INTO-OA: Interpretable Topology Optimization for Operational Amplifiers

Jinyi Shen[1], Fan Yang[1,*], Li Shang[2], Zhaori Bi[1], Changhao Yan[1], Dian Zhou[1] and Xuan Zeng[1,*]
[1] State Key Laboratory of Integrated Chips and Systems, School of Microelectronics, Fudan University, Shanghai, China
[2] State Key Laboratory of Integrated Chips and Systems, School of Computer Science, Fudan University, Shanghai, China

***Abstract*—This paper presents INTO-OA, an interpretable topology optimization method for operational amplifiers (op-amps). We propose a Bayesian optimization-based approach to effectively explore the high-dimensional, discrete topology design space of op-amps. Our method integrates a Gaussian process surrogate model with the Weisfeiler-Lehman graph kernel to extract structural features from a dedicated circuit graph representation. It also employs a candidate generation strategy that combines random sampling with mutation to balance global exploration and local exploitation. Additionally, INTO-OA enhances interpretability by assessing the impact of circuit structures on performance, providing designers with valuable insights into generated topologies and enabling the interpretable refinement of existing designs. Experimental results demonstrate that INTO-OA achieves higher success rates, a 1.84× to 19.10× improvement in op-amp performance, and a 3.20× to 14.33× increase in topology optimization efficiency compared to state-of-the-art methods.**

I. Introduction

The operational amplifier (op-amp) is a fundamental building block in analog circuits. Op-amp customization, including topology optimization and parameter sizing, is a must due to the diverse design specifications in various applications. Despite recent success in automated sizing [1]–[3], topology optimization remains a manual process, which is time-consuming, labor-intensive, and error-prone. The design quality is constrained by the variety of available op-amp templates and the expertise of analog designers, which takes years to develop. With the growing demand for high-performance analog circuits and the pressing time-to-market constraints, there is an urgent need for automated op-amp topology synthesis to generate customized topologies and expedite the design process.

Till today, a widely accepted solution for automated op-amp topology synthesis is yet to be established due to two fundamental challenges:

High-dimensional and discrete design space. The topology design space for multi-stage op-amps is high-dimensional and discrete, making it unsuitable to employ algorithms designed for smaller and continuous parameter sizing spaces. Topology selection methods employ either heuristic rules [4], [5] or data-driven machine learning models [6], [7] to choose topologies from a manually developed topology library. They suffer from a limited coverage of topologies and require excessive setup effort. The mainstream topology synthesis approaches focus on topology generation [8]–[13]. To reduce the design space dimensionality, constraints are imposed on either the number of stages [8]–[10] or the complexity of compensation schemes [11]–[13]. Attempts to address discreteness include evolutionary search [8]–[11] and reinforcement learning [12], [13], applied to topological representations in terms of mesh [8], hierarchical building blocks [9], [10], upper triangular matrix [11] and state vector [12], [13]. However, these methods limit the design space, hence the performance potential and the novelty of op-amp topologies. Recent studies using feature embedding [14] and variational graph autoencoder (VGAE) [15], [16] to generate topological representation also fail to deliver satisfactory design space exploration efficacy.

Interpretability. Considering the high cost and hence the risk of circuit fabrication, designers often seek minimal modifications to their trusted designs through a refinement process within their "interpretable zone" to ensure design reliability. Therefore, topology interpretability is a prerequisite for automated topology synthesis, to assist human reasoning of the synthesized topologies. However, existing methods focus on generating novel topologies from scratch and do not support topology refinement, nor provide necessary insights into the generated designs. As a result, lack of interpretability has been a major obstacle for designers to adopt an automated topology synthesis design flow.

Motivated by the aforementioned challenges, this paper presents INTO-OA, an interpretable, full-design space exploration method for op-amp topology optimization. INTO-OA aims to answer the following two research questions:

How to enable efficient exploration of the high-dimensional discrete topology design space? We develop a graph representation for op-amp topologies that reduces the high-dimensional design space into a lower-dimensional graph space. We propose a Weisfeiler-Lehman (WL) graph kernel-based Bayesian optimization method for op-amp topology optimization that utilizes a surrogate model for graphs and a novel candidate generation strategy for discrete space. The Gaussian Process (GP) surrogate model predicts the circuit performance based on the topological structures extracted using the WL graph kernel. The candidates are generated from random sampling and mutation of the current best topologies, effectively balancing global exploration and local exploitation to achieve efficient full-design space exploration.

How to enable interpretability of the topology optimization algorithm? As the GP model predicts circuit performance using interpretable graph features extracted by the WL kernel that correspond to specific subcircuit structures, the gradient of the GP inherently captures the influence of different subcircuit structures on circuit performance. This allows for the identification of performance-critical subcircuit structures in synthesized designs by analyzing the GP gradient, which enhances designers' understanding of novel topologies. Additionally, we propose an op-amp topology refinement method that adjusts

*Corresponding authors: {yangfan, xzeng}@fudan.edu.cn.

978-3-9826741-0-0/DATE25/© 2025 EDAA

existing designs based on the GP gradient, improving circuit performance while maintaining design reliability.

Experimental studies demonstrate that INTO-OA achieves state-of-the-art success rate, op-amp performance, and optimization efficiency. The identification of critical structures is validated through design knowledge and sensitivity analysis results. Furthermore, INTO-OA allows for the refinement of two existing topologies, resulting in performance enhancements while requiring only minimal modifications.

The remainder of this paper is organized as follows. Section II presents the problem formulation and introduces the preliminaries. INTO-OA is described in detail in section III. The experimental results are presented in Section IV. Section V concludes this work.

II. Preliminaries

A. Problem Formulation

We define topology synthesis as a constrained optimization problem. The goal is to find an op-amp topology (represented as graph G) in the topology design space $\mathcal{G}$ with the highest figure of merit (FoM) while satisfying all performance constraints,

$$\begin{aligned} \underset{G\in\mathcal{G}}{\text{maximize}} \quad & f(G, x^*) \\ \text{s.t.} \quad & x^* = \arg\max_{x\in S_G} \{f(G,x) : c_i(G,x) < 0\}, \end{aligned} \tag{1}$$

where f is the FoM, c_i is the i-th performance constraint, x is a vector consisting of tunable circuit parameters, and S_G is the parameter range for topology G. For each searched topology, an automated sizing method [1] based on Bayesian Optimization (BO) finds the best sizing x^* under performance constraints.

B. Bayesian Optimization

BO is an efficient algorithm for optimizing black-box functions in continuous space. BO employs an online surrogate model to approximate the behavior of the objective function and utilizes an acquisition function to balance global exploration and local exploitation. We use GP as the surrogate model because it can provide predictions with analytically calculated posterior mean and variance. The hyper-parameters of GP are determined through maximum likelihood estimation. The weighted expected improvement (wEI) [1] acquisition function is used for constrained topology optimization.

C. Behavior-level Topology Design Space

We adopt the behavior-level op-amp topology design space of [14]. The behavioral model for single-stage amplifier consists of a voltage-controlled current source (g_{mi}), accompanied by a parasitic resistor (R_{oi}) and a capacitor (C_{oi}). There are five circuit nodes in a three-stage op-amp: v_{in}, v_1, v_2, gnd, and v_{out}, as shown in Fig. 1. The three main amplifier stages g_{m1}-g_{m3} are fixed. For the 5 *variable subcircuits* between a pair of nodes, at most 25 types are available:

- A single R or C (2 types)
- R and C connected in parallel or serial (2 types)
- A g_m with different polarities and directions (4 types)
- A g_m with R or C connected in parallel or serial (16 types)
- No connection (1 type)

To ensure proper op-amp functionality, the types for each variable subcircuit are constrained by a set of op-amp topology

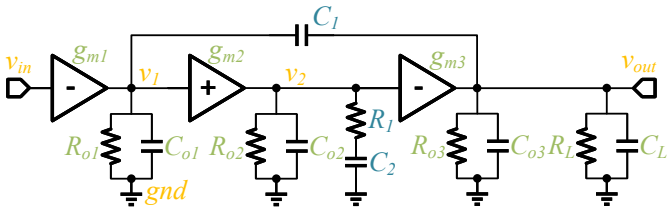


Fig. 1: An example behavior-level op-amp topology. The name of circuit nodes, fixed subcircuits, and *variable subcircuits* are colored in yellow, green, and blue, respectively.

design space rules $\mathcal{R}$ [14]. Specifically, the variable subcircuits between v_{in} and v_2 and between v_{in} and v_{out} can take 7 types, the variable subcircuit between v_1 and v_{out} can take 25 types, while the variable subcircuits between v_1 and gnd and between v_2 and gnd can take 5 types. Consequently, the behavior-level topology design space for three-stage op-amps encompasses 30625 distinct topologies.

Compared to transistor-level approaches, behavioral-level topology synthesis reduces the design space by abstracting away the fine details of transistor-level implementation, which also lowers computational costs by avoiding complex transistor models. This allows for efficient exploration of a wider range of design possibilities within a limited computational budget.

Transistor-level implementations of these behavioral models can be generated using the g_m/i_d-based mapping method [16]. In this process, the transistor-level circuit structure is derived from the behavioral-level topology. Specifically, the amplifier stage connected to v_{in} is converted into a differential amplifier with a current mirror load, while the remaining stages are transformed into common-source amplifiers, as illustrated in Fig. 6. The transistor dimensions are then determined based on behavioral-level parameters using the g_m/i_d lookup tables.

III. Proposed Approach

The framework of INTO-OA, depicted in Fig. 2, is presented in this section. We first purpose a graph representation for op-amp topologies, which transforms the high-dimensional topology design space into a lower-dimensional graph space. Building on this, we propose a topology optimization approach that utilizes a WL kernel-based BO to enable effective exploration of the discrete design space. This approach features a WL kernel-based GP (WL-GP) surrogate model for graphs and a candidate generation strategy. A key feature of INTO-OA is its interpretability, achieved by analyzing the gradient of the WL-GP. This gradient analysis identifies performance-critical structures in the synthesized topologies and allows for topology refinement with minimal modifications to the existing designs.

A. Graph Representation for Op-amp Topology

The structure of an op-amp topology can be effectively represented as a graph. Fig. 3 depicts our graph-based representation of the op-amp shown in Fig. 1. In this representation, both subcircuits and circuit nodes are modeled as graph nodes, while their connections are represented by graph edges. Each graph node is assigned a label indicating the corresponding subcircuit type or node name.

Our graph representation preserves the inherent graph structure of the circuit topology while reducing the high-dimensional topology design space into a lower-dimensional graph space.

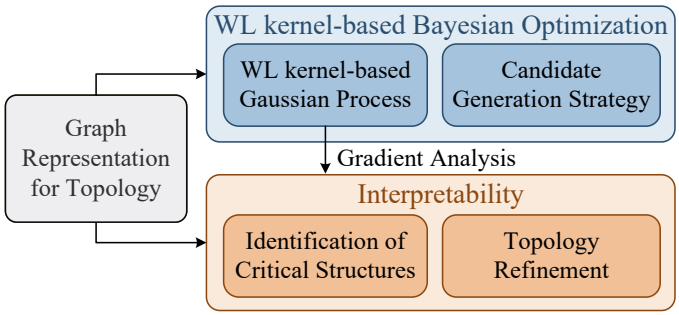


Fig. 2: The framework of INTO-OA.

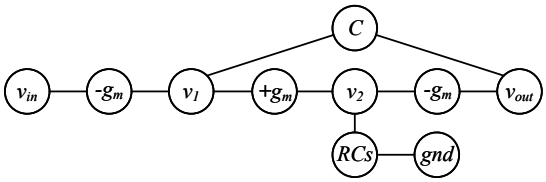
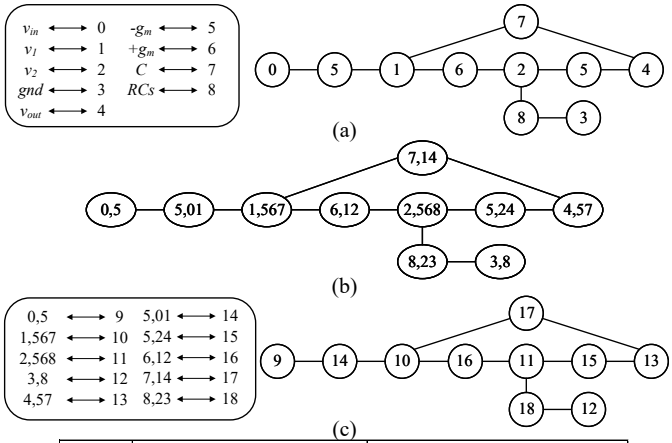


Fig. 3: Circuit graph of the behavior-level topology in Fig. 1. RCs denotes the series-connected R and C.



(d)

Extracted Features	h=0 Labels								h=1 Labels									
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$\varphi^{(1)}(G)$	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1

Fig. 4: Illustration of the feature extraction of the WL kernel for the circuit graph in Fig. 3 with the number of iterations set to h=1. (a) Labeling at h=0. (b) Neighborhood aggregation. (c) Labeling at h=1. (d) The extracted graph features.

This approach offers three key advantages over the graph representation in [16]. First, it employs undirected graphs and allows for the presence of loops in op-amp topologies, where loops are formed by feedforward and feedback modules. This prevents the information loss seen in the directed acyclic graphs used in [16]. Second, subcircuits are represented as graph nodes rather than as labels on edges, enabling the extraction of interpretable circuit structures via the WL kernel. Third, we eliminate redundant subcircuits of the "no connection" type, rather than assigning them a specific type as in [16], resulting in a more concise and accurate representation that closely aligns with the actual circuit connections.

B. WL kernel-based Surrogate Model

We develop a GP-based surrogate model that incorporates the WL graph kernel [17] to adapt to the graph space for topology optimization. This model estimates the performance of different topologies by capturing their structural characteristics. Fig. 4 illustrates the feature extraction process of the WL kernel for a circuit graph with the number of iterations set to h=1. In the first iteration (h=0), a unique label is assigned to each graph node based on its type, and the frequency of each label is counted to form the initial graph feature at h=0. Next, higher-order graph structures are collected through neighborhood aggregation, where each node's label is combined with the labels of its neighbors. During the second iteration (h=1), the concatenated labels from the neighborhood aggregation are compressed into new symbols using a hash function, and each graph node is relabeled with these new symbols. The graph feature at h=1 is then formed by combining the original graph feature at h=0 with the count of the newly assigned labels. This iterative feature extraction process enables the WL kernel to capture both fine-grained subcircuit structures and broader circuit-level features. As a result, the extracted features provide an expressive and comprehensive topological representation, enabling more accurate performance estimation and optimization in the design space.

Based on the extracted features $\varphi^{(h)}(G)$ and $\varphi^{(h)}(G')$ from two circuit graphs G and G' with h WL iterations, the WL kernel is computed as

$$k_{WL}^{(h)}(G, G') = \left\langle \varphi^{(h)}(G), \varphi^{(h)}(G') \right\rangle, \tag{2}$$

where $< \cdot >$ denotes the vector inner product. The WL graph kernel effectively measures the similarity between two circuit graphs by both comparing their local and global structures.

The WL kernel offers two advantages over the VGAE [16]. Firstly, the performance of the WL kernel remains stable because it does not rely on hyperparameters apart from h, which can be determined through maximum likelihood estimation in WL-GP. In contrast, VGAE's performance is highly sensitive to manually selected network structures and parameters, which require tuning in a separate training stage. The training stage demands substantial training data and extended training time. Secondly, the WL kernel facilitates direct optimization in the graph space by measuring graph similarity based on the extracted features. Conversely, VGAE forces the discrete topology design space into a continuous one, leading to performance discontinuity that hinders optimization.

The utilization of the WL kernel makes GP suitable for the graph space. Given $t-1$ circuit graph observations, our WL-GP model computes the posterior mean $\mu(G_t)$ and variance $\sigma^2(G_t)$ for a new graph G_t as

$$\mu(G_t) = \mathbf{k}(G_t, G_{1:t-1})\mathbf{K}_{t-1}^{-1}\mathbf{y}_{1:t-1}, \tag{3}$$

$$\sigma^2(G_t) = k(G_t, G_t) - \mathbf{k}(G_t, G_{1:t-1})\mathbf{K}_{t-1}^{-1}\mathbf{k}(G_{1:t-1}, G_t), \tag{4}$$

where $G_{1:t-1} = \{G_1, \ldots, G_{t-1}\}$ and $\mathbf{y}_{1:t-1} = [y_1, \ldots, y_{t-1}]^T$ represent the observed graphs and their corresponding circuit performance. $[\mathbf{K}_{t-1}]_{i,j} = k_{WL}^{(h)}(G_i, G_j)$. $\mathbf{k}(G_t, G_{1:t-1}) = \left[k_{WL}^{(h)}(G_1, G_t), \ldots, k_{WL}^{(h)}(G_{t-1}, G_t)\right] = \mathbf{k}(G_{1:t-1}, G_t)^T$.

The computational cost of WL-GP is comparable to that of conventional GP. According to [17], calculating the gram matrix using the WL kernel on N training graphs has a computational

complexity of $O\left(Nhm + N^2hn\right)$, where n and m represent the number of nodes and edges in the graphs, respectively. Given that $n \leq 13$, $m \leq 16$, and $h \leq 6$ suffice for our op-amp circuit graphs, the computational cost of the WL kernel is overshadowed by the $O\left(N^3\right)$ cost of GP training.

C. Interpretability

The interpretability of INTO-OA is derived from the interpretable topological features extracted by the WL kernel. These features, extracted at different iterations of the WL kernel, correspond to different levels of circuit structure. For instance, the extracted features at h=0 represent individual subcircuits, while the features at h=1 capture the connections of these subcircuits. Since the WL-GP predicts circuit performance based on these features, interpretability is achieved by analyzing the gradient of the WL-GP. The use of GP enables the analytical computation of derivatives [18]. The derivative of circuit performance y with respect to $\varphi_j(G_t)$, the j-th element in $\varphi(G_t)$, is a Gaussian distribution with an expected value of

$$\mathbb{E}_{p(y|G_t,\mathcal{D}_{t-1})}\left[\frac{\partial y}{\partial \varphi_j(G_t)}\right]=\frac{\partial \left\langle \varphi(G_t), \Phi_{1:t-1}\right\rangle}{\partial \varphi_j(G_t)}\mathbf{K}_{t-1}^{-1}\mathbf{y}_{1:t-1} \quad (5)$$

where $\Phi_{1:t-1} = [\varphi(G_1), \ldots, \varphi(G_{t-1})]^T$. As the value of $\varphi(G)$ represents the count of various circuit structures in topology G, their impact on circuit performance can be assessed by examining the derivatives of the WL-GP posterior mean with respect to $\varphi(G)$. The sign and magnitude of the derivatives at $\varphi(G)$ reflect the direction and sensitivity of topology G's performance about its circuit structures, respectively.

The gradient of the WL-GP allows identifying the most critical structures in an op-amp for each performance metric. This capability is valuable for novel topologies that circuit designers may not readily understand. By establishing connections between different performance metrics and specific circuit structures, designers can focus their analysis on particular circuit elements rather than the entire circuit. This approach enhances their comprehension of the functionality of each structure and promotes the exploration of new structures. Section IV-B presents the identification of critical structures.

Besides providing performance explanations, the surrogate gradient information can guide the refinement of existing topologies, enhancing specific performance with minimal modifications. Assuming better performance corresponds to a lower value, the refinement process begins by identifying the variable subcircuit with the largest gradient, as it contributes most negatively to performance. This subcircuit is then replaced with the most promising alternative, which has the smallest gradient. Next, the modified circuit part is resized, and the resulting design is simulated to evaluate its performance. If the specifications are not met, the subcircuit with the second-smallest gradient is considered. This process emulates the manual refinement conducted by circuit designers, but in a fully automatic manner guided by the gradient of the WL-GP. Due to the limited number of attempted topologies and sizing parameters per topology, the cost of refinement is significantly lower than that of conducting topology optimization from scratch. Furthermore, the reliability of the refined designs is preserved due to the minimal changes made to the original topology. Topology refinement results are presented in Section IV-C.

D. WL kernel-based Bayesian Optimization

We summarize our WL kernel-based BO method for topology optimization in Alg.1. It utilizes a WL-GP surrogate model and a novel candidate generation strategy. We construct individual WL-GP models for each performance metric and utilize the wEI [1] acquisition function α_{wEI} to handle constraints.

Since it is computationally infeasible to predict the performance of all graphs within the huge design space, we propose to generate candidates by combining mutation and random sampling. In mutation, candidates are generated by randomly modifying the variable subcircuit types within the current best topologies, adhering to the topology design space rules $\mathcal{R}$. We assign equal probabilities to mutate each variable subcircuit and set the expected number of mutated subcircuits to one. This approach ensures thorough coverage of the neighborhood of the current best topologies, facilitating effective local exploitation. On the other hand, random sampling distributes samples evenly across the entire design space, promoting global exploration. Additionally, we avoid redundant evaluations of the same topology by eliminating visited topologies from the design space. By striking a balance between exploration and exploitation, our candidate generation strategy enables efficient performance optimization across the entire design space with BO.

Algorithm 1: WL kernel-based Bayesian Optimization

input : Iterations T, performance number M
Get initial dataset $\mathcal{D}_0$ with random sampling from $\mathcal{G}$;
Train a WL-GP for each performance metric with $\mathcal{D}_0$;
$\mathcal{G}_{visited} \leftarrow \emptyset$, add the topologies in $\mathcal{D}_0$ to $\mathcal{G}_{visited}$;
for $t = 1$ **to** T **do**
  Generate candidates $\mathcal{P}_t$ from $\mathcal{G} \setminus \mathcal{G}_{visited}$;
  Select $G_t = \arg\max_{G \in \mathcal{P}_t} \alpha_{wEI}\left(G|\mathcal{D}_{t-1}\right)$;
  Evaluate G_t to obtain its performances $\{y_{t,i}\}_{i=1}^{M}$;
  Update dataset $\mathcal{D}_t \leftarrow \mathcal{D}_{t-1} \cup \left(G_t, \{y_{t,i}\}_{i=1}^{M}\right)$;
  Update the M WL-GP surrogate models with $\mathcal{D}_t$;
  $\mathcal{G}_{visited} \leftarrow \mathcal{G}_{visited} \cup \{G_t\}$;
return the current best topology G^*

IV. Experiment Results

A. Op-amp Topology Optimization Results

We evaluate INTO-OA in behavior-level topology optimization for three-stage op-amps. We compare INTO-OA with two state-of-the-art topology optimization methods: a genetic algorithm with feature embedding (FE-GA) [14] and Bayesian optimization with variational graph autoencoder (VGAE-BO) [16]. To ensure a fair comparison, we only compare with methods that utilize a behavior-level op-amp topology design space. In each BO iteration, INTO-OA generates half of the candidates through mutation and the other half through random sampling. For comparison, INTO-OA-r generates all candidates through random sampling, while INTO-OA-m generates all candidates through mutation. The candidate pool size for INTO-OA and number of iterations for optimizing the acquisition function for the baselines are both set to 200. All methods use 10 random initial topologies and undergoes 50 iterations. For performance evaluation, each op-amp topology undergoes sizing using BO with 10 initial points and 30 iterations. All

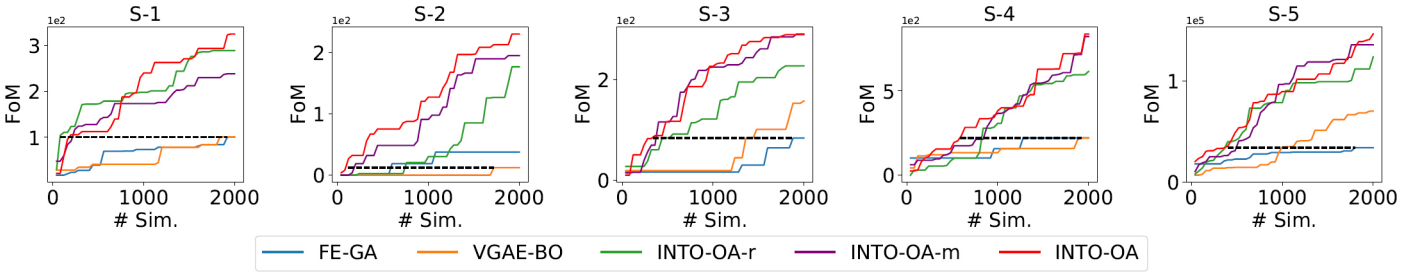


Fig. 5: Behavior-level op-amp optimization curves averaged from ten runs. The number of simulations for each method to achieve the same FoM (marked with the dashed lines) is given in Table II.

methods are implemented in Python. The experiments are conducted on a Linux workstation with two *Intel Xeon Gold 6226R* CPUs. Circuit simulations are carried out using *Hspice*.

TABLE I: The Design Specification Sets

Specs	Gain(dB)	GBW(MHz)	PM(°)	Power(μW)	C_L(pF)
S-1	>85	>0.5	>55	<750	10
S-2	>110	>0.5	>55	<750	10
S-3	>85	>5	>55	<750	10
S-4	>85	>0.5	>55	<150	10
S-5	>85	>0.5	>55	<750	10000

The following metrics are considered for op-amps: open-loop gain (Gain), gain-bandwidth product (GBW), phase margin (PM), power consumption (Power), and load capacitance (C_L). We conduct design tasks across diverse specifications (Specs), as detailed in Table I. Op-amp FoM is computed as

$$FoM = \frac{GBW\ [\text{MHz}] \cdot C_L\ [\text{pF}]}{Power\ [\text{mW}]}. \tag{6}$$

The supply voltage is set to 1.8V. We repeat the optimization for ten runs and evaluate the success rate (Suc. Rate, the proportion of runs that find op-amps meeting all constraints), average optimization results of the successful runs (Final FoM), the average number of simulations required to achieve the same FoM as indicated by the dashed lines in Fig. 5 (# Sim.), and the speedup in the number of simulations (Sim. Speedup).

The optimization results are summarized in Fig. 5 and Table II. INTO-OA significantly outperforms the baselines in success rate, optimization results and efficiency. Both INTO-OA-m and INTO-OA consistently identify feasible designs in all ten runs for all specifications. Compared to FE-GA and VGAE-BO, INTO-OA achieves a 1.84×-19.10× improvement in the average final FoM and also achieves the same FoM while requiring 3.20×-14.33× fewer simulations. INTO-OA outperforms both INTO-OA-r and INTO-OA-m, emphasizing the effectiveness of our candidate generation strategy, which balances exploration and exploitation to facilitate comprehensive design space exploration.

The performance of the best op-amps at the behavior-level are shown in Table III. The op-amps found by INTO-OA exhibit the highest FoM across all the specifications.

B. Identification of Critical Structures

The best op-amp identified by INTO-OA for S-4, shown in Fig. 6(a), is used to illustrate the use of the WL-GP gradient for identifying critical circuit structures. With the main amplifier stages fixed, we focus on two variable subcircuits: the

TABLE II: Behavior-level Op-amp Optimization Results

Specs	Method	Suc. Rate	Final FoM	# Sim.	Sim. Speedup
S-1	FE-GA	**10/10**	100.60	1920	1.00×
	VGAE-BO	7/10	100.98	1840	1.04×
	INTO-OA-r	**10/10**	288.69	**80**	**24.00**×
	INTO-OA-m	**10/10**	238.27	240	8.00×
	INTO-OA	**10/10**	**324.84**	200	9.60×
S-2	FE-GA	2/10	37.71	600	2.87×
	VGAE-BO	1/10	12.16	1720	1.00×
	INTO-OA-r	7/10	176.67	760	2.26×
	INTO-OA-m	**10/10**	194.87	200	8.60×
	INTO-OA	**10/10**	**230.20**	**120**	**14.33**×
S-3	FE-GA	4/10	83.91	1880	1.00×
	VGAE-BO	4/10	157.80	1480	1.27×
	INTO-OA-r	8/10	227.80	440	4.27×
	INTO-OA-m	**10/10**	**291.27**	400	4.70×
	INTO-OA	**10/10**	289.91	**360**	**5.22**×
S-4	FE-GA	3/10	220.87	1280	1.50×
	VGAE-BO	6/10	220.96	1920	1.00×
	INTO-OA-r	9/10	615.93	840	2.29×
	INTO-OA-m	**10/10**	823.56	880	2.18×
	INTO-OA	**10/10**	**838.58**	**600**	**3.20**×
S-5	FE-GA	**10/10**	33978	1800	1.00×
	VGAE-BO	**10/10**	70111	1000	1.70×
	INTO-OA-r	**10/10**	123644	**400**	**4.50**×
	INTO-OA-m	**10/10**	135578	520	3.46×
	INTO-OA	**10/10**	**146311**	**400**	**4.50**×

serially connected $-g_m$ and R between v_{in} and v_2 ($-g_m Rs$) and the serially connected R and C between v_1 and v_{out} (RCs), both of which affect the GBW and PM of the op-amp. Gradients are computed from the WL-GP models for GBW and PM, which were trained during optimization. For GBW, the gradients for $-g_m Rs$ and RCs are 0.04 and 0.36, respectively. For PM, the gradients for $-g_m Rs$ and RCs are -0.02 and -0.52, respectively. Both the sign and magnitude of the gradients are consistent with the sensitivity analysis results, where removing the $-g_m Rs$ increases GBW by 3.21 MHz but decreases PM by 3.53°, and removing the RCs increases GBW by 23.61 MHz but decreases PM by 112.45°. The gradients also align with design knowledge, as the RCs achieves frequency compensation through pole splitting and introduces a zero. Thus, the WL-GP gradient effectively indicates the impact of different structures on circuit performance, aiding designers in understanding the role of each subcircuit.

C. Op-amp Topology Refinement

A unique advantage of INTO-OA is that it allows the generation of reliable designs that meet target specifications through the topology refinement. The behavior-level topologies of two three-stage op-amps from top journals [19], [20] are selected as our initial topologies. These topologies, denoted as

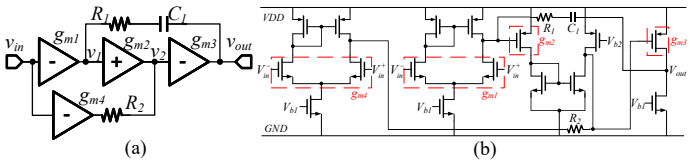


Fig. 6: (a) The behavior-level topology and (b) circuit structure of the best op-amp for S-3 found by INTO-OA. Parasitics, C_L, and the bias circuits are omitted for brevity.

TABLE III: Behavior-level Op-amp Performance

Specs	Method	Gain (dB)	GBW (MHz)	PM (°)	Power (μW)	FoM
S-1	FE-GA	92.31	12.69	67.15	419.52	302.57
	VGAE-BO	100.02	14.45	62.02	619.92	233.15
	INTO-OA	95.90	2.12	72.96	37.95	**558.63**
S-2	FE-GA	113.66	2.08	62.38	550.80	37.76
	VGAE-BO	111.47	0.79	69.21	646.62	12.17
	INTO-OA	111.35	5.87	76.08	175.17	**335.29**
S-3	FE-GA	101.66	5.29	66.39	388.14	136.38
	VGAE-BO	89.35	13.81	70.88	610.83	226.03
	INTO-OA	89.54	9.09	72.34	302.43	**300.46**
S-4	FE-GA	95.04	2.33	76.55	80.19	290.98
	VGAE-BO	96.92	1.23	60.99	28.77	428.69
	INTO-OA	94.11	5.00	67.69	38.55	**1297.02**
S-5	FE-GA	94.01	0.60	65.52	69.99	85727
	VGAE-BO	96.70	1.54	61.25	106.47	144642
	INTO-OA	99.85	2.61	64.63	61.69	**423083**

C1 and *C2*, are shown in Fig. 7. Their performances while driving the large capacitor of S-5 is shown in Table III, with the critical performances that do not meet S-5 underlined.

Guided by the WL-GP gradient for the critical performances, topology refinement enables performance enhancement in established topologies. For *C1*, the parallel-connected $-g_m$ and C between v_1 and v_{out} is replaced with $-g_m$. For *C2*, the $-g_m$ connected between v_{in} and v_2 is substituted by a series-connected $+g_m$ and C. The performances of the refined topologies, referred to as *R1* and *R2*, after solely sizing the modified circuit components, are presented in Table 7. The critical performance of each trusted topology improves after topology refinement, successfully satisfying S-5. The refinement succeeds for both circuits in the initial attempt, requiring only 40 simulations. Furthermore, reliability is maintained as the majority of the original designs' topology and component sizes remain unchanged.

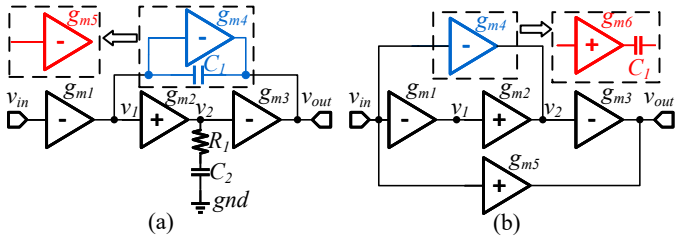


Fig. 7: Topology refinement of (a) *C1* and (b) *C2*. The worst-performing subcircuit structures (blue) are replaced with more promising alternatives (red), while the remaining subcircuits remain unaltered. Parasitics and C_L are omitted for brevity.

TABLE IV: Behavior-level Op-amp Performance before and after Topology Refinement

Circuit	Gain(dB)	GBW(MHz)	PM(°)	Power(μW)	FoM
C1	86.81	1.65	46.92	111.83	147545
R1	86.87	1.88	56.13	111.83	168112
C2	82.04	1.49	58.7	102.31	145636
R2	90.58	1.51	56.24	107.80	140074

TABLE V: Transistor-level Op-amp Performance

Specs	Method/ Circuit	Gain (dB)	GBW (MHz)	PM (°)	Power (μW)	FoM
S-1	FE-GA	88.09	8.83	70.14	425.89	207.33
	VGAE-BO	94.12	10.04	69.25	510.77	196.57
	INTO-OA	97.70	1.87	71.10	56.42	**331.44**
S-2	FE-GA	114.35	1.80	60.15	644.79	27.92
	VGAE-BO	116.27	0.66	68.99	661.27	9.98
	INTO-OA	111.26	5.93	70.80	232.80	**254.73**
S-3	FE-GA	95.75	5.34	68.92	477.84	111.75
	VGAE-BO	89.40	9.89	61.03	657.00	150.53
	INTO-OA	97.19	6.92	77.76	215.70	**320.82**
S-4	FE-GA	99.74	1.63	76.51	129.21	126.15
	VGAE-BO	91.72	1.01	69.39	45.46	222.17
	INTO-OA	94.61	4.82	62.67	43.57	**1106.27**
S-5	FE-GA	92.57	0.55	71.02	92.29	59595
	VGAE-BO	90.50	1.30	70.99	159.08	81720
	INTO-OA	97.60	2.38	70.05	78.23	**304231**
S-5	*R1*	89.79	1.30	56.40	100.17	129779
	R2	86.62	0.95	67.01	102.37	92801

D. Transistor-level Results

The op-amps are converted into the transistor level using the automated transistor mapping method [16] to validate their functionality. The performance of both the optimized op-amps and the refined op-amps are shown in Table V. Despite a decrease in FoM for most op-amps resulting from the inaccurate estimation of parasitics and power at the behavior level, all transistor-level op-amps meet the specifications and the op-amps found by INTO-OA maintain the highest FoM.

V. Conclusion

This paper introduces INTO-OA, an interpretable op-amp topology optimization method. Op-amp topologies are represented as graphs to preserve their inherent structures. By utilizing a WL kernel-based BO method, INTO-OA enables efficient full-design space exploration of the discrete topology design space. INTO-OA offers interpretability by measuring the impact of structures on op-amp performance through the surrogate gradients. This feature enables the identification of performance-critical subcircuit structures and facilitates targeted topology refinement. Experimental results confirm that INTO-OA achieves a higher success rate, a 1.84×-19.10× improvement in op-amp FoM and a 3.20×-14.33× improvement in efficiency compared to state-of-the-art methods. Our approach has the potential to be extended to other types of circuits, which will be investigated in our future research.

Acknowledgements

This research is supported partly by National Key R&D Program of China 2020YFA0711900, 2020YFA0711901, partly by the National Natural Science Foundation of China (NSFC) Research Projects under Grant 62474050, Grant 62304052, Grant 62141407, Grant 92473207, Grant 62474051, Grant 92373207 and Grant 62090025.

REFERENCES

[1] W. Lyu, P. Xue, F. Yang, C. Yan, Z. Hong, X. Zeng, and D. Zhou, "An efficient bayesian optimization approach for automated optimization of analog circuits," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 6, pp. 1954–1967, 2017.

[2] H. Wang, K. Wang, J. Yang, L. Shen, N. Sun, H.-S. Lee, and S. Han, "Gcn-rl circuit designer: Transferable transistor sizing with graph neural networks and reinforcement learning," in *2020 57th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2020, pp. 1–6.

[3] W. Cao, J. Gao, T. Ma, R. Ma, M. Benosman, and X. Zhang, "Rose-opt: Robust and efficient analog circuit parameter optimization with knowledge-infused reinforcement learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2024.

[4] R. Harjani, R. A. Rutenbar, and L. R. Carley, "Oasys: A framework for analog circuit synthesis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 8, no. 12, pp. 1247–1266, 1989.

[5] K. Swings, S. Donnay, and W. Sansen, "Hector: a hierarchical topology-construction program for analog circuits based on a declarative approach to circuit modeling," in *Proceedings of the IEEE 1991 Custom Integrated Circuits Conference*. IEEE, 1991, pp. 5–3.

[6] M. Fayazi, M. T. Taba, E. Afshari, and R. Dreslinski, "Angel: Fully-automated analog circuit generator using a neural network assisted semi-supervised learning approach," *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2023.

[7] S. Poddar, A. Budak, L. Zhao, C.-H. Hsu, S. Maji, K. Zhu, Y. Jia, and D. Z. Pan, "A data-driven analog circuit synthesizer with automatic topology selection and sizing," in *2024 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2024, pp. 1–6.

[8] A. Das and R. Vemuri, "Topology synthesis of analog circuits based on adaptively generated building blocks," in *Proceedings of the 45th annual Design Automation Conference*, 2008, pp. 44–49.

[9] T. McConaghy, P. Palmers, M. Steyaert, and G. G. Gielen, "Variation-aware structural synthesis of analog circuits via hierarchical building blocks and structural homotopy," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 9, pp. 1281–1294, 2009.

[10] ——, "Trustworthy genetic programming-based synthesis of analog circuit topologies using hierarchical domain-specific building blocks," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 4, pp. 557–570, 2011.

[11] Ž. Rojec, Á. Bűrmen, and I. Fajfar, "Analog circuit topology synthesis by means of evolutionary computation," *Engineering Applications of Artificial Intelligence*, vol. 80, pp. 48–65, 2019.

[12] Z. Zhao and L. Zhang, "Analog integrated circuit topology synthesis with deep reinforcement learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 12, pp. 5138–5151, 2022.

[13] Z. Zhao, J. Luo, J. Liu, and L. Zhang, "Signal-division-aware analog circuit topology synthesis aided by transfer learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 11, pp. 3481–3490, 2023.

[14] J. Lu, L. Lei, F. Yang, C. Yan, and X. Zeng, "Automated compensation scheme design for operational amplifier via bayesian optimization," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2021, pp. 517–522.

[15] J. Lu, L. Lei, F. Yang, L. Shang, and X. Zeng, "Topology optimization of operational amplifier in continuous space via graph embedding," in *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2022, pp. 142–147.

[16] J. Lu, L. Lei, J. Huang, F. Yang, L. Shang, and X. Zeng, "Automatic op-amp generation from specification to layout," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2023.

[17] N. Shervashidze, P. Schweitzer, E. J. Van Leeuwen, K. Mehlhorn, and K. M. Borgwardt, "Weisfeiler-lehman graph kernels." *Journal of Machine Learning Research*, vol. 12, no. 9, 2011.

[18] B. Ru, X. Wan, X. Dong, and M. Osborne, "Interpretable neural architecture search via bayesian optimisation with weisfeiler-lehman kernels," *arXiv preprint arXiv:2006.07556*, 2020.

[19] B. K. Thandri and J. Silva-Martínez, "A robust feedforward compensation scheme for multistage operational transconductance amplifiers with no miller capacitors," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 2, pp. 237–243, 2003.

[20] X. Peng, W. Sansen, L. Hou, J. Wang, and W. Wu, "Impedance adapting compensation for low-power multistage amplifiers," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 2, pp. 445–451, 2010.