Learn-by-Compare: Analog Performance Prediction using Contrastive Regression with Design Knowledge

Zihu Wang*
zihu_wang@ucsb.edu
University of California, Santa Barbara
Santa Barbara, California, USA

Karthik Somayaji N.S.*
karthi@ucsb.edu
University of California, Santa Barbara
Santa Barbara, California, USA

Peng Li
lip@ucsb.edu
University of California, Santa Barbara
Santa Barbara, California, USA

Abstract

This paper introduces Learn-by-Compare (LbC), a novel approach for analog performance modeling by employing semi-supervised contrastive regression. LbC employs a deep neural network encoder to come up with latent representations of sizing solutions by comparing similarity/dissimilarity of the underlying performance. Leveraging two levels of transistor level sizing data augmentation (DA), namely LS-DA and GS-DA, LbC produces new data samples by employing design knowledge. Experimental results highlight LbC's superior predictive accuracy compared to traditional regression methods. Offering a streamlined semi-supervised learning methodology, LbC effectively incorporates simple design knowledge and representation learning for efficient analog performance modeling.

Keywords: Analog Performance Modeling, Representation Learning, Contrastive Regression Learning, Learning with Design Knowledge

1 Introduction

Modeling the performance of analog circuits stands as a critical aspect in the domain of integrated circuit (IC) design and verification. The essence of analog performance modeling lies in establishing precise connections between the performance of an analog circuit and its sizing parameters. Traditional methods use analytical equations to detail these performance dynamics in relation to sizing (as discussed in [1, 6]), which works well for smaller-scale circuits. However, for complex designs at advanced technology nodes, the application of analytical equations to predict performance becomes increasingly challenging, primarily due to the intricate complexity in representing such analytical formulations.

On the other hand, black-box performance models can be built from circuit simulation data, e.g., collected from a SPICE simulator that assesses the intricate relationship between analog sizing and performance. However, with the escalating complexity of designs, simulation-based data collection encounters challenges such as prolonged run times. In these instances, users are constrained to a limited number of high-accuracy samples from the circuit simulator. To address this, black-box machine learning (ML) models have become popular for approximating analog circuit performance, offering rapid performance modeling and prediction. [3] use polynomial fitting on the simulation data to obtain regression models capable of predicting new performance values. The work by [7] builds simple circuit performance models using neural networks for analog performance modeling. [9] use graph representations in conjunction with neural networks to use topological information for efficient performance modeling. However, the effectiveness of these large neural network-based models heavily relies on the availability of high-quality training data. With limited data, issues like overfitting can lead to inaccurate predictions. This underscores the need for analog performance modeling tools that require minimal high-accuracy labeled data and ensure quick inference.

In this paper, we introduce a novel analog performance modeling approach `Learn-by-Compare` , where latent representation learning is performed to ease the regression learning from sparse simulation data. In addition, we propose `GS-DA` and `LS-DA` , two novel analog data augmentation techniques for generating analog data from the existing dataset by analog design knowledge. Furthermore, the proposed augmentations are integrated into `Learn-by-Compare` to achieve state-of-the-art performance in analog performance modeling relying on limited data. The key contributions of this work are as follows:

1. Propose a novel contrastive regression approach for analog performance modeling, emphasizing capturing the continuous nature of analog data in a latent representation space.
2. Design a novel global sizing data augmentation technique (`GS-DA`) for generating analog data and corresponding performance, utilizing analog design principles, particularly simple analog scaling rules, to identify long-range correlations in the analog sizing space.
3. Introduce a local sizing data augmentation technique (`LS-DA`) that generates supplementary sizing data by exploiting the continuous nature of analog performance concerning sizing variations.
4. Experiment on benchmark circuits and evaluate `LbC` by various analog performance modeling tasks, which reveal significant improvements in accuracy over conventional baseline methods.

*Both authors contributed equally to this research.

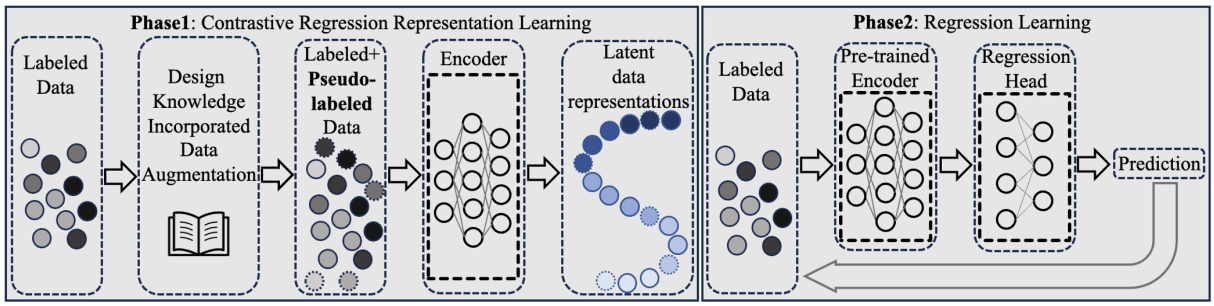


Figure 1. Learn-by-Compare framework consists two phases of training: (1) Training the encoder to learn latent representations; (2) Training a regression head on the top of the pre-trained encoder to learn to make predictions.

2 Analog Circuit Performance Modeling as a Regression Problem

In analog performance modeling, the design parameter vector $x \in \mathbb{R}^D$ is represented by the analog sizing solution. The corresponding circuit performance measure (referred to as the label), is represented by $y = g(x) \in \mathbb{R}^K$. The goal in analog performance modeling is to construct a model $f : \mathbb{R}^D \to \mathbb{R}^K$, that approximates the true circuit behavior function $g : \mathbb{R}^D \to \mathbb{R}^K$. In modern machine learning, the analog performance modeling function f is learned by a parameterized neural network with parameters θ. The output $\hat{y} = f_\theta(x)$ of the parameterized function is optimized to approximate the true performance measure y by minimizing a loss function $\mathcal{L} : \mathbb{R}^K \times \mathbb{R}^K \to \mathbb{R}$.

$$\theta^* = \arg\min_{\theta} \mathcal{L}(f_\theta(x), y) \tag{1}$$

In conventional regression learning methods, commonly used loss functions include the *L1* loss, *MSE* loss, *Huber* loss, etc.

3 Semi-supervised Contrastive Regression for Analog Performance Prediction

3.1 Contrastive Regression Framework

Deep regression models usually learn in an end-to-end fashion without emphasis on the latent representation distribution. To fully utilize the limited amount of circuit simulation data, we introduce an additional training phase coupled with various data augmentation techniques for learning the representation distribution of the sizing solution space during analog performance modeling.

Instead of training an end-to-end regression model $f(\cdot)$, we construct the model as a composition function consisting of an encoder $v(\cdot)$ and a regression head $h(\cdot)$, i.e., $f = h \circ v$. Here, the encoder $v_{\theta_e}(\cdot) : \mathbb{R}^D \to \mathbb{R}^M$, parameterized by θ_e, maps the sizing solution to a latent M-dimensional representation space, and the parameterized regression head $h_{\theta_h}(\cdot) : \mathbb{R}^M \to \mathbb{R}^K$ uses these latent representations to predict circuit performance. The training procedure thus consists of two phases and is depicted in Figure 1: (1) Learning good latent representations by training the encoder; (2) Connecting a regression head to the pre-trained encoder and tuning the model to learn to make predictions.

Learning an effective encoder is key to the overall performance of the model. In order to make full use of the simulation data and better train the encoder, we propose several novel design knowledge incorporated data augmentation approaches for generating 'pseudo-labeled' data from which the encoder can be better trained. The proposed augmentations are discussed in detail in Section 4. With augmentation functions $T_x(\cdot)$ and $T_y(\cdot)$ and labeled simulation data from the labeled dataset $(x_i, y_i) \in \mathcal{D}_l = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, new samples $(x', y') = (T_x(x_i), T_y(y_i))$ are generated to form $\mathcal{D}_u = \{(x'_1, y'_1), \ldots, (x'_m, y'_m)\}$. Such new samples are termed as 'pseudo-labeled' dataset. The size of the original dataset is thus enlarged by introducing pseudo-labeled data. A proposed LbC loss $\mathcal{L}_{LbC}$ is formulated to learn the optimal encoder from the enlarged 'semi-supervised' dataset $\{\mathcal{D}_l, \mathcal{D}_u\}$.

$$\theta_e^* = \arg\min_{\theta_e} \mathcal{L}_{LbC}(\theta_e | \mathcal{D}_l, \mathcal{D}_u) \tag{2}$$

We describe in detail the construction of $\mathcal{L}_{LbC}$ in Section 3.2. After encoder training in Phase 1, a regression head is connected to the encoder to learn to make predictions on the performance in Phase 2 by calling a regression loss $\mathcal{L}_{regression}$ (e.g., L1 loss, MSE loss, and Huber loss).

$$\theta_h^* = \arg\min_{\theta_h} \mathcal{L}_{regression}(h_{\theta_h}(v_{\theta_e}(x)), y | (x, y) \in \mathcal{D}_l) \tag{3}$$

3.2 Learning Latent Representations by Comparing Label Distance

At the core of the above described contrastive regression is the representation learning in Phase 1. However, traditional contrastive learning is developed in the context of discrete classification tasks [2], which is not capable of learning representations that capture the continuous nature of circuit performance in the latent representation space. Although a recent work has shed light on contrastive learning in visual regression problems [8], contrastive regression and its effectiveness have not been explored extensively in circuit performance modeling. We thus introduce `Learn-by-Compare`

(LbC), a contrastive regression approach that finds application in accurately predicting continuous valued circuit performance metrics, while retaining appealing properties of contrastive learning. In LbC , the latent space similarity is reflected by the data label distance, i.e., data samples with similar labels (performance values) are designed to have similar latent representations, so as to ultimately achieve superior accuracy for circuit performance modeling.

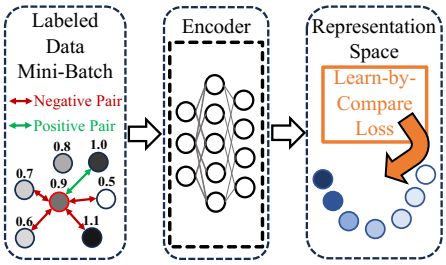


Figure 2. Pairs of data are **compared** according to their label distance. LbC loss guides the representation learning procedure.

Formally, given a mini-batch of N labeled data $\mathcal{B} = \{x_1, x_2, \ldots, x_N\}$ and their corresponding labels (i.e., circuit performance values) from the semi-supervised dataset, the label distance from each sample to all the other samples is calculated by means of the Euclidean distance. The representations of the batch $\mathcal{B}_z = \{z_1, z_2, \ldots, z_N\}$ are calculated by feeding data to the encoder, i.e., $z_i = v_{\theta_e}(x_i)$. The label distance from the i_{th} sample label to the j_{th} one is denoted by $d_{i,j}$. Next, in order to generate label distance aware representations, positive and negative pairs are formed in the batch according to the label distance. Positive pairs constitute sizing solution pairs whose performance values are similar, while negative pairs constitute sizing solution pairs whose performance values are dissimilar. Furthermore, such positive pairs are pairs of data samples whose latent space similarities are maximized, while negative pairs are pairs of data samples whose latent space similarities are minimized. As shown in Figure 2, when a positive pair is formed between a chosen sample z_i and another sample z_j, negative pairs are formed between z_i and all other samples z_k, such that $d_{i,k} > d_{i,j}$. Specifically, we denote the set of negative samples for a positive pair z_i, z_j as $\mathcal{N}_{i,j} = \{z_k | k \neq i, d_{i,k} > d_{i,j}\}$. We define a negative log-likelihood loss for the positive pair z_i and z_j as follows:

$$\ell_{i,j} = -log \frac{exp(sim(z_i, z_j)/\tau)}{\sum_{z_k \in \mathcal{N}_{i,j}} exp(sim(z_i, z_k)/\tau)} \tag{4}$$

where $sim(\cdot,\cdot)$ normalizes two given representation vectors and calculates the cosine similarity between them. τ is a temperature hyperparameter. Intuitively, minimizing $\ell_{i,j}$ is thus equivalent to pulling together the latent space representations, z_i and z_j while simultaneously pushing apart negative pair representations z_i and z_k.

In order to avoid bringing together very dissimilar samples, for a given sample z_i, positive pairs are formed between z_i and all other samples $z_j \in \mathcal{P}_i = \{z_j | j \neq i, d_{i,j} < \eta\}$, where η is a threshold hyperparameter for filtering out samples with dissimilar labels. We form positive pairs between z_i and all eligible samples z_j in the batch and define the per-sample LbC loss as follows:

$$\ell^i_{LbC} = -\frac{1}{\|\mathcal{P}_i\|} \sum_{z_j \in \mathcal{P}_i} log \frac{exp(sim(z_i, z_j)/\tau)}{\sum_{z_k \in \mathcal{N}_{i,j}} exp(sim(z_i, z_k)/\tau)} \tag{5}$$

Intuitively, given the i_{th} sample, its label distance to all other samples is **compared**. z_j's that are chosen as positive samples are brought close to z_i in the representation space. All z_k's whose label distance is greater than z_j are required to be aligned farther away from z_i than z_j is in the latent space. This process is pictorially depicted in Figure 2, where latent representations corresponding to a performance value of 0.9 form positive pairs with latent representations whose performance values are similar (0.8, 1.0 etc.), while they form negative pairs with latent representations whose performance values are dis-similar (1.1, 0.7, 0.6, 0.5 etc). Finally, the ℓ_{LbC} is summed up to form the *LbC loss* in mini-batches.

$$\begin{aligned} \mathcal{L}_{LbC} &= \frac{1}{\|\mathcal{B}\|} \sum_{z_i \in \mathcal{B}} \ell^i_{LbC} \\ &= -\frac{1}{\|\mathcal{B}\|} \sum_{z_i \in \mathcal{B}} \frac{1}{\|\mathcal{P}_i\|} \sum_{z_j \in \mathcal{P}_i} log \frac{exp(sim(z_i, z_j)/\tau)}{\sum_{z_k \in \mathcal{N}_{i,j}} exp(sim(z_i, z_k)/\tau)} \end{aligned} \tag{6}$$

4 Data Augmentation for Analog Performance Prediction

4.1 Global Sizing Data Augmentation (GS-DA)

To overcome the problem of scarce data during analog performance modeling, we propose Global Sizing Data Augmentation (GS-DA), a data augmentation approach that enhances the number of data points to refine the representations learned by the encoder. GS-DA uses approximate simple scaling rules in the analog sizing space to generate new simulation data in the form of pseudo-labeled data. The simple approximate scaling rules in traditional analog design suggest how the performance values scale as the input sizing solutions scale. For a sizing solution $x \in \mathbb{R}^D$ and a corresponding performance value vector $y \in \mathbb{R}^K$ (denoted as $x \rightarrow y$), simple analog design scaling rules specify relationships of the form:

$$x \rightarrow y \implies \alpha \cdot x \rightarrow \beta \cdot y \tag{7}$$

for some $\alpha \in \mathbb{R}^D, \beta \in \mathbb{R}^K$.

In large scale circuits, these rules may be approximate, but provide valuable cues on the relationship between the sizing solution and the ranges of performance values they map to. We illustrate the use of such scaling rules on the two-stage differential amplifier circuit. A schematic of the same is provided in Figure 3(a). We track three metrics, namely the

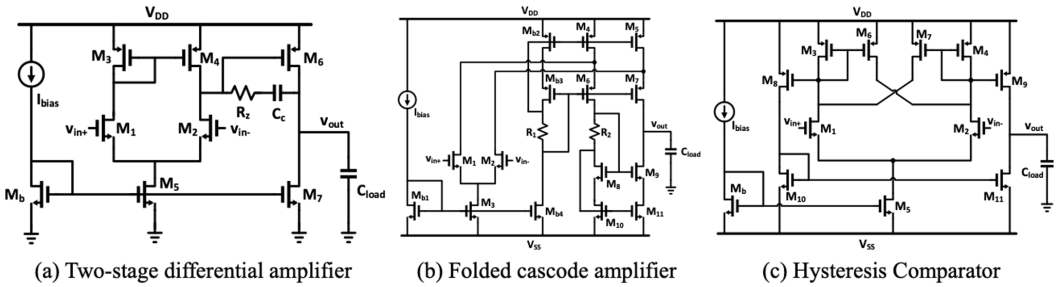


Figure 3. Schematic of the three circuits used for experiments.

gain, the unity gain frequency (UGF) and the common-mode rejection ratio (CMRR). The gain, CMRR and UGF [1, 6], of the two-stage differential amplifier in terms of the small signal parameters are defined as:

$$\text{Gain} = \left(\frac{g_{m_2}}{g_{ds_2} + g_{ds_4}}\right) \cdot \left(\frac{g_{m_7}}{g_{ds_7} + g_{ds_6}}\right) \tag{8}$$

$$\text{CMRR} = \frac{g_{m_1}^2}{(g_{ds_3} + g_{ds_4}) \cdot g_{ds_5}} \tag{9}$$

$$\text{UGF} = \frac{g_{m_2}}{C_c} \tag{10}$$

Scaling the sizing solution x, corresponds to scaling the W/L ratio of the corresponding transistors or scaling the capacitors, which scales the small signal parameters. This in turn scales performance metrics such as the gain, UGF and CMRR for instance. In Figure 4, we illustrate such simple scaling rules for the two-stage differential amplifier circuit. Scaling rules based on Equations 8, 9, 10, suggest that all the three metrics remain unchanged while scaling the W/L ratio (i.e. scaling the sizing solution). In Figure 4, we illustrate the value of the metric (or equivalently the figure of merit (FOM)) y, corresponding to the un-scaled sizing solution x in Equation 7 on the X-axis. The Y-axis indicates the value of the metric $\beta \cdot y$ corresponding to the scaled sizing solution $\alpha \cdot x$. All three illustrations in Figure 4 verify that the simple scaling rules defined through design knowledge hold in practice.

Next, we propose to use these scaling laws to increase the number of 'pseudo-labeled' data points, i.e. for every labeled data pair (x, y), we populate approximately accurate $(T_x(x), T_y(y)) = (\alpha \cdot x, \beta \cdot y)$ pairs as pseudo-labeled data to the current dataset.

In particular the scaling rules applied to the current sizing solutions, x, can generate new sizing solutions, $\alpha \cdot x$, distant from the current ones with approximately known values of performance for the newer sizing solutions due to the use of design knowledge. Such pseudo-labeled data in combination with the original training data is used to provide greater number of data points to train the encoder architecture in Phase 1. By varying the scaling parameter α, GS-DA achieves global coverage of the sizing solution space and extrapolates to unseen regions of the sizing solution space. GS-DA also provides accurate labeling for such unseen sizing solutions by employing simple scaling rules.

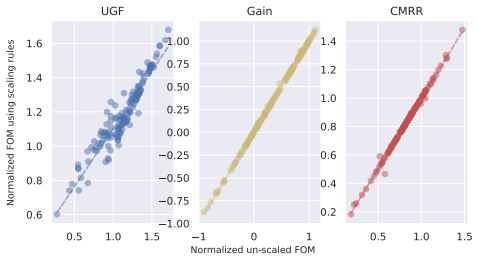


Figure 4. Simple, approximate scaling rules verify that scaled sizing solutions provide scaled performance values

4.2 Local Sizing Data Augmentation (LS-DA)

In global sizing data augmentation (GS-DA) long-range correlations in the sizing solution space are captured through simple scaling rules. In contrast, we additionally propose a local data augmentation technique called local sizing data augmentation (LS-DA). In LS-DA , new sizing solutions $x' \in \mathbb{R}^D : x' = x + \Delta x$ are generated from existing sizing solutions $x \in \mathbb{R}^D$ by introducing local perturbations $\Delta x \in \mathbb{R}^D$.

For small perturbations around a sizing solution, the performance values stay consistent. We exploit this local continuity property of performance values to generate new performance values for the newly generated sizing solution x'. Thus LS-DA generates multiple pseudo-labeled data in the neighborhood of the training data. Similar to GS-DA , the LS-DA approach also increases the number of pseudo-labeled data to train the encoder architecture in Phase 1.

In Figure 5, we illustrate the nature of data augmentations GS-DA and LS-DA with a simple example. We consider the two-stage differential amplifier circuit (Figure 3(a)) in which only the normalized width of transistor M2 is taken to be the 1-dimensional sizing solution, while the normalized gain is taken to be the 1-dimensional performance value. The sizing solutions used as part of the training dataset are indicated in the range $(0.5, 1)$. LS-DA covers new sizing solutions in the vicinity of the training sizing solutions. However, due to the scaling associated with GS-DA , it discovers new sizing solutions that are distant to the training sizing solutions and extrapolates to unseen normalized sizing solutions in the

Table 1. Performance comparison of LbC and existing regression learning methods. 100 simulation data samples are used for training each method. MAE error is used as the metric.

	2-Stage			Folded Cascode			Hysteresis		
Methods	ugf	gain	cmrr	ugf	gain	cmrr	gain	bandwidth	hyst. err.
L1	0.3527	0.3684	0.1881	0.1840	0.3402	0.3253	0.2519	0.1822	0.1747
MSE	0.4666	0.3760	0.2088	0.1995	0.3534	0.3497	0.2768	0.1974	0.1819
Huber	0.3798	0.3723	0.1925	0.1812	0.3599	0.3402	0.2697	0.1933	0.1820
GPR	0.3552	0.4705	0.4311	0.1668	0.3266	0.5079	0.2203	0.2151	0.1618
LbC(**Ours**)	**0.2631**	**0.2723**	**0.1268**	**0.0803**	**0.2416**	**0.2448**	**0.1751**	**0.1106**	**0.0823**

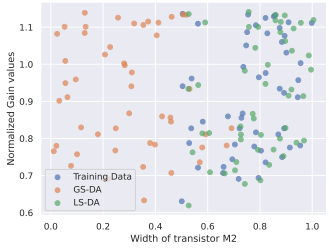


Figure 5. Simplified illustration of the coverage of the normalized sizing solution space (consisting of only the width of transistor M2 in Figure 3(a)) for the two-stage differential amplifier. GS-DA achieves global long range coverage far away from the training sizing solution data, while LS-DA achieves local coverage around the training data.

range $(0, 0.5)$ too. The scaling rules define the performance values on the Y-axis. Thereby, the proposed data augmentation techniques (LS-DA and GS-DA) demonstrate extensive and global coverage of the sizing solution space, which generate pseudo-labeled data for training the encoder architecture in Phase 1. This has the overall effect of increasing the accuracy of analog performance modeling by ensuring richer representations of the sizing solution space.

5 Experiments

5.1 Experiment Settings

We evaluate the effectiveness of the proposed LbC by comparing its performance on three different circuits which are designed under a commercial 90nm CMOS technology framework: a two-stage differential amplifier (2-Stage), a folded cascode amplifier (FC), and a hysteresis comparator (Hyst) topology, as illustrated in Figure 3. 2-Stage, FC, and Hyst are designed with 14, 18, and 12 distinct parameters. We compare LbC with other popular existing regression methods L1 loss-based, MSE loss-based, Huber loss-based methods, and Gaussian Process Regression [4] (GPR). Mean Absolute Error (MAE) is adopted to demonstrate performance of all methods. We focus on UGF, gain, and CMRR prediction for 2-Stage and FC. For Hyst, bandwidth , gain and hysteresis error are considered as performance metrics.

Dataset. Synopsys HSpice is employed for simulating all circuits. Each metric, as indicated above, is first normalized to fit into a range of $[-1, 1]$.Similarly, the sizing solution space is normalized to a range $[0, 1]$. We generate labeled training data samples $(x, y) \in \mathcal{D}_l$, where the input normalized sizing solution $x \in (0.5, 1)^D$. To assess the generalizability of trained models, we generate test sample $(x_t, y_t) \in \mathcal{D}_t$ where x_t are randomly sampled from a larger range $(0, 1)^D$.

Table 2. Performance comparison of LbC and existing methods trained on different amount of data. MAE error is used as the metric.

	Number of simulation data				
method	50	100	200	500	1000
L1	0.3909	0.3215	0.3112	0.2767	0.2353
MSE	0.3813	0.3279	0.3189	0.2896	0.2382
LbC(**Ours**)	**0.2794**	**0.2592**	**0.2343**	**0.1952**	**0.1546**

LbC *training settings.* The model of LbC consists of two parts, an encoder and a regression head. We adopt a two-layer MLP (multi-layer perceptron) as the encoder whose output embedding is 32-dimensional. The regression head is another two-layer MLP for prediction. In phase 1, the encoder is trained for 2000 epochs with an Adam optimizer with a learning rate (lr) of 10^{-4}. In phase 2, the regression head is connected to the trained encoder and trained for 3000 epochs with an Adam optimizer of $lr = 5 \times 10^{-4}$. A cosine scheduler is adopted for scheduling the lr in both phases. The threshold hyperparameter η is set to 0.2, and temperature is set to $\tau = 2.0$.

Training setting of baseline methods. For fair comparisons, L1 loss-based, MSE loss-based, and Huber loss-based methods are designed with a four-layer MLP with 32-dimensional latent embedding, identical to the model architecture of LbC . Models are trained for 5000 epochs with an Adam optimizer of $lr = 5 \times 10^{-4}$, and the lr is scheduled by a cosine scheduler. For GPR, we use Radial Basis Function as the kernel function.

5.2 Results

Prediction accuracy comparison. Table 1 demonstrates the comparison between LbC and other methods in predicting various performance measures. Models are trained with only 100 data samples and tested on 20000 samples. Table 2 shows

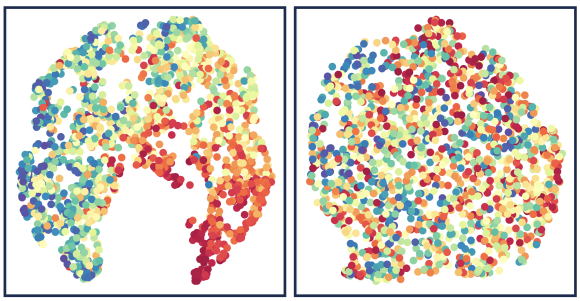
Table 3. Performance comparison of LbC with and without data augmentation. MAE error is used as the metric.

method	2-Stage	Folded Cascode	Hysteresis
L1	0.3741	0.3546	0.2618
LbC(w/o aug.)	0.3037	0.2771	0.1902
LbC(w/ aug.)	**0.2709**	**0.2314**	**0.1750**

Table 4. Comparison of LbC with and without GS-DA . Folded cascode data is used for training and testing. Training and testing dataset cover different areas in sizing solution space.

method	ugf	gain	cmrr
L1	0.2591	0.4560	0.3632
LbC(w/o GSDA)	0.1347	0.3206	0.2198
LbC(w/ GSDA)	**0.0850**	**0.2535**	**0.1885**

the performance comparison in predicting all three performance measures (used as weighted figures of merit (FOM)) of 2-Stage for methods trained on different amounts of data samples. The averaged error of predicting all 3 measures is reported. **LbC achieves up to a two-fold reduction in MAE** which suggests that learning latent data representation enhances the encoder's performance in downstream analog performance prediction. We further compare the latent data representations in Figure 6. 1700 2-stage test dataset's sample representations are plotted using UMAP [5], a dimension reduction technique for visualizing high dimensional manifold on a plane. Different colors represent corresponding data label value (analog performance value), similar values share similar colors. Compared to MSE-based method which fails to understand the underlying continuous information in analog performance, LbC captures data continuity w.r.t. analog performance values.



(a) Learn-by-Compare (b) MSE-based regression

Figure 6. Representation space data distribution comparison of LbC and MSE-based regression method.

Effectiveness of proposed augmentations. To evaluate the LbC framework's augmentations, we compare LbC trained with and without these augmentations, as shown in Table 3, including a comparison with L1 loss-based regression. The results, presented as MAE of the FOM of each circuit, show notable accuracy improvements with LbC even without augmentations, which are further enhanced by the proposed augmentations due to increased data distribution coverage.

To assess GS-DA 's impact, we test LbC with and without GS-DA on distinct test datasets ($\mathcal{D}_l = (x_t, y_t)|x_t \in (0, 0.5)$, different from the training set $\mathcal{D}_l = (x, y)|x \in (0.5, 1)$). As Table 4 shows, LbC notably improves the performance on diverse distribution areas, with GS-DA reducing the error rate by 1.8 to 3 times compared to the L1 loss-based method.

6 Conclusion

In this work, we introduce LbC , a novel semi-supervised contrastive regression framework designed for precise analog performance modeling in scenarios with limited data. Learn-by-Compare leverages representation learning to effectively map the sizing solution space in alignment with key performance metrics. To address the challenge of sparse training data, we develop local and global data augmentation strategies that incorporate analog designers' expertise, enabling exploration of performance-correlated regions. In addition to the improved accuracy in performance modeling,s The versatility of our representation learning and data augmentation methods could also be applied to analog optimization in Bayesian optimization and reinforcement learning contexts.

7 Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. 1956313.

References

[1] Chan David Johns Kenneth W Martin Carusone, Tony. 2012. *Analog Integrated Circuit Design* (2 ed.). John Wiley Sons, USA.

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

[3] Walter Daems, Georges Gielen, and Willy Sansen. 2003. Simulation-based generation of posynomial performance models for the sizing of analog integrated circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 22, 5 (2003), 517–534.

[4] Hanbin Hu, Peng Li, and Jianhua Z. Huang. 2018. Parallelizable Bayesian Optimization for Analog and Mixed-Signal Rare Failure Detection with High Coverage. In *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. 1–8.

[5] Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* (2018).

[6] Behzad Razavi. 2000. *Design of Analog CMOS Integrated Circuits* (1 ed.). McGraw-Hill, Inc., USA.

[7] Sayed Mohammad Ali Zanjani and Alireza Pourkhalili. 2024. Design of a Two-Stage Operational Amplifier Using Artificial Neural Network. *Journal of Intelligent Procedures in Electrical Technology* 2, 58 (2024).

[8] Kaiwen Zha, Peng Cao, Jeany Son, Yuzhe Yang, and Dina Katabi. 2023. Rank-N-Contrast: Learning Continuous Representations for Regression. In *Thirty-seventh Conference on Neural Information Processing Systems*.

[9] Zhenxin Zhao and Lihong Zhang. 2021. Efficient performance modeling for automated CMOS analog circuit synthesis. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 29, 11 (2021), 1824–1837.