

Require: Circuit netlist $\mathcal{C}$, specs $\{(s_j, t_j, \bowtie_j)\}$, budget B
$\mathbf{p}_1, \ldots, \mathbf{p}_k \leftarrow$ LLM-DERIVE-AND-SIZE($\mathcal{C}$, specs) *// PI with multi-start initialization*
for $i = 1$ to k **do**
  $\mathbf{r}_i \leftarrow \text{SIM}(\mathcal{C}, \mathbf{p}_i)$
end for
$i^* \leftarrow \arg\max_i \text{Score}(\mathbf{r}_i)$
$\mathbf{p}_{\text{cur}} \leftarrow \mathbf{p}_{i^*}$, $\mathbf{r}_{\text{cur}} \leftarrow \mathbf{r}_{i^*}$
$\mathbf{p}_{\text{best}} \leftarrow \mathbf{p}_{\text{cur}}$, $\mathbf{r}_{\text{best}} \leftarrow \mathbf{r}_{\text{cur}}$
if PASS($\mathbf{p}_{\text{best}}$) **then**
  return $\mathbf{p}_{\text{best}}$
end if
$\mathbf{S} \leftarrow$ SENSITIVITY($\mathcal{C}, \mathbf{p}_{\text{cur}}$) *// Sensitivity matrix*
for $t = 1$ to remaining budget **do**
  chain $\leftarrow$ LLM-REASON($\mathbf{p}_{\text{cur}}, \mathbf{r}_{\text{cur}}, \mathbf{S}$, specs) *// PGR*
  $\mathcal{U}_{\text{valid}} \leftarrow \emptyset$
  for action $a \in$ chain.actions **do**
    if VERIFY($a, \mathbf{S}, \mathbf{r}_{\text{cur}}$, specs) **then**
      $\mathcal{U}_{\text{valid}} \leftarrow \mathcal{U}_{\text{valid}} \cup \{a\}$ *// SV: per-action directional screening*
    end if
  end for
  if $\mathcal{U}_{\text{valid}} \neq \emptyset$ **then**
    $\mathbf{p}' \leftarrow$ APPLY-ACTIONS($\mathbf{p}_{\text{cur}}, \mathcal{U}_{\text{valid}}$) *// Jointly apply accepted actions*
    $\mathbf{r}' \leftarrow \text{SIM}(\mathcal{C}, \mathbf{p}')$ *// One SPICE evaluation for the joint proposal*
    $(\mathbf{p}_{\text{cur}}, \mathbf{r}_{\text{cur}}) \leftarrow$ UPDATE-SEARCH-STATE($\mathbf{p}_{\text{cur}}, \mathbf{r}_{\text{cur}}, \mathbf{p}', \mathbf{r}'$)
    if Score($\mathbf{r}'$) > Score($\mathbf{r}_{\text{best}}$) **then**
      $\mathbf{p}_{\text{best}} \leftarrow \mathbf{p}'$, $\mathbf{r}_{\text{best}} \leftarrow \mathbf{r}'$
    end if
    if PASS($\mathbf{p}'$) **then**
      return $\mathbf{p}'$
    end if
  end if
  if stalled for 3 iterations **then**
    $(\mathbf{p}_{\text{cur}}, \mathbf{r}_{\text{cur}}) \leftarrow$ SENSITIVITY-GUIDED-SEARCH($\mathcal{C}, \mathbf{p}_{\text{cur}}, \mathbf{S}$) *// Sensitivity Search*
    if Score($\mathbf{r}_{\text{cur}}$) > Score($\mathbf{r}_{\text{best}}$) **then**
      $\mathbf{p}_{\text{best}} \leftarrow \mathbf{p}_{\text{cur}}$, $\mathbf{r}_{\text{best}} \leftarrow \mathbf{r}_{\text{cur}}$
    end if
  end if
end for
return $\mathbf{p}_{\text{best}}$

The accepted actions, $I_{\text{bias}} \times 1.41$ and $W_1 \times 1.20$, were applied jointly to the same parameter vector and followed by a single SPICE evaluation. The resulting design achieved 15.85 MHz GBW while satisfying all specifications. SV screens the constituent actions individually; the final SPICE evaluation determines whether their joint nonlinear effect is successful.

C PER-CIRCUIT RESULTS

C.1 PER-CIRCUIT ABLATION READING

For Table 4 in Sec. 4.5, the per-condition rescued-circuit pattern is as follows. Each ablated component is decisive on a *distinct, mostly non-overlapping* subset of circuits, and several circuits are insensitive to any single-component removal (e.g. Wilson mirror, ring oscillator, LDO regulator are 15/15 in every condition; Telescopic CS amp is 0/15 in every condition).

Reading the columns:

Field	Content	Check / Role
OBSERVATION	GBW = 12.16 MHz vs target 15 MHz (21% shortfall)	Numerical check
EQUATION	$\mathrm{GBW} = g_{m1}/(2\pi C_L)$; $g_{m1} \propto \sqrt{I_{\text{bias}} \cdot W_1/L_1}$	Structured rationale
CONSTRAINT	$P = 300\,\mu\mathrm{W} \leq 500\,\mu\mathrm{W}$; the power budget permits increasing I_{bias} and W_1	Numerical check
DIAGNOSIS	Insufficient g_{m1} limits GBW; jointly increasing I_{bias} and W_1 is proposed to increase g_{m1}	Structured rationale
ACTION	$I_{\text{bias}} \times 1.41$ and $W_1 \times 1.20$; e.g., $\partial\mathrm{GBW}/\partial I_{\text{bias}} = +0.546$	SV direction check; joint SPICE validation

Table 5: **Structured PGR reasoning trace for the telescopic OTA.** Numerical observations and constraints are directly checkable, while equation and diagnosis fields record the PGR's structured reasoning. Action directions are screened individually by SV, and the jointly applied proposal is ultimately evaluated by SPICE.

Table 6: **Per-circuit ablation breakdown.** Pass-count (out of 15 runs) for every ⟨condition, circuit⟩ cell. Numbers in parentheses are the change versus the Full Pipeline column; *negative means the component matters there*. In each ablation column, the largest drop is shown in red and the largest gain in green. Bold marks Full Pipeline (the reference). 27 circuits × 6 conditions = 162 cells, each from 15 runs at 50-sim budget.

Tier	Circuit	Full	w/o SV	w/o Sens. Search	w/o Multi-Start	w/o PGR	w/o PI
Easy	Current mirror	**15**	15 (+0)	15 (+0)	1 (−14)	15 (+0)	5 (−10)
	Common-source amp	**13**	15 (+2)	14 (+1)	1 (−12)	15 (+2)	0 (−13)
	Differential pair	**12**	15 (+3)	15 (+3)	1 (−11)	15 (+3)	0 (−12)
	Cascode mirror	**15**	15 (+0)	12 (−3)	10 (−5)	15 (+0)	5 (−10)
	Diode-load amp	**11**	12 (+1)	8 (−3)	15 (+4)	14 (+3)	12 (+1)
	Wilson mirror	**15**	15 (+0)	15 (+0)	15 (+0)	15 (+0)	15 (+0)
	Common-gate amp	**15**	7 (−8)	3 (−12)	12 (−3)	11 (−4)	1 (−14)
	PMOS mirror	**15**	12 (−3)	15 (+0)	15 (+0)	3 (−12)	11 (−4)
	CMOS inverter	**5**	1 (−4)	0 (−5)	2 (−3)	0 (−5)	0 (−5)
	Active-load CS	**4**	7 (+3)	0 (−4)	2 (−2)	2 (−2)	2 (−2)
	Source-degen CS	**15**	13 (−2)	15 (+0)	14 (−1)	14 (−1)	0 (−15)
Medium	Five-T OTA	**14**	15 (+1)	13 (−1)	6 (−8)	13 (−1)	2 (−12)
	Two-stage Miller OTA	**15**	13 (−2)	11 (−4)	14 (−1)	14 (−1)	4 (−11)
	Telescopic OTA	**15**	13 (−2)	13 (−2)	6 (−9)	3 (−12)	2 (−13)
	PMOS-input OTA	**15**	15 (+0)	15 (+0)	9 (−6)	15 (+0)	12 (−3)
	Source follower	**15**	6 (−9)	8 (−7)	4 (−11)	5 (−10)	4 (−11)
	3-stage ring osc	**15**	15 (+0)	15 (+0)	15 (+0)	15 (+0)	8 (−7)
	Cascode CS amp	**12**	4 (−8)	2 (−10)	3 (−9)	4 (−8)	0 (−12)
	Symmetric OTA	**10**	7 (−3)	1 (−9)	8 (−2)	5 (−5)	8 (−2)
	PMOS cascode amp	**2**	1 (−1)	0 (−2)	1 (−1)	0 (−2)	4 (+2)
	PMOS-input 2-stage	**15**	15 (+0)	12 (−3)	14 (−1)	13 (−2)	12 (−3)
	Regulated cascode CS	**0**	1 (+1)	0 (+0)	3 (+3)	1 (+1)	5 (+5)
	Folded cascode (PMOS in)	**15**	14 (−1)	13 (−2)	12 (−3)	12 (−3)	15 (+0)
Hard	Folded cascode OTA	**15**	15 (+0)	14 (−1)	13 (−2)	15 (+0)	14 (−1)
	LDO regulator	**15**	15 (+0)	15 (+0)	15 (+0)	15 (+0)	15 (+0)
	Three-stage OTA	**14**	12 (−2)	0 (−14)	12 (−2)	13 (−1)	15 (+1)
	Telescopic CS amp	**0**	0 (+0)	0 (+0)	0 (+0)	0 (+0)	0 (+0)
Total passes / 405		**322 (79.5%)**	288 (−34)	244 (−78)	223 (−99)	262 (−60)	161 (−161)

- **w/o SV (−8.4pp aggregate).** Largest single-circuit drops are source_follower ↓ 9, common_gate_amp ↓ 8, and cascode_cs_amp ↓ 8. Smaller drops are concentrated on circuits where PGR is close to satisfying all constraints; here the verifier's role is to suppress wrong-direction updates that, when allowed through, consume 2–4 simulations of recovery effort. Several circuits actually *improve* slightly when SV is removed (common_source +2, differential_pair +3, five_T OTA +1): on already-easy initializations the extra reasoning

Table 7: **SizingAgent per-circuit results.** Per-circuit statistics for 15 runs on each of 27 circuits (405 trials). Ten circuits pass in all 15 runs. Score is the geometric mean of clipped per-spec ratios, and Avg Sims counts actual SPICE evaluations per run.

Difficulty	Circuit	Score (mean±std)	Avg Sims	Met Rate
Easy	Current mirror	1.58±0.12	24.1	9/15
	Common-source amp	1.14±0.05	3.1	15/15
	Differential pair	1.02±0.01	7.5	15/15
	Cascode mirror	1.53±0.15	1.9	15/15
	Diode-load amp	1.34±0.39	18.5	12/15
	Wilson mirror	1.75±0.00	1.1	15/15
	Common-gate amp	1.32±0.09	1.3	15/15
	PMOS mirror	1.62±0.01	1.2	15/15
	CMOS inverter	1.09±0.27	32.9	7/15
	Active-load CS	1.23±0.46	40.7	5/15
	Source-degen CS	1.28±0.16	8.1	15/15
Medium	Five-T OTA	1.19±0.05	13.0	14/15
	Two-stage Miller OTA	1.25±0.19	31.5	10/15
	Telescopic OTA	1.12±0.05	22.3	11/15
	PMOS-input OTA	1.28±0.06	1.3	15/15
	3-stage ring osc	1.50±0.16	4.1	15/15
	Source follower	1.04±0.07	7.9	15/15
	Cascode CS amp	1.26±0.04	13.2	12/15
	Symmetric OTA	1.30±0.04	25.6	14/15
	PMOS cascode amp	1.14±0.19	40.4	6/15
	PMOS-input 2-stage	1.36±0.13	23.6	14/15
	Regulated cascode CS	1.17±0.20	49.1	1/15
	Folded cascode (PMOS in)	0.93±0.66	25.0	9/15
Hard	Folded cascode OTA	1.24±0.06	19.9	14/15
	LDO regulator	1.50±0.45	6.3	13/15
	Three-stage OTA	0.78±0.38	49.5	1/15
	Telescopic CS amp	1.04±0.39	49.5	1/15
Total (mean)		**1.26**	**19.4**	**303/405**

iterations sometimes find a borderline action that just meets specs but would be rejected by the strict check. Net effect across 27 circuits is still -34 met-cells.

- **w/o Sensitivity Search (-19.3pp).** Largest drops are three_stage_ota $\downarrow 14$ (15/15 $\rightarrow$ 1/15; this circuit is solved primarily by Sensitivity Search), common_gate_amp $\downarrow 12$, cascode_cs_amp $\downarrow 10$, symmetric_ota $\downarrow 9$. The pattern: high-parameter or many-spec circuits where PGR identifies the right direction but cannot reach the joint-feasible region in the remaining budget; Sensitivity Search closes the last steps.
- **w/o Multi-Start Initialization (-24.4pp, largest drop in aggregate Pass@1).** Largest drops are current_mirror $\downarrow 14$, common_source_amp $\downarrow 12$, differential_pair $\downarrow 11$, source_follower $\downarrow 11$: all low-parameter circuits where a single PI call frequently lands in an infeasible region. Without the multi-start fallback, the pipeline also uses 29.7 sims/run (vs. 16.9 with), most of which is spent recovering from infeasible initializations.
- **w/o PGR (-14.8pp).** Largest drops are pmos_current_mirror $\downarrow 12$, telescopic_ota $\downarrow 12$, source_follower $\downarrow 10$, cascode_cs_amp $\downarrow 8$. Sensitivity Search alone handles many circuits, but mid-difficulty Medium circuits with non-monotonic sensitivities benefit specifically from the structured five-step reasoning.
- **w/o PI (-39.8pp, largest drop).** Replacing LLM-derived equations and gm/Id LUT with random sampling within parameter ranges drops the pipeline to 161/405. Largest single-circuit drops span almost the entire Easy and Medium tiers: source_degen_cs_amp $\downarrow 15$, common_gate_amp $\downarrow 14$, common_source_amp $\downarrow 13$, telescopic_ota $\downarrow 13$, pmos_ota $\downarrow 13$, five_T OTA $\downarrow 12$, two_stage_miller_ota $\downarrow 11$, source_follower $\downarrow 11$. Avg sims/run rises to 36.3 (the highest of any condition) because the agent burns most of its budget recovering from infeasible random starts; on a few intrinsically hard circuits where the LLM init is

Table 8: **GA and BO succeed only on circuits where random sampling works; both fail on every multi-transistor OTA.** Per-circuit 15-run results, 50-sim budget. GA solves 9 topologies (2 fully stable), BO 18 topologies (4 stable); contrast with SizingAgentś 27/27 and 10/27 respectively in Table 7. One GA run on `regulated_cascode_cs_amp` hit a non-converging ngspice transient and was excluded; GA total is therefore 67/404 rather than 67/405.

Circuit	GA Met	BO Met
Current mirror	10/15	6/15
Common-source amp	0/15	1/15
Differential pair	0/15	0/15
Cascode mirror	4/15	3/15
Diode-load amp	15/15	15/15
Wilson mirror	15/15	15/15
Common-gate amp	0/15	6/15
PMOS mirror	0/15	2/15
CMOS inverter	0/15	3/15
Active-load CS	0/15	7/15
Source-degen CS	0/15	4/15
Five-T OTA	1/15	1/15
Two-stage Miller OTA	0/15	2/15
Telescopic OTA	0/15	3/15
PMOS-input OTA	0/15	1/15
3-stage ring osc	1/15	15/15
Source follower	0/15	0/15
Cascode CS amp	0/15	0/15
Symmetric OTA	9/15	2/15
PMOS cascode amp	0/15	0/15
PMOS-input 2-stage	1/15	2/15
Regulated cascode CS	0/14	0/15
Folded cascode (PMOS in)	0/15	1/15
Folded cascode OTA	0/15	0/15
LDO regulator	10/15	15/15
Three-stage OTA	1/15	2/15
Telescopic CS amp	0/15	0/15
Total	**67/404**	**106/405**

itself confidently-wrong (regulated_cascode_cs_amp $+5$, pmos_cascode_amp $+2$), random init is occasionally luckier.

The five rescued-circuit sets overlap in only a few cases (cascode_cs_amp shows up under four of the five conditions; source_follower under four of five). This near-orthogonality provides stronger evidence that the five components are non-redundant than the aggregate -8 to -40pp deltas would on their own.

D Failure Mode Analysis

On the circuit sizing benchmark, SizingAgent has three particularly difficult 1/15 cases (`three_stage_ota`, `telescopic_cs_amp`, and `regulated_cascode_cs_amp`). We analyze reasoning traces from three representative cases (`inverter_amp`, `telescopic_cs_amp`, `differential_pair`) to identify root causes spanning measurement-primitive limits, multi-parameter coupling, and initialization variance.

inverter_amp (7/15 met; partial-success bottleneck). The CMOS inverter amplifier has a narrow useful bias region: the transfer-curve "trip voltage," where both transistors operate in saturation, lies near 0.7 V $\pm$ 0.05 V in SKY130 and depends strongly on W_p/W_n. To probe whether the LLM was wasting budget searching over V_{bias}, we ran an exploratory 5-run pilot on a *DC-sweep oracle* variant (`inverter_amp_v2`; not part of the main benchmark) that automatically sweeps V_{in} to locate the trip point and re-biases the circuit before AC analysis, thereby removing V_{bias} from the design space.

The oracle lifts the pilot Pass@1 from 0/5 to 1/5, indicating that part of the residual difficulty is measurement-side.

The remaining bottleneck is algorithmic. With V_{bias} fixed by the oracle, the search must still coordinate (W_p, W_n, L_p, L_n), and successful designs typically require the channel lengths to approach the minimum value (0.15 μm). PGR can jointly apply up to two parameter updates in one reasoning round, but this action cardinality is insufficient to directly coordinate all four tightly coupled variables. In addition, SV screens the constituent actions independently using single-parameter local sensitivities and therefore does not explicitly model their joint nonlinear interaction. Thus, additional measurement primitives alone are unlikely to close the remaining gap.

telescopic_cs_amp (1/15 met; score 1.04±0.39). This circuit contains four stacked transistors whose operating regions depend on three coupled cascode bias voltages (V_{cn}, V_{cp}, V_{bp}). The valid bias region is narrow, and isolated changes to one bias can move multiple stacked devices away from saturation. Although PGR can execute two accepted parameter updates jointly, the current action set cannot directly coordinate all three bias voltages within a single reasoning round. Moreover, SV evaluates the proposed directions independently against single-parameter sensitivity estimates, without explicitly modeling interaction terms among jointly modified biases. This combination of limited joint-action cardinality and interaction-agnostic verification makes narrow bias-window circuits particularly difficult.

differential_pair (15/15 met). All 15 runs in the main experiment satisfy the circuit specifications.

These failures point to two extensions: (i) specialized measurement primitives for bias-point-critical circuits, and (ii) richer joint-action representations that support higher-cardinality coordinated updates, together with interaction-aware verification for strongly coupled parameters and narrow bias windows.

E Budget Sensitivity

A potential confound is whether SizingAgent's advantage stems from algorithmic gain rather than from spending more simulation budget. To isolate the two, we sweep $B \in \{10, 20, 50, 100\}$ simulations per circuit on a 10-circuit subset (mixed Easy/Medium/Hard, excluding the three hardest where all methods struggle), running SizingAgent, GA, and BO at each budget. Figure 4 reports the result:

Figure 4: **SizingAgent at B=10 beats GA and BO at B=100.** Pass@1 versus simulation budget on a 10-circuit subset. SizingAgent reaches 80% with only 10 simulations per circuit; GA and BO need 10× more budget to reach merely 20–40%. SizingAgent saturates at 100% by B=50 and does not improve further at B=100, showing that the remaining failures are not budget-limited.

The sweep shows three trends. The efficiency advantage is not a consequence of budget: at B=10 SizingAgent already outperforms GA and BO at B=100 by 2–8×. SizingAgent also saturates at 100% Pass@1 by B=50 on this subset and gains 0 additional successes on the full benchmark when extended to B=100, indicating that the remaining failures are structural in nature (Sec. D) rather than budget-limited. The non-monotonic dip from 80% at B=10 to 70% at B=20 is a sensitivity-computation artifact: at B=10, circuits with 7 or more parameters exhaust budget during sensitivity computation before PGR can fire and fall back to PI (which meets only for easy circuits); at B=20, sensitivity completes but PGR has only 1–2 iterations to close the gap. A future refinement would cap the sensitivity-analysis sim count under small budgets.

F Benchmark Construction Details

This appendix documents the four design principles, the per-circuit construction protocol, the parameter-range and spec-target choices, and the achievability-verification procedure that backs the circuit sizing benchmark.

Table 9: **Circuit sizing benchmark: 4–17 parameters per circuit and four spec families.** Difficulty is set by parameter count and number of simultaneously-active constraints; Hard circuits combine 11 or more parameters with gain–bandwidth–stability–power tradeoffs. All circuits share a SPICE template with `{{PARAM}}` placeholders and a 30-second ngspice timeout.

Difficulty	Circuit	Params	Key Specs
Easy	Current mirror	5	Error, Rout, Vout_min
	Common-source amp	4	Gain, BW, Power
	Differential pair	4	Gain, BW, Power
	Cascode mirror	5	Error, Rout, Vout_min
	Diode-load amp	5	Gain, BW, Power
	Wilson mirror	5	Error, Rout, Vout_min
	Common-gate amp	4	Gain, BW, Power
	PMOS mirror	5	Error, Rout, Vout_max
	CMOS inverter	5	Gain, BW, Power
	Active-load CS	6	Gain, BW, Power
	Source-degen CS	5	Gain, BW, Power
Medium	Five-T OTA	7	Gain, GBW, PM, Power
	Two-stage Miller OTA	13	Gain, GBW, PM, Power
	Telescopic OTA	11	Gain, GBW, PM, Power
	PMOS-input OTA	7	Gain, GBW, PM, Power
	Source follower	4	Gain, BW, Power
	3-stage ring osc	4	Frequency, Power
	Cascode CS amp	7	Gain, BW, Power
	Symmetric OTA	9	Gain, GBW, PM, Power
	PMOS cascode amp	7	Gain, BW, Power
	PMOS-input 2-stage	13	Gain, GBW, PM, Power
	Regulated cascode CS	11	Gain, BW, Power
	Folded cascode (PMOS in)	9	Gain, GBW, PM, Power
Hard	Folded cascode OTA	11	Gain, GBW, PM, Power
	LDO regulator	11	Accuracy, Dropout, PSRR, Power
	Three-stage OTA	17	Gain, GBW, PM, Power
	Telescopic CS amp	12	Gain, BW, Power

F.1 Circuit Listing

F.2 Design Principles

The benchmark follows four principles: **(P1) Scale and difficulty diversity.** 27 circuits across three difficulty tiers (11 Easy, 12 Medium, 4 Hard), parameter counts ranging from 4 to 17, and four spec families (small-signal: gain / GBW / BW / phase margin; large-signal / system: error / Rout / dropout / PSRR / frequency). The Hard tier contains circuits no traditional optimizer or prior LLM baseline solves at 30% Pass@1 or above, ensuring the benchmark remains discriminating as methods improve.

(P2) Reproducibility from a single command. Every circuit is a ⟨`template.sp`, `config.json`⟩ pair: a SPICE netlist with `{{PARAM}}` placeholders and a JSON declaring parameter ranges, target specs, and the ngspice variables to extract. A unified Python wrapper (`benchmark/sim_wrapper.py`) materializes the netlist, calls `ngspice_con` with a fixed 30-second timeout, parses the output, and returns the same metric dictionary regardless of circuit. There is no per-circuit Python code: adding a new circuit requires only the two declarative files. The entire benchmark, including the four LLM-baseline re-implementations and the GA and BO traditional baselines, runs end-to-end on a single workstation.

(P3) Open foundry-grade PDK. We use SKY130 (130 nm, open-source) with the official BSIM4 device models, run through ngspice. Closed-source industrial PDKs, while higher-fidelity, prevent reproducibility and exclude academic groups; SKY130 has become the de facto open-source PDK

for analog research and our templates port directly to GF180MCU and IHP130 (also open) with only `.lib` substitution.

(P4) Non-trivial spec targets. Spec targets are set by hand based on textbook design targets for each canonical analog block (e.g. a typical Miller OTA target of 60–80 dB gain, 10–30 MHz GBW, 60° phase margin) and then refined through pilot simulations. As a feasibility check, a 256-point scrambled Sobol cloud per circuit shows that unguided quasi-random sampling rarely reaches the joint spec-feasible region (median 0%, mean 3.3% across 27 circuits); per-circuit statistics are in Sec. F.5.

F.3 Release

The full benchmark (27 circuits, four LLM-baseline re-implementations, GA and BO baselines, sim wrapper, and the SizingAgent reference implementation) is released under the Apache 2.0 license. We provide a Docker image with ngspice and SKY130 pre-installed; running `python run_quick_test.py` reproduces the headline number on Gemini 2.5 Flash in under one hour.

F.4 Per-Circuit Construction Protocol

For every circuit we executed the same four-step pipeline:

1. **Topology selection.** We chose textbook circuits drawn from Razavi Razavi (2005) and Sansen Sansen (2007) that span (a) the canonical small-signal building blocks every analog-design course covers (current mirror, common-source, differential pair, cascode, source follower); (b) multi-stage compositions that exhibit gain–bandwidth–stability tradeoffs (Miller-compensated two-stage OTAs, telescopic and folded-cascode OTAs, three-stage NMC OTAs); and (c) one system-level block exercising large-signal specs (LDO regulator). Topology coverage was prioritized over exhaustive enumeration of variants: a method that solves a 5-transistor OTA generalizes readily to its 7-transistor variant, but cascode coordination and three-stage compensation each represent qualitatively different reasoning challenges and are included separately.
2. **SPICE template.** Each topology was hand-encoded as a self-contained `template.sp` file with `{{PARAM}}` placeholders for tunable variables. Templates use the SKY130 `nfet_01v8` / `pfet_01v8` primary devices and the official BSIM4 model files. We deliberately avoid SKY130 transistor variants (`lvt`, `hvt`, RF flavors) to keep device-physics differences out of the sizing problem itself. Each template ends with an ngspice `.control` block that runs the necessary analyses (DC operating point, AC for small-signal, transient for the ring oscillator) and prints the spec values via `print`.
3. **Parameter range selection.** For width/length parameters we used the SKY130 design-rule extremes ($W \in [0.42\,\mu\text{m}, 200\,\mu\text{m}]$, $L \in [0.15\,\mu\text{m}, 10\,\mu\text{m}]$) clipped to physically meaningful sub-ranges per device role (e.g. tail-current devices use L of at least $0.5\,\mu$m to control λ effects). Bias voltage parameters use a $[0.4\,\text{V}, 1.5\,\text{V}]$ range matching SKY130 `1.8V` supply nominal operating points. Reference currents use $[1\,\mu\text{A}, 200\,\mu\text{A}]$ which covers analog-IC typical bias densities. The ranges intentionally include some non-functional regions (e.g. V_{bias} below threshold) so that sizing methods must reason about device operating regions rather than exploit narrow always-feasible windows.
4. **Spec target selection.** Spec targets are set by hand based on textbook design targets for the corresponding canonical analog block (e.g. a typical Miller OTA target of 60–80 dB gain, 10–30 MHz GBW, 60° phase margin), then refined through a small set of pilot simulations.

F.5 Achievability and Non-Triviality Stats

We verify each circuit's spec set via a Sobol-cloud feasibility check: for each circuit we generate a 256-point scrambled Sobol cloud Sobol (1967) over its parameter space (log-uniform over W/L, linear elsewhere; `scipy.stats.qmc.Sobol`, seed=42) and simulate every sample. Table 10 reports per-circuit met fractions: median 0% (16 of 27 circuits never satisfy all specs in the 256-sample cloud), mean 3.3%. We treat this as a sanity check rather than a proof of feasibility: a $0/256$ result implies the joint feasible region is small under unguided sampling but does not preclude a

solution existing. The reading is that the benchmark cannot be solved by black-box random search alone; a Pass@1 above the Sobol mean therefore reflects search and reasoning ability rather than benign spec-feasibility.

Table 10: **Sobol-cloud feasibility per circuit.** 256 quasi-random samples per circuit, simulated end-to-end through the same ngspice wrapper used for the main experiments. "Met" counts samples that satisfy all specs simultaneously. Total simulations: 6,912. Sobol clouds are released alongside the benchmark in `benchmark/circuits/*/sobol_validation.json`.

Difficulty	Circuit	Met / 256	%
Easy	wilson_current_mirror	97	37.9
	diode_load_amp	12	4.7
	common_gate_amp	1	0.4
	inverter_amp	1	0.4
	active_load_cs_amp	0	0.0
	cascode_current_mirror	0	0.0
	common_source_amp	0	0.0
	current_mirror	0	0.0
	differential_pair	0	0.0
	pmos_current_mirror	0	0.0
	source_degen_cs_amp	0	0.0
Medium	ring_oscillator_3stage	36	14.1
	source_follower	23	9.0
	symmetric_ota	2	0.8
	pmos_ota	1	0.4
	pmos_input_two_stage_ota	1	0.4
	folded_cascode_pmos_input	1	0.4
	cascode_cs_amp	0	0.0
	five_transistor_ota	0	0.0
	pmos_cascode_amp	0	0.0
	telescopic_ota	0	0.0
	two_stage_miller_ota	0	0.0
	regulated_cascode_cs_amp	0	0.0
Hard	ldo_regulator	51	19.9
	folded_cascode_ota	0	0.0
	telescopic_cs_amp	0	0.0
	three_stage_ota	0	0.0
Median / Mean		**0/256**	**0.0% / 3.3%**

F.6 Spec Family Conventions

We report three classes of metrics with consistent units across all circuits:

- **Small-signal AC** (gain, GBW, BW, phase margin): extracted from a `ac dec 200 1 10G` sweep with 1 V AC stimulus on the input. Gain is reported in dB. GBW and BW are reported in MHz (the frequency at which gain drops to 1 and $1/\sqrt{2}$ of low-frequency gain respectively). Phase margin is the phase at the gain-crossover frequency.
- **Large-signal DC** (output resistance Rout, output swing Vout_min/Vout_max, current-mirror error): extracted from a DC sweep of the output voltage / reference current.
- **System metrics** (LDO accuracy / dropout / PSRR; ring-oscillator frequency; total power): extracted from operating-point currents and dedicated sweeps (frequency from a transient analysis with FFT).

Power is uniformly reported in μW as $V_{\text{DD}} \cdot I_{\text{DD,total}}$ averaged over the analysis interval.

F.7 Difficulty Tier Assignment

The Easy / Medium / Hard partition is set by the rule of thumb: **Easy** = 6 or fewer parameters and 3 or fewer active specs; **Medium** = 7–10 parameters or includes a phase-margin spec (i.e. stability tradeoff is in scope); **Hard** = 11 or more parameters *and* simultaneous gain / GBW / phase-margin / power constraints, or contains coupled bias-window coordination (telescopic cascode). The 11-parameter LDO regulator is grouped Hard despite a structurally diagonal spec set because its 11 parameters span pass-transistor sizing, compensation network, and bias network, requiring multi-stage reasoning. The tier assignments are thus not a 1-D function of parameter count; they reflect the kind of reasoning the circuit demands.

F.8 What the Benchmark Does Not Cover (Yet)

The current release deliberately scopes to: (a) SKY130 130 nm only (cross-PDK transfer is future work); (b) DC, AC, and simple transient analyses (PLLs and switched-capacitor blocks require time-domain spec primitives we have not implemented); (c) closed-loop stability via phase-margin proxy only (no Bode-plot post-processing); (d) noise specs are not in scope (input-referred noise integration is a planned addition). The Limitations section discusses these scoping decisions in more depth.

G Baseline Reimplementation Details

The four LLM baselines we compare against (EEsizer, ADO-LLM, LEDRO, AutoSizer) report headline numbers under conditions that are mutually inconsistent and inconsistent with our setup:

- **PDK / device models.** EEsizer uses Predictive-Technology-Model (PTM) 90 / 130 / 180 nm rather than SKY130; ADO-LLM uses a commercial 90 nm process; LEDRO uses PTM-MG FinFET nodes (7–16 nm) with Cadence Spectre. Only AutoSizer reports on SKY130 with ngspice.
- **Simulator.** Spectre / HSPICE / ngspice differ in numerical kernel, default tolerances, and convergence heuristics, all of which affect achievability of any given spec set.
- **Budget.** EEsizer 20–25 LLM iterations (each itself triggering simulations); ADO-LLM 105 evaluations; LEDRO 1,200 evaluations; AutoSizer 300 samples; ours 50 simulations.
- **Spec set.** Per-paper headlines (e.g. AutoSizer's 100% Easy/Medium/Hard, ADO-LLM's FoM 3.52 with 0 missed specs on its 2-circuit benchmark) are computed against each paper's own spec set; there is no shared spec set across the prior papers.

A controlled comparison therefore requires fixing the substrate: the same circuits, simulator, budget, and metrics, with the same Gemini 2.5 Flash backbone. The 50-sim budget we hold constant is tighter than every prior setup, so the observed gap between SizingAgent and the LLM baselines is conservative. Giving the baselines more budget within the same substrate would not close the algorithmic difference without first matching the three core modules of SizingAgent (Physics Initializer, Physics-Guided Reasoner, Sizing Verifier) together with its two stabilizing strategies (multi-start initialization, sensitivity search).

As a sanity check on the re-implementations, EEsizer reaches 93–100% Pass@1 on the four simplest current-mirror circuits (Wilson, current_mirror, cascode mirror, PMOS mirror), the regime where its published version reports near-100% success; on harder Easy circuits (e.g. `differential_pair`), EEsizer drops to 0% as the spec window narrows. Across all 11 Easy circuits, EEsizer averages 50% Pass@1, ADO-LLM 42%, LEDRO 31%, AutoSizer 27%; the variance reflects each method's prompt design and search-strategy choices under our shared substrate. Code for every re-implementation is in the released artifact for inspection.

G.1 GA and BO Hyperparameters

Bayesian Optimization (BO). We use `scikit-optimize` 0.10 with a Gaussian-process surrogate, Matérn 5/2 kernel, length-scale per parameter dimension, and Expected Improvement acquisition

with ξ=0.01. The 50-simulation budget is split as 10 Sobol-sequence initial points followed by 40 acquisition-driven evaluations. Parameters are normalized to $[0, 1]$ before GP fitting, and the GP hyperparameters are refit by maximum marginal likelihood after each new observation. All runs use a fixed random seed per circuit and per repetition.

Genetic Algorithm (GA). We use a GA with population size 10, tournament selection (tournament size 3), arithmetic crossover at rate 0.8, and Gaussian mutation at rate 0.2 with per-parameter standard deviation set to 10% of the parameter range. Generations run until the 50-simulation budget is exhausted (5 generations of 10 individuals each). Elitism preserves the best individual across generations.

Both methods evaluate the same fitness function as SizingAgent (the geometric-mean Score in Sec. 4), and a run is counted as met as soon as a single individual satisfies all specs simultaneously, regardless of remaining budget. Implementation source is in `baselines/ga.py` and `baselines/bo.py` in the released artifact.

H LLM Prompt Templates

For reproducibility, we list the two LLM prompts used by SizingAgent verbatim. `{...}` denotes runtime substitutions; the rest is fixed text. All five backbones (Gemini 2.5 Flash, GPT-4o-mini, Claude Haiku 4.5, DeepSeek-Chat, GPT-5.5) receive identical prompts and decoding parameters, routed through a unified OpenRouter endpoint; no backbone-specific reasoning-effort or thinking-mode flags are enabled, so GPT-5.5 runs in the same default chat-completion mode as the other four.

H.1 Physics Initializer: LLM Equation Derivation & Transistor Sizing

System prompt:

```
You are an expert analog circuit designer.  Derive equations
and size transistors.  Output only JSON.
```

User prompt (issued once per circuit at T=0.4):

```
You are an expert analog circuit designer.  Analyze this
SPICE netlist on SKY130 130nm.
NETLIST:
{template[:2000]}
TARGET SPECS:
{specs_desc}
TUNABLE PARAMETERS (with allowed ranges):
{params_desc}
VDD = {vdd}V
TASK: Derive the circuit equations and compute initial
parameter values.
CRITICAL: Start from the POWER BUDGET. This is the most
common mistake.
Think step by step:
1.  Identify the circuit topology and count how many current
branches draw from VDD
2.  POWER FIRST: Find the power spec target.  Compute max
total current:  I_total = power_target / VDD.  Then divide
by number of branches to get Ibias.  Use 70% of max (safety
margin).
Example:  power<=500uW, VDD=1.8V, 2 branches → Ibias = 0.7 *
500e-6 / (2 * 1.8) = 97uA
3.  Write the key small-signal equation for EACH spec
4.  From target specs + the Ibias you computed, work
backwards to find gm targets
5.  Size transistors:  gm = 2*Id/Vov, choose Vov=0.15V, then
W/L = 2*Id/(mu_cox*Vov^2)
```

```
6. For minimize specs (power, Vout_min), ensure your values
are WELL BELOW target
SKY130 device data:
NMOS: mu_cox=222uA/V2, Vth=0.57V, lambda=0.083/V (at L=1um)
PFET: mu_cox=47uA/V2, Vth=0.93V, lambda=0.04/V (at L=1um)
gm = 2*Id/Vov, where Vov = Vgs - Vth (typically 0.1-0.2V for
moderate inversion)
W/L = 2*Id / (mu_cox * Vov^2)
Output ONLY JSON with keys: {topology, equations{spec_name:
eqn}, transistor_sizing[{param_W, param_L, device_type,
gm_target, Id_budget, L_choice, reasoning}]}.
```

The full Physics Initializer source is in `agent/llm_equation_derive.py`.

H.2 PGR Prompt (with Sensitivity Grounding)

System prompt:

```
You are an analog circuit expert. Output ONLY valid JSON,
no markdown fences, no text before or after.
```

User prompt (issued each PGR iteration at T=0.3):

```
You are sizing a {circuit_name} circuit (SKY130 130nm
process).
Current parameters: {json: param_name →
scientific-notation value}
Simulation results:
{json list of {name, actual, target, direction, met, gap_pct}
for each spec}
Most critical gap: {spec} (actual={x}, target={y},
gap={z}%)
Sensitivity analysis (from simulation, ∂spec/∂param
normalized):
{param1}: {spec_a: +0.55, spec_b: −0.31, ...}
{param2}: {spec_a: −0.12, ...}
...
Parameter ranges: {json: param_name → [min, max]}
History: {recent score trajectory}
Analyze the circuit and output a reasoning chain as JSON:
{
"reasoning_steps": [
{type: "observation", spec, actual, target, direction,
gap_pct},
{type: "equation", lhs, rhs, predicted_value, description},
{type: "constraint", constraint, lhs_value, rhs_value,
satisfied},
{type: "diagnosis", root_cause, mechanism,
limiting_device},
{type: "action", param, direction, factor, reason,
equation_basis}
],
"actions": [{param, direction, factor, reason}]
}
Rules:
- Output ONLY valid JSON, no markdown, no explanation
- Use the sensitivity data to choose parameter and direction
- For equations: include predicted_value so it can be
cross-validated
- factor should be between 1.05 and 3.0, calculated from the
gap and sensitivity
- Check constraints BEFORE suggesting current/power
```

```
 increases
- Focus on the most critical unmet spec
- Maximum 2 actions per response
```

The full PGR prompt builder is in `agent/sizing_agent.py::_build_reasoning_prompt`. The sensitivity matrix block (5–12 rows of $\partial s/\partial p$ values, filtered to absolute sensitivity of at least 0.05) is the new ingredient compared to prior LLM sizers.

H.3 Generation Hyperparameters

All experiments use temperature $T{=}0.3$ for PGR (lower than PI's $T{=}0.4$ to reduce variance during iterative refinement) and a 4096-token cap. Multi-start initialization produces up to 3 candidate sizings by re-sampling from the gm/Id table at different random seeds; the prompt sent to the LLM is identical across candidates, but each candidate draws a different L value from the gm/Id grid. No model fine-tuning or in-context exemplars are used; the prompts are zero-shot for every circuit.