Late Breaking Results: Encoder-Decoder Generative Diffusion Transformer Towards Push-Button Analog IC Sizing

Filipe Azevedo, Nuno Lourenço and Ricardo Martins
Instituto de Telecomunicações, Lisboa, Portugal / Instituto Superior Técnico – Universidade de Lisboa, Lisboa, Portugal
ricmartins@lx.it.pt

2025 62nd ACM/IEEE Design Automation Conference (DAC) | 979-8-3315-0304-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/DAC63849.2025.11133224

***Abstract*—In this paper, disruptive research using generative diffusion models (DMs) with an attention-based *encoder-decoder* backbone is conducted to automate the sizing of analog integrated circuits (ICs). Unlike time-consuming optimization-based methods, the *encoder-decoder* DM is able to sample accurate solutions at push-button speed by solving the inverse sizing problem. Experimental results show that the proposed model outperforms the most recent deep learning-based techniques, presenting higher generalization capabilities to performance targets not seen during training.**

Keywords—analog integrated circuits, diffusion models, electronic design automation, inverse sizing problem

I. Introduction

Even though tremendous efforts were made by the electronic design automation community to automate the analog IC sizing task in the past decades, the classical "experience and trial" design methods carried iteratively by designers are still a common practice. Analyzing a circuit/system functional behavior given a topology and its component values can be seen as the direct sizing problem, while synthesizing the components values to fulfill certain functional metrics can be seen as the inverse sizing problem [1]. The majority of existent automatic sizing techniques, either knowledge-based or optimization-based [2], focus on solving the direct problem, iterating the components' dimensions and analyzing its impact on the overall behavior. Further works, based on machine learning (ML), have also supported the direct sizing problem either by replacing the simulator [2] or by speeding-up the optimization mechanism [3]. On a different direction, some preliminary efforts were focused on solving the inverse sizing problem [1][4][5][6], exploiting deep learning (DL) models that produce sizing solutions when requested with a set of performances. At the time of writing, solving the inverse problem is the only way to pursue push-button speed sizing for the analog domain.

This work follows the most recent efforts on automatic analog IC sizing based on ML/DL [4]-[6], in particular, with the use of generative artificial intelligence to solve the inverse sizing problem. When compared with other state-of-the-art works, the adopted DM has shown superior generalization beyond its training data. Additionally, while the authors of [6] employed a simple multilayer perceptron (MLP) as the DM backbone, here an attention-based encoder-decoder is chosen instead. The complexity of the model increases as shown in Fig. 1, but the accuracy of the generated points improves drastically, while keeping the same MLP's flexibility to be adapted to new circuit topologies & integration technologies.

II. Proposed Diffusion Transformer Architecture

A. Diffusion Models Foundations

DMs were introduced in [7] and its primary idea is to add random noise to the training data, and then train an artificial neural network (ANN) to predict the noise that was added. After training, new data can be simply generated by giving the ANN pure noise. DMs consist of 3 steps, the forward, reverse and sampling processes. The forward process corresponds to the addition of noise to the data over T sequential timesteps. A common way to do this is to sample noise from a Gaussian distribution and therefore the forward distribution q can be described by Eq. (1), with mean μ and standard deviation Σ, where β is the diffusion rate, a hyperparameter that controls the amount of noise that is added at each timestep t, x is the data, and I the identity matrix:

$$q(x_t|x_{t-1}) \coloneqq \mathcal{N}(x_t;\mu,\Sigma) \coloneqq \mathcal{N}\left(x_t;\sqrt{1-\beta_t}x_{t-1},\beta_t I\right) \quad (1)$$

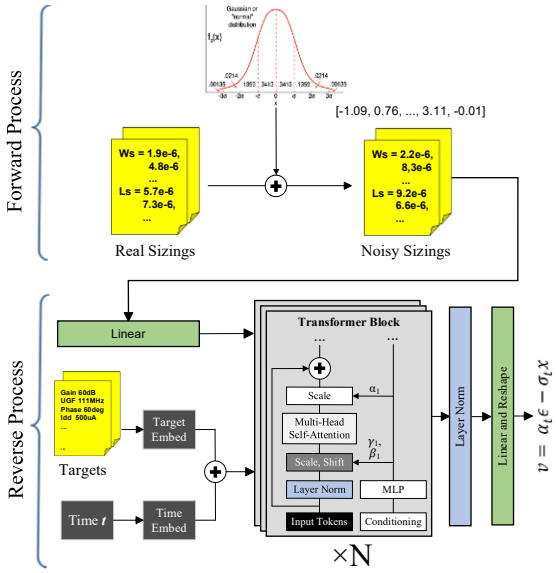

Fig. 1. Attention-based transformer for analog IC sizing.

The reverse process aims to reverse the forward process by training an ANN to predict the noise ϵ_θ added to the data. It can be summarized by Eq. (2), where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_s^t \alpha_s$:

$$p_\theta(x_{t-1}|x_t) \coloneqq \mathcal{N}\left(x_{t-1};\mu_\theta(x_t,t),\Sigma_\theta(x_t,t)\right)$$
$$\coloneqq \mathcal{N}\left(x_{t-1};\frac{1}{\sqrt{\alpha_t}}\left(x_t-\frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t,t)\right),\beta_t I\right) \quad (2)$$

After training, the ANN can generate entirely new data when fed with random noise, and iteratively removing the predicted noise from the initial random noise: this is the sampling process. Lastly, in order to guide the model to generate meaningful solutions during the sampling process, context-free guidance [8] is the most common strategy. The ANN is trained conditionally and unconditionally at the same time by randomly dropping the class of the data. Then, during sampling, the ANN is requested to predict the noise with and without context. In this problem, the context would be the circuit performances. Both outputs are combined by Eq. (3), where hyperparameter w is the weight given to each output:

$$\bar{\epsilon}_\theta(x|y) = w\epsilon_\theta(x) + (1-w)\epsilon_\theta(x|y) \quad (3)$$

This research is funded by FCT/MCTES through national funds and, when applicable cofounded European Union (EU) funds under the projects UIDB/50008/2020 (DOI identifier 10.54499/UIDB/50008/2020) and ACTON (DOI identifier 10.54499/2023.11981.PEX), and also, by Sony Semiconductor Solutions (project GENERALISE).

B. *Transformer Backbone*

DMs are usually applied to image generation and restauration, however, here, a disruptive implementation is made for the inverse sizing problem of analog ICs, as a way to pursue push-button speed synthesis. The architecture of the proposed model uses an attention-based transformer as its backbone, as detailed in Fig. 1. First, noise is sampled from a Gaussian distribution and iteratively added to the sizing array. Then, this array passes through a linear layer and is fed to the transformer block. At the same time, embeddings are created for the performance targets and current timestep (since the model is trained to predict noise at each timestep). These embeddings are added together and also fed to the transformer block. Inside, the sizings and embeddings go through normalization, linear, multi-head Attention, ReLU and dropout layers. Finally, the resulting values undergo normalization and linear layers, and are reshaped for post-processing. The number of transformer blocks, N, is a hyperparameter, since multiple passes through attention layers can be beneficial.

It can also be seen in Fig. 1 that the model predicts an equation $v = \alpha_t \epsilon - \sigma_t x$ (where σ_t is the variance of the forward distribution), instead of only the noise ϵ that was mentioned in Section II.A. This has some advantages, the most important one being the possibility of achieving a true signal-to-noise ratio (SNR) of zero without causing a zero-division error at the last timestep T [9], which is important to prevent discrepancies between training and sampling: if the SNR is never zero, during training the model never sees pure noise, while that is the case at the beginning of sampling.

III. Experimental Results

The model was coded in python with the pytorch library, and tested on state-of-the-art operational transconductance amplifiers (OTAs) for biomedical (ECG) signals, shown on Fig. 2, for the TSMC65nm technology. The models were trained on datasets containing sizing solutions and their respective performances, and its details are shown in Table I. For each circuit, the width, length and number of fingers for each transistor, as well as the supply & bias voltages, comprise the design parameters. Before training the models, the datasets also underwent normalization, 2nd order polynomial feature expansion, and augmentation by a factor of 10. Also shown in Table I is the training time of the proposed model (DM-Transf.), as well as the models available on the repositories of [4] and [6] for the same datasets. The training was carried on an Intel i7-13700H CPU with an NVIDIA GeForce RTX 4060 Ti GPU. When comparing with [6], where the authors adopted DMs but with a simple MLP backbone, the transformer adds complexity to the model, hence the larger training time, but ultimately providing an improved guiding strategy.

For the DM-Transf. hyperparameters a timestep T=1000 was chosen, a cosine schedule for β, N=8 transformer blocks, guidance weight w=0.1 and the training was made for 100 epochs. For the other models, the hyperparameters chosen were the ones presented in the original references. In Table II, the sampling performance of the models can be thoroughly analyzed. For each OTA, 100 points were sampled for target values that were selected for the four metrics – power, input-referred noise (IRN), gain (G_{DC}), and bandwidth (BW). It is important to note that the selected target values were never observed simultaneously within the same data point during training. The objective is to have power and IRN lower than the target, while G_{DC} and BW should be higher. In Table II, the closest sampled solution for all the performance targets is displayed. It is observable that the DM-Transf. has a superior performance, at the cost of a slightly larger sampling time. It is the only model that can sample at least one feasible solution for each example, with DM-MLP and ANN failing on OTAs 2, 3 and 4. Additionally, a simulation-based technique was run for each target. An evolutionary algorithm [2] with a population of 512 elements optimized through 500 generations was used, where calling a simulator for each point to assess its performance may result in 256K calls. The times reported correspond to the amount of time taken to find at least one feasible solution. By observing Table II, the simulation-based solution did find feasible points for 3 out of 4 OTAs, however, it takes a non-negligible amount of time and computational power, and therefore, far from push-button, an aspect where the DM-Transf. exceeds.

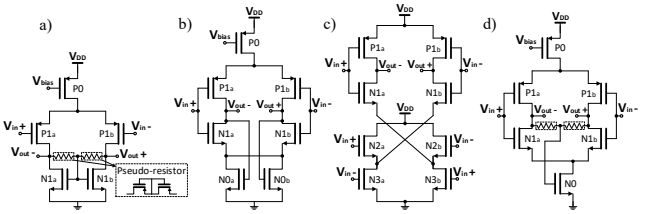

Fig. 2. Schematics of the: (a) OTA1, (b) OTA2, (c) OTA3 and (d) OTA4.

TABLE I. Dataset Details and Training Times

Topology	OTA1	OTA2	OTA3	OTA4
#Design Parameters	14	14	14	17
#Dataset samples	2024	1744	969	1016
ANN [4] training time	0.9 min	0.8 min	0.6 min	0.7 min
DM-MLP [6] training time	6 min	5 min	2.5 min	3 min
DM-Transf. training time	26 min	22 min	12 min	13 min

TABLE II. Performance of Sampled Designs

	power [nW]	IRN [μV]	G_{DC} [dB]	BW [KHz]	Accuracy	Time
OTA1	≤ 27	≤ 5.2	≥ 30	≥ 1.55	-	
Sim.-based [2]	21	5.1	31	1.94	-	35 min
ANN [4]	27	4.4	33	1.61	2%	0.10 sec
DM-MLP [6]	25	4.7	32	2.30	2%	3.60 sec
DM-Transf.	26	4.9	33	1.70	2%	7.39 sec
OTA2	≤ 21	≤ 4.5	≥ 29	≥ 2.90	-	
Sim.-based [2]	19	3.8	29	4.37	-	75 min
ANN [4]	19	**5.4**	29	3.92	0%	0.11 sec
DM-MLP [6]	19	2.8	35	**1.14**	0%	3.56 sec
DM-Transf.	20	4.0	32	3.60	14%	7.43 sec
OTA3	≤ 24	≤ 4.3	≥ 31	≥ 10.50	-	
Sim.-based [2]	24	2.7	**27**	10.80	-	2150 min
ANN [4]	24	3.5	**30**	12.06	0%	0.13 sec
DM-MLP [6]	23	2.9	**27**	17.00	0%	3.60 sec
DM-Transf.	22	3.9	31	11.25	3%	7.41 sec
OTA4	≤ 21	≤ 3.5	≥ 32	≥ 4.20	-	
Sim.-based [2]	19	2.8	32	5.29	-	224 min
ANN [4]	12	3.3	36	**2.04**	0%	0.12 sec
DM-MLP [6]	19	2.6	35	**2.65**	0%	3.60 sec
DM-Transf.	19	2.7	33	4.60	14%	7.46 sec

References

[1] P. Beaulieu, et al., "Analog rf circuit sizing by a cascade of shallow neural networks", *IEEE TCAD*, vol 42, no 12, pp. 4391–4401, Dec 2023.

[2] B. Liu, et al., "A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems", in *IEEE Trans. on Evolutionary Comp.*, vol. 18, no. 2, pp. 180–192, April 2014.

[3] J. Zhang, et al., "Automated Design of Complex Analog Circuits with Multiagent based Reinforcement Learning", in *DAC Conf.*, July 2023.

[4] N. Lourenco, et al., "On the exploration of promising analog ic designs via artificial neural networks," in *15th Int. Conf. on SMACD*, July 2018.

[5] M. Leibl and H. Graeb, "Optimizer-Free Sizing of OpAmps Leveraging Structural and Functional Properties," *20th Conf. on SMACD*, July 2024.

[6] P. Eid, et al., "Solving the Inverse Problem of Analog Integrated Circuit Sizing with Diffusion Models," in *20th Int. Conf. on SMACD*, July 2024.

[7] J. Sohl-Dickstein, et al., "Deep Unsupervised Learning using Nonequilibrium Thermodynamics," in *Proc. of the ICML*, July 2015.

[8] J. Ho and T. Salimans, "Classifier-Free Diffusion Guidance," in *Workshop on Deep Generative Models and Downstream Appl.*, 2021.

[9] S. Lin, et al., "Common Diffusion Noise Schedules and Sample Steps are Flawed," in *IEEE/CVF WACV*, Jan. 2024.