WAVEFORMER: A BEHAVIOR-AWARE TRANSFORMER FOR ANALOG PERFORMANCE PREDICTION VIA CIRCUIT RESPONSE PRETRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Analog-circuit response curves describe how observed voltages and currents vary with stimulus and physical coordinates, providing supervision beyond individual scalar specifications. We introduce WAVEFORMER, a framework that learns circuit representations from DC, complex AC, and transient responses and transfers its shared circuit–testbench backbone to scalar performance prediction. Shared target-aware circuit and function-aware testbench encoders condition three analysis-specific multiscale response heads, jointly trained with masked value and shape losses. The WAVEFORMER BENCHMARK pairs 33,695 parameter configurations across 30 topologies with response curves and supports evaluation of response fidelity, representation transfer, and model-guided sizing. On held-out configurations of known topologies, WAVEFORMER reduces response normalized root mean squared error (NRMSE) by 25.9–46.5% relative to DeepONet. With the same performance-predictor architecture and fine-tuning budget, response pretraining reduces Primary NRMSE by 4.7%, with improvements in DC and AC metrics but a slight degradation in transient metrics. The resulting performance predictor also guides SPICE-verified sizing, connecting response-supervised learning to circuit design.[1]

1 INTRODUCTION

Analog and mixed-signal circuits connect physical signals to digital computation in sensing, communication, and power-conversion systems (Gray et al., 2009). Their behavior depends on connectivity, device parameters, operating conditions, and stimulus (Gray et al., 2009; Xu et al., 2024). SPICE characterizes this behavior through DC operating points and transfer curves, AC frequency responses, and transient dynamics (Nagel, 1975; Kundert & Chang, 2006; Gray et al., 2009). Repeating these analyses across candidate designs and test conditions makes simulation a recurring computational bottleneck (Xu et al., 2024; Kundert & Chang, 2006).

Performance surrogates predict specifications such as gain, bandwidth, power, and settling time from design inputs (Dong et al., 2023; Poddar et al., 2025; Yang et al., 2025). These scalar outputs summarize selected properties under a given testbench. Response curves additionally describe how an observed voltage or current varies over sweep values, frequency, or time. This motivates our central question: ***Can learning target- and stimulus-conditioned response curves provide useful circuit representations for downstream performance prediction?***

PGNN learns representations from DC node voltages and evaluates their adaptation to new topologies and circuit properties (Hakhamaneshi et al., 2023). Other approaches investigate single-circuit operator learning, topology-conditioned transient prediction, and generative transient synthesis (Zhang et al., 2023; Yazici, 2026; Karipidis et al., 2026). Building on these directions, we study response curves as supervision for learning representations that can be adapted to scalar performance tasks. The learning problem connects a discrete circuit graph, source functions, and an observation target to a response over physical coordinates. We therefore examine both response fidelity and the benefit of transferring the learned shared backbone to a performance predictor.

[1] We will open-source the dataset and code upon publication.

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.

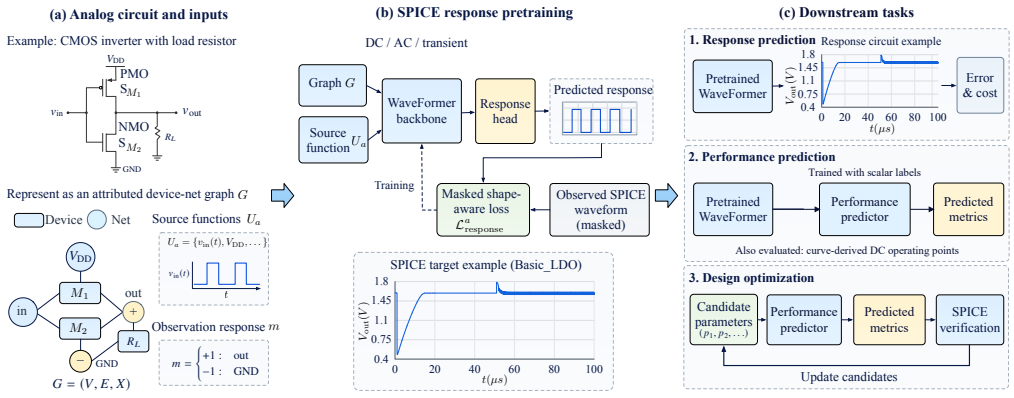


Figure 1: Overview of WAVEFORMER. (a) Circuit structure, sampled stimuli, and the observation target define the model inputs. (b) DC, AC, and transient responses provide pretraining supervision. (c) Evaluation covers response prediction, performance prediction using the transferred backbone, and circuit sizing with SPICE verification.

To study this problem, WAVEFORMER BENCHMARK pairs parameterized circuit graphs and sampled sources with DC, complex AC, and transient responses. Its common data interface and configuration-disjoint split connect response pretraining to three evaluation tasks (Figure 1). *Response Curve Prediction* measures curve fidelity and inference cost on held-out configurations. *Performance Prediction* evaluates backbone transfer to scalar metric regression and separately assesses response-derived DC operating points. *Design Optimization* tests the resulting performance predictor in SPICE-verified sizing.

WAVEFORMER takes a circuit graph G, sampled source functions U_a, and a signed observation target m as inputs. A target-aware circuit encoder and a function-aware testbench encoder form a shared backbone. Three independent, analysis-specific multiscale response heads combine branch–trunk and Fourier-residual paths and are trained jointly with masked value and shape losses. For performance prediction, we transfer both encoders and their learnable fusion and fine-tune them with a new metric head using scalar labels.

Our contributions are:

- **Response supervision and evaluation.** WAVEFORMER BENCHMARK connects DC, complex AC, and transient response supervision to performance prediction and sizing through a common data interface. Its fixed split evaluates unseen parameter configurations of 30 known topologies.
- **Conditional response learning and backbone transfer.** WAVEFORMER learns response curves from circuit structure, observation targets, and source functions using analysis-specific multiscale decoders and masked value and shape losses. The learned shared backbone initializes a jointly fine-tuned scalar performance predictor.
- **Response fidelity, transfer, and design utility.** Compared with the cutting-edge baselines, WAVEFORMER improves both amplitude accuracy and shape fidelity across the three analyses. Controlled initialization comparisons quantify analysis-dependent pretraining benefits, while SPICE-verified sizing evaluates the performance predictor's design utility.

2 RELATED WORK

PGNN, KCLNet, and Ckt2Vec learn circuit representations using DC voltages, electrical equivalence, and device I–V curves, respectively (Hakhamaneshi et al., 2023; Xu et al., 2026b;a). CktGNN, INSIGHT, and ZeroSim predict scalar specifications through hierarchical graphs, metric dependencies, and cross-topology embeddings (Dong et al., 2023; Poddar et al., 2025; Yang et al., 2025). DeepONet- and LSTM-based circuit models learn stimulus–response mappings for particular circuits or subcircuits (Zhang et al., 2023; Hu et al., 2025). These approaches address complementary aspects of circuit modeling: electrical structure, behavioral supervision, and the prediction interface.

Table 1: Reported capabilities of circuit representation and prediction methods (dashes indicate capabilities not reported in the cited work).

Capability	PGNN	KCLNet	Ckt2Vec	INSIGHT	ZeroSim	DeepONet	LSTM	WAVEFORMER (Ours)
Circuit-graph encoding	✓	✓	✓	–	✓	–	–	✓
Sampled-excitation conditioning	–	–	–	–	–	✓	✓	✓
All three response types	–	–	–	–	–	–	–	✓
Circuit-metric prediction	✓	–	–	✓	✓	–	–	✓

WAVEFORMER connects them by learning DC, complex AC, and transient responses with a shared circuit–testbench backbone and analysis-specific heads. Evaluating its transfer to scalar regression under a fixed predictor architecture and fine-tuning budget tests whether response supervision yields useful representations beyond the original curve-prediction task. As shown in Table 1, we compare reported capabilities with those evaluated for WAVEFORMER and Section D discusses the learning and evaluation scope.

3 PROBLEM FORMULATION AND BENCHMARK

Response prediction. For analysis $a \in \mathcal{A} = \{\mathrm{DC}, \mathrm{AC}, \mathrm{Transient}\}$, the inputs are an attributed circuit graph G, sampled source functions and roles U_a, observation target m, and physical query coordinate $q \in \mathcal{Q}_a$. The graph encodes connectivity and device parameters, m selects one net or an ordered differential pair, and q denotes a sweep value, frequency, or time. The prediction target is the registered SPICE observable:

$$\widehat{y}_a(q) = \mathcal{F}^a_{\theta_a}(G, U_a, m, q) \in \mathcal{Y}_a, \qquad q \in \mathcal{Q}_a. \tag{1}$$

$\theta_a = (\eta, \psi_a)$ combines shared backbone parameters η and analysis-specific head parameters ψ_a. DC and transient outputs are real; complex AC outputs use real and imaginary channels.

Benchmark design and evaluation. WAVEFORMER BENCHMARK pairs parameterized netlists and sampled sources with DC, complex AC, and transient responses, extending AnalogGym to response-level supervision (Li et al., 2024). It contains 33,695 configurations across 30 topologies; splits are disjoint by parameter configuration, with every topology represented in each partition. This shared split supports response-curve evaluation, backbone transfer to performance regression, and SPICE-verified sizing guided by the performance predictor. Sections A and C detail dataset construction, data formats, and evaluation protocols.

4 METHOD

WAVEFORMER learns from response curves and transfers its shared circuit–testbench backbone to performance prediction. During pretraining, shared target-aware circuit and function-aware testbench encoders condition three independent multiscale response heads at physical query coordinates (Figure 2). Masked value and shape losses jointly train the backbone and response heads. For performance regression, both pretrained encoders and their learnable fusion are fine-tuned with a newly initialized joint metric head that replaces the response heads; the resulting predictor is used for sizing (Section C.7).

The two encoders form an instance representation conditioned on the observation target and stimulus:

$$h_a = E_\eta(G, U_a, m) = [h_{G,a}; h_{U,a}]. \tag{2}$$

Here $h_{G,a}$ describes the circuit at the selected port, while $h_{U,a}$ describes the applied stimulus. We omit a on internal quantities where unambiguous.

4.1 TARGET-AWARE CIRCUIT ENCODING

Changing the observation port changes the prediction target even when the netlist is fixed. Terminal roles and declared matching relations also carry information beyond connectivity. The encoder in Figure 3 incorporates electrical relations into message aggregation and identifies the selected port in the readout. We represent the flattened netlist with device, net, and matching nodes. Device nodes

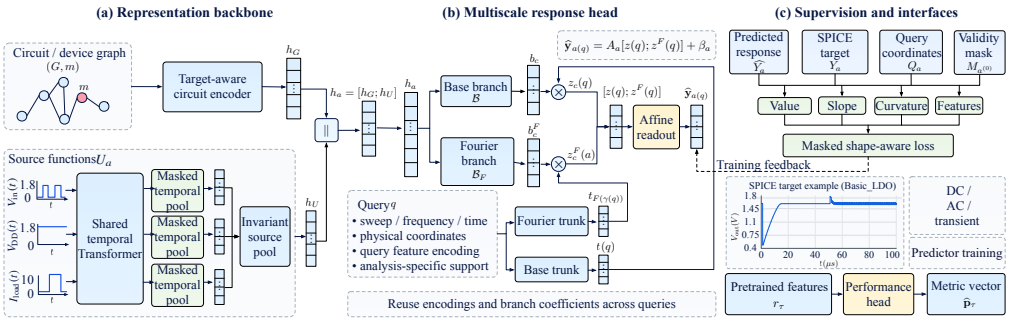


Figure 2: WAVEFORMER architecture. (a) Shared circuit and testbench encoders form h_a. (b) Each analysis-specific head decodes responses through base and Fourier-residual paths, reusing encodings and branch coefficients across queries. (c) Masked response pretraining and the performance head trained on the resulting representations.

carry classes and numerical parameters, net nodes carry testbench roles, and matching nodes connect each declared matching group. Messages pass in both directions along graph incidences, preserving terminal and matching-slot labels. Signed node features mark the observed net or the positive and negative nets of a differential target. For attention head ν, the message from j to i uses

$$e_{ij}^{(\nu)} = \frac{(W_Q^{(\nu)} h_i)^\top (W_K^{(\nu)} h_j)}{\sqrt{d_\nu}} + b_{\text{dir}}^{(\nu)}(d_{ij}) + b_{\text{term}}^{(\nu)}(r_{ij}) + b_{\text{slot}}^{(\nu)}(s_{ij}) + b_{\text{dist}}^{(\nu)}(\delta_{ij}), \tag{3}$$

where d_ν is the head dimension and $d_{ij}, r_{ij}, s_{ij}, \delta_{ij}$ denote direction, terminal, matching slot, and shortest-path distance in the augmented graph. Learned bias tables map these descriptors to attention offsets, with null labels for inapplicable relations. Attention thus depends on electrical relationships as well as node content. Global pooling and ordered positive/negative target embeddings form h_G; single-ended targets use a learned null embedding for the negative target. The global term supplies circuit-wide context, while the ordered target terms retain the measurement location and differential orientation. Section B.1 gives the complete attention, masking, and readout computation.

4.2 FUNCTION-AWARE TESTBENCH ENCODING

Changing a source function changes circuit behavior; reordering the source records leaves the testbench unchanged. The testbench encoder respects this distinction by modeling samples within each source before aggregating across sources. Let $H_s^{U,(0)}$ collect source-sample embeddings combining numerical values, semantic roles, and physical coordinates. A shared temporal Transformer processes the valid samples of each source; masked temporal pooling gives h_s^U, followed by source-order-invariant attention pooling:

$$h_U = \text{SourcePool}\left(\left\{\text{MaskedPool}\left(\text{Transformer}(H_s^{U,(0)})\right)\right\}_{s=1}^{S}\right). \tag{4}$$

Temporal attention models dependencies among coordinate-indexed samples. The learned-query SourcePool combines the resulting source representations while remaining invariant to their order. Source roles distinguish identical sampled functions used for different purposes in the testbench. Sources may have different physical supports; token embeddings, masks, and pooling are specified in Section B.2.

4.3 MULTISCALE RESPONSE DECODING

Each analysis has a complete DeepONet head parameterized by ψ_a, with independent branches, trunks, and affine readout. It factorizes prediction into instance-dependent coefficients and coordinate-dependent bases, with a base path and a Fourier-residual path for variation at different scales. For each real output component c, the branches produce coefficients $b_c = \mathcal{B}_c(h_a)$ and $b_c^F = \mathcal{B}_{F,c}(h_a)$, while trunks evaluate the query features associated with physical coordinate q:

$$\begin{aligned} z_c(q) &= \langle b_c, t(q) \rangle, & z_c^F(q) &= \langle b_c^F, t_F(\gamma(q)) \rangle, \\ \widehat{\mathbf{y}}_a(q) &= A_a[z(q); z^F(q)] + \beta_a. \end{aligned} \tag{5}$$

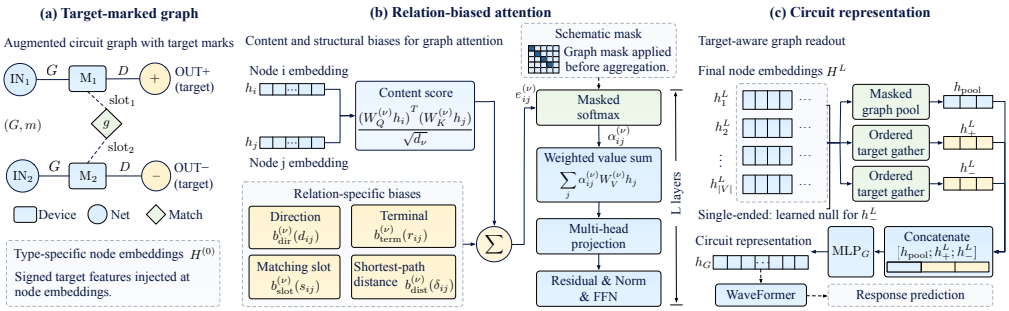


Figure 3: Circuit encoder within the representation backbone. (a) Device, net, and matching nodes exchange messages in both directions. (b) Direction, terminal, matching-slot, and shortest-path biases modify the attention logits before masked softmax and value aggregation. (c) Global pooling and ordered positive/negative target embeddings form $h_{G,a}$ through MLP_G.

Here γ is a fixed Fourier embedding, and z, z^F collect one component for DC/transient or two real/imaginary components for AC. The base trunk learns a coordinate basis, and the residual trunk uses multiscale Fourier features. An affine readout combines their component vectors; the two paths and readout are trained jointly. For each instance and analysis, encodings and branch coefficients are computed once and reused across queries.

4.4 Masked Value and Shape Supervision

We combine value, derivative, and curve-feature supervision to capture both response levels and brief transitions:

$$\mathcal{L}^a_{\text{response}} = \lambda_{\text{fit},a}\mathcal{L}^a_{\text{fit}} + \lambda_{\text{slope},a}\mathcal{L}^a_{\text{slope}} + \lambda_{\text{curv},a}\mathcal{L}^a_{\text{curv}} + \lambda_{\text{feat},a}\mathcal{L}^a_{\text{feat}}. \tag{6}$$

For training-normalized targets $\bar{Y}_{a,i}$ and predictions $\hat{\bar{Y}}_{a,i}$, define $r_1 = \text{slope}$ and $r_2 = \text{curv}$:

$$\ell^a_{\text{fit},i} = \left\langle \mathcal{H}\Big(\hat{\bar{Y}}_{a,i} - \bar{Y}_{a,i}\Big)\right\rangle_{M^{(0)}_{a,i}}, \tag{7}$$

$$\ell^a_{r_k,i} = \left\langle \mathcal{H}\Big(\operatorname{asinh}(D_k\hat{\bar{Y}}_{a,i}) - \operatorname{asinh}(D_k\bar{Y}_{a,i})\Big)\right\rangle_{M^{(k)}_{a,i}}, \quad k = 1, 2. \tag{8}$$

Here $\mathcal{H}$ is Smooth-L1 ($\delta_H = 0.5$); $\langle\cdot\rangle_M$ averages valid entries and D_k takes kth-order divided differences on normalized coordinates. Masks exclude invalid targets and undefined differences; asinh compresses derivative magnitudes while preserving signs. The feature error matches endpoints, signed net change, maximum absolute slope, and a curvature-weighted transition centroid. Each term averages $\log(1+\ell^a_{r,i})$ over valid curves. Section B.4 specifies implementation details and coefficients; Section 5.2 evaluates loss components. Each step jointly updates the shared backbone and three heads with $\mathcal{L}_{\text{pretrain}} = \frac{1}{3}\sum_{a\in\mathcal{A}}\mathcal{L}^a_{\text{response}}$.

5 Experiments

We evaluate three capabilities of WaveFormer: response prediction on held-out configurations (Section 5.2), transfer of response-pretrained representations to scalar performance prediction (Section 5.3), and guidance for circuit sizing with SPICE verification (Section 5.4).

5.1 Experimental Setup

Response pretraining. We jointly pretrain WaveFormer on all three analyses and select one checkpoint by their lowest mean validation normalized waveform MAE. All response predictors use the same parameter-instance split, target information, source records, 128-query responses, and optimization budget. Normalization uses training data only; topology and instance identifiers are not model inputs. Validity masks exclude failed or padded response entries. The held-out configurations belong to the benchmark's 30 known topologies.

Table 2: Topology-macro response prediction on the fixed test split.

Metric	DeepONet	PGNN	KCLNet	Ckt2Vec	LSTM	WAVEFORMER
(a) Response accuracy and shape fidelity (symmetric L_2, Pearson ρ)						
DC symmetric $L_2 \downarrow$	$0.189_{\pm 0.002}$	$0.289_{\pm 0.023}$	$0.439_{\pm 0.018}$	$0.278_{\pm 0.012}$	$0.414_{\pm 0.059}$	$\mathbf{0.118_{\pm 0.000}}$
DC Pearson $\rho \uparrow$	$0.864_{\pm 0.004}$	$0.446_{\pm 0.024}$	$0.369_{\pm 0.065}$	$0.308_{\pm 0.067}$	$0.209_{\pm 0.024}$	$\mathbf{0.928_{\pm 0.001}}$
AC symmetric $L_2 \downarrow$	$0.463_{\pm 0.004}$	$1.209_{\pm 0.026}$	$1.448_{\pm 0.003}$	$1.520_{\pm 0.007}$	$1.391_{\pm 0.058}$	$\mathbf{0.203_{\pm 0.008}}$
AC Pearson $\rho \uparrow$	$0.898_{\pm 0.002}$	$0.503_{\pm 0.023}$	$0.509_{\pm 0.018}$	$0.431_{\pm 0.014}$	$0.375_{\pm 0.052}$	$\mathbf{0.968_{\pm 0.002}}$
Tran. symmetric $L_2 \downarrow$	$0.287_{\pm 0.000}$	$1.282_{\pm 0.243}$	$1.163_{\pm 0.110}$	$0.986_{\pm 0.297}$	$1.356_{\pm 0.086}$	$\mathbf{0.192_{\pm 0.004}}$
Tran. Pearson $\rho \uparrow$	$0.645_{\pm 0.003}$	$0.267_{\pm 0.085}$	$0.105_{\pm 0.102}$	$0.355_{\pm 0.186}$	$0.107_{\pm 0.018}$	$\mathbf{0.786_{\pm 0.008}}$
(b) Normalized amplitude errors (NMAE, NRMSE)						
DC NMAE $\downarrow$	$0.239_{\pm 0.009}$	$68.241_{\pm 16.058}$	$171.866_{\pm 68.009}$	$37.582_{\pm 9.607}$	$46.976_{\pm 14.903}$	$\mathbf{0.178_{\pm 0.045}}$
DC NRMSE $\downarrow$	$0.259_{\pm 0.009}$	$82.270_{\pm 20.496}$	$188.179_{\pm 70.736}$	$45.109_{\pm 11.438}$	$198.958_{\pm 83.790}$	$\mathbf{0.192_{\pm 0.045}}$
AC NMAE $\downarrow$	$0.154_{\pm 0.002}$	$142.535_{\pm 54.742}$	$468.879_{\pm 79.660}$	$223.179_{\pm 76.787}$	$234.641_{\pm 145.955}$	$\mathbf{0.065_{\pm 0.002}}$
AC NRMSE $\downarrow$	$0.256_{\pm 0.003}$	$225.382_{\pm 110.601}$	$764.128_{\pm 195.749}$	$335.437_{\pm 106.508}$	$1566.132_{\pm 954.948}$	$\mathbf{0.137_{\pm 0.004}}$
Tran. NMAE $\downarrow$	$0.177_{\pm 0.001}$	$38.459_{\pm 19.733}$	$33.122_{\pm 11.941}$	$23.991_{\pm 28.926}$	$7.726_{\pm 1.491}$	$\mathbf{0.083_{\pm 0.003}}$
Tran. NRMSE $\downarrow$	$0.223_{\pm 0.001}$	$42.685_{\pm 22.122}$	$36.289_{\pm 12.682}$	$27.433_{\pm 31.229}$	$41.381_{\pm 8.007}$	$\mathbf{0.123_{\pm 0.004}}$

Notes: Values are means with $\pm$ sample standard deviations in subscripts; bold marks the best mean in each row.

Baselines. We compare against DeepONet, LSTM, and PGNN/KCLNet/Ckt2Vec encoders adapted with a common query-conditioned decoder (Lu et al., 2021; Hu et al., 2025; Hakhamaneshi et al., 2023; Xu et al., 2026b;a). These adaptations extend the graph encoders' original output protocols to the same response task. We compare operating-point predictions with PGNN-DC and direct metric predictions with INSIGHT and ZeroSim (Hakhamaneshi et al., 2023; Poddar et al., 2025; Yang et al., 2025). Sizing baselines include AutoSizer, FD-MAGRPO, FALCON, and CPN (Yu et al., 2026; Jiang et al., 2026; Mehradfar et al., 2025; Liu et al., 2026).

Metrics and execution. We report symmetric L_2, Pearson correlation ρ, normalized mean absolute error (NMAE), and normalized root mean squared error (NRMSE). Topology-macro scores weight topologies equally, and AC metrics combine valid real and imaginary entries. Means and sample standard deviations summarize repeated training runs; the pretraining comparison reports the available point estimates. Sizing uses normalized hypervolume (HV) and IGD+ against the union of verified fronts, with 20 SPICE calls per method during optimization. GPU inference and CPU simulation are timed under isolated execution using NVIDIA A100 (80 GB) GPUs and Intel Xeon Platinum 8368 CPUs. Section C specifies hardware, baseline adaptations, sizing objectives, and metric definitions.

5.2 RESPONSE CURVE PREDICTION

The first task evaluates the pretrained response model directly on held-out circuit configurations. We measure the accuracy and shape fidelity of the predicted curves, examine the pretraining components, and compare inference cost with circuit simulation.

Response prediction. Table 2 shows that WAVEFORMER achieves the best mean on all 12 response metrics, with DeepONet consistently second. Relative to DeepONet, NRMSE decreases by 25.9%, 46.5%, and 44.8% for DC, AC, and transient responses, respectively. The largest relative NRMSE reduction occurs on AC, while transient responses show the largest absolute correlation gain, from 0.645 to 0.786. Lower NMAE and symmetric L_2 accompany the correlation gains, indicating that the model better captures both response amplitudes and their variation across physical coordinates. Section C.2 provides the metric definitions and the corresponding error–correlation plots.

Component ablations. We evaluate four component substitutions under the experimental setup in Section 5.1. w.o. circuit attention uses a message-passing neural network (MPNN) retaining graph relations and target readout; w.o. testbench attention uses a pointwise multilayer perceptron (MLP) and mean pooling while retaining source values, roles, and physical coordinates. w.o. multiscale decoder uses an MLP receiving the same representations, query coordinates, and fixed Fourier features. w.o. shape supervision retains value fitting alone, removing slope, curvature, and feature

Table 3: DC and transient component comparisons.

Metric	WAVEFORMER (Full)	w.o. Circuit attention	w.o. Testbench attention	w.o. Multiscale decoder	w.o. Shape supervision
(a) Response accuracy and shape fidelity (symmetric L_2, Pearson ρ)					
DC symmetric $L_2 \downarrow$	**0.118**	0.126	0.132	0.130	0.128
DC Pearson $\rho \uparrow$	**0.928**	0.924	0.921	0.925	0.875
Tran. symmetric $L_2 \downarrow$	**0.192**	0.273	0.256	0.255	0.196
Tran. Pearson $\rho \uparrow$	**0.786**	0.655	0.695	0.637	0.785
(b) Normalized amplitude errors (NMAE, NRMSE)					
DC NMAE $\downarrow$	**0.178**	0.283	0.211	0.214	0.255
DC NRMSE $\downarrow$	**0.192**	0.298	0.226	0.229	0.272
Tran. NMAE $\downarrow$	**0.083**	0.146	0.137	0.126	0.086
Tran. NRMSE $\downarrow$	**0.123**	0.196	0.180	0.174	0.126

Note: Bold marks the best reported value; w.o. denotes the component substitution defined in the text.

Table 4: Inference latency and memory on the evaluated CPU/GPU systems.

Metric	HSPICE	Spectre	APS	ngspice	WAVEFORMER (Ours)
Mean minimum runtime [ms] $\downarrow$	347.6	1187.5	2122.0	2179.3	**8.682**
CPU peak RSS [MB]	54.7	98.0	97.9	189.6	–
GPU peak tensor allocation [MB]	–	–	–	–	19.3
GPU process memory [MB]	–	–	–	–	557.8

Notes. Runtime averages per-case minima; bold marks the lowest reported runtime. RSS denotes resident set size. CPU RSS, GPU tensor allocation, and GPU process memory use separate accounting; – denotes an inapplicable.

supervision. Table 3 reports the results; each substitution changes one component while retaining the task inputs.

The component effects differ by analysis. Removing shape supervision causes the largest DC correlation drop, from 0.928 to 0.875. On transient responses, architecture substitutions produce larger error increases than removing shape supervision, and the full configuration has the lowest reported errors.

Inference efficiency. Table 4 reports a mean per-case minimum inference time of 8.682 ms for WAVEFORMER, giving simulator-to-model runtime ratios of 40.0–$251.0\times$. These measurements compare GPU forward passes with resident weights and inputs against CPU execution of the circuit analyses. GPU tensor allocation and process memory are reported separately from simulator CPU resident memory.

5.3 PERFORMANCE PREDICTION

Evaluation setting. The second task fine-tunes the shared circuit–testbench backbone with a joint metric head for scalar prediction and sizing (Section 5.4). We separately evaluate DC operating points extracted from predicted curves. Macro errors weight topologies equally, pooled errors combine all paired cases, and paired wins measure how often absolute error is lower. For direct prediction, we compute RMSE for each eligible topology–metric cell and normalize it by a common global scale fitted on training data. Primary NRMSE averages cell errors within each topology and then across topologies; DC/AC/transient denote metric categories. Sections C.5 and C.7 detail the evaluation and training protocols.

Effect of response pretraining. Figure 4(d) compares response-pretrained and randomly initialized shared backbones within the same performance-predictor architecture and fine-tuning protocol. Pretraining reduces Primary NRMSE by 4.7%, with reductions of 8.4% and 8.2% for DC and AC, respectively. Transient NRMSE increases from 0.035 to 0.036, showing that the aggregate benefit depends on the metric category. This comparison isolates the initialization choice under a fixed fine-tuning budget; the pretrained model additionally uses response supervision, as detailed in Section C.7. This supports response supervision as a useful but task-dependent starting point for scalar prediction.

Table 5: Performance prediction: response-derived DC operating points and direct circuit-metric prediction.

(a) DC operating-point accuracy			(b) Direct circuit-metric prediction			
Metric	PGNN-DC	WaveFormer	Metric	INSIGHT	ZeroSim	WaveFormer
Macro MAE [mV] ↓	$1.782_{\pm 0.888}$	$\mathbf{0.480}_{\pm \mathbf{0.201}}$	Primary NRMSE ↓	$0.058_{\pm 0.003}$	$0.180_{\pm 0.000}$	$\mathbf{0.028}_{\pm \mathbf{0.009}}$
Macro RMSE [mV] ↓	$1.971_{\pm 0.895}$	$\mathbf{0.584}_{\pm \mathbf{0.248}}$	Median NRMSE ↓	$\mathbf{0.003}_{\pm \mathbf{0.000}}$	$0.137_{\pm 0.000}$	$0.008_{\pm 0.001}$
Pooled MAE [mV] ↓	$1.497_{\pm 0.990}$	$\mathbf{0.392}_{\pm \mathbf{0.151}}$	P90 NRMSE ↓	$0.217_{\pm 0.010}$	$0.487_{\pm 0.000}$	$\mathbf{0.069}_{\pm \mathbf{0.043}}$
Pooled RMSE [mV] ↓	$2.442_{\pm 1.106}$	$\mathbf{0.793}_{\pm \mathbf{0.338}}$	P95 NRMSE ↓	$0.352_{\pm 0.012}$	$0.585_{\pm 0.000}$	$\mathbf{0.096}_{\pm \mathbf{0.084}}$
Median AE [mV] ↓	$0.928_{\pm 0.693}$	$\mathbf{0.097}_{\pm \mathbf{0.041}}$	P99 NRMSE ↓	$0.497_{\pm 0.014}$	$0.735_{\pm 0.000}$	$\mathbf{0.347}_{\pm \mathbf{0.044}}$
P90 AE [mV] ↓	$3.438_{\pm 2.429}$	$\mathbf{1.080}_{\pm \mathbf{0.406}}$	DC NRMSE ↓	$0.046_{\pm 0.004}$	$0.167_{\pm 0.000}$	$\mathbf{0.023}_{\pm \mathbf{0.005}}$
Pooled R^2 ↑	$0.997_{\pm 0.003}$	$\mathbf{1.000}_{\pm \mathbf{0.000}}$	AC NRMSE ↓	$0.061_{\pm 0.009}$	$0.221_{\pm 0.000}$	$\mathbf{0.027}_{\pm \mathbf{0.009}}$
Accuracy@5 mV [%] ↑	$95.025_{\pm 5.775}$	$\mathbf{99.602}_{\pm \mathbf{0.456}}$	Tran. NRMSE ↓	$0.060_{\pm 0.004}$	$0.162_{\pm 0.000}$	$\mathbf{0.036}_{\pm \mathbf{0.009}}$
Paired wins [%] ↑	$14.627_{\pm 10.473}$	$\mathbf{85.373}_{\pm \mathbf{10.473}}$	Failure cells ↓	$1_{\pm 2}$	$359_{\pm 0}$	$\mathbf{0}_{\pm \mathbf{0}}$

Notes. Values are means with ± sample standard deviations in subscripts; bold marks the best mean for each statistic.

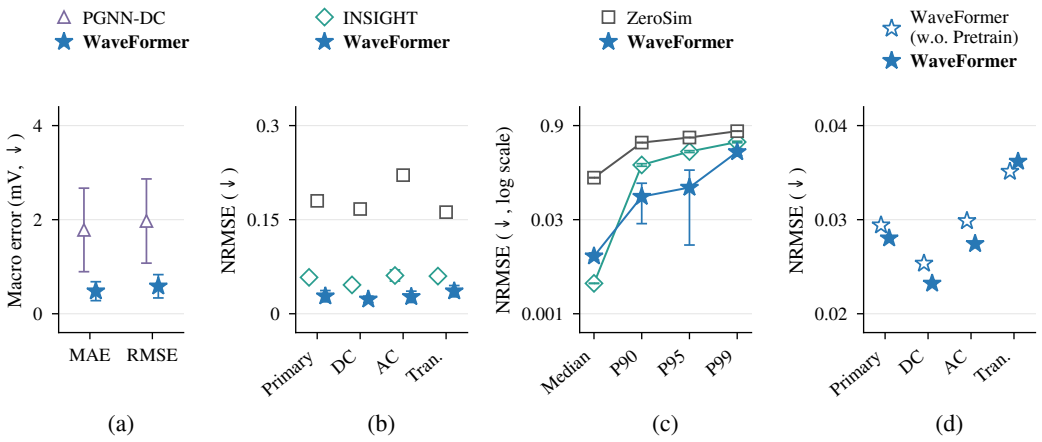


Figure 4: Performance prediction: (a) topology-macro errors of response-derived DC operating points; (b) Primary and analysis-wise NRMSE of direct-metric models; (c) error quantiles across eligible topology–metric cells, connected on a log axis; (d) direct-metric prediction with and without response pretraining. Error bars in (a)–(c) show one sample standard deviation across training runs.

Response-derived measurements. As shown in Figure 4(a), operating points extracted from predicted DC responses reduce topology-macro MAE by 73.1% relative to PGNN-DC, reaching 0.480 mV. The paired-win rate is 85.373%, and pooled RMSE falls by 67.5%. Table 5 shows improvements in all nine operating-point statistics, including error quantiles and accuracy within 5 mV. The agreement between macro and pooled errors shows that the improvement holds under both equal topology weighting and aggregation over paired cases.

Direct performance prediction. WaveFormer's direct-metric predictor reduces Primary NRMSE by 51.7% relative to INSIGHT and 84.4% relative to ZeroSim, as shown in Figure 4(b). The quantiles in Figure 4(c) reveal different error profiles across eligible topology–metric cells. INSIGHT has the lower median, 0.003 versus WaveFormer's 0.008, while WaveFormer reduces P90 and P95 errors by 68.2% and 72.7%, respectively, relative to INSIGHT. The reduced upper-tail errors are relevant to evaluating multiple specifications together, where large individual errors can limit a predictor's usefulness.

5.4 Design Optimization

Task and budget. The third task uses the performance predictor from Section 5.3 to guide circuit sizing. An ensemble of these predictors estimates candidate specifications during the search, and SPICE verifies the resulting designs. On `Basic_LDO`, we optimize five coupled device-group variables to minimize area, DC line variation, and transient excursion while maximizing AC gain

Table 6: Fixed-budget multi-objective sizing on `Basic_LDO`.

Metric	AutoSizer	FD-MAGRPO	FALCON	CPN	WAVEFORMER (Ours)
Normalized HV [10^{-3}] ↑	**21.717**	18.515	13.139	11.771	19.411
IGD+ [10^{-3}] ↓	57.465	64.614	57.907	140.991	**42.335**
Feasible rate [%] ↑	**30**	15	25	25	**30**
Pareto points ↑	**6**	3	3	5	5

Notes. Bold marks the best value in each row.

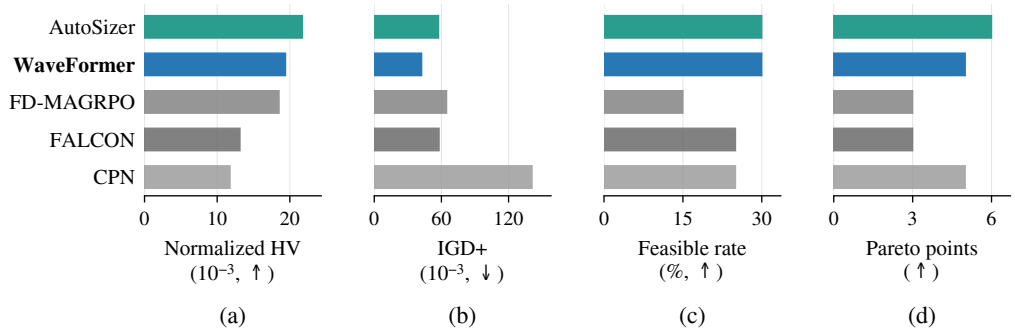


Figure 5: SPICE-verified `Basic_LDO` sizing with an optimization budget of 20 SPICE calls per method: (a) normalized HV; (b) IGD+; (c) feasible-query rate; (d) verified Pareto-set size.

under DC/AC/transient feasibility constraints. Each method receives 20 SPICE calls during optimization. Normalized HV and IGD+ against the union of verified fronts assess Pareto-front quality. Feasible-query rate and verified Pareto-set size measure how the evaluation budget translates into usable designs.

Search and verification. WAVEFORMER's cross-entropy search ranks candidates by feasibility, nondominated rank, and crowding distance. SPICE residuals update an online calibration model; every reported Pareto design is independently re-simulated after optimization.

Verified sizing results. Figure 5 shows that WAVEFORMER produces five verified Pareto designs and ranks second in normalized HV and first in IGD+, ahead of FD-MAGRPO, FALCON, and CPN on both quality measures. The comparison with AutoSizer depends on the metric: WAVEFORMER has lower HV but better IGD+, with the same 30% feasible-query rate and one fewer verified Pareto design. Independent SPICE verification ties the reported trade-offs to simulated circuit behavior under a limited evaluation budget. This tests the predictor's practical use beyond held-out regression accuracy, while the differing HV and IGD+ rankings highlight complementary aspects of Pareto-set quality. Section C.6 specifies the search space, constraints, optimization procedure, and independent verification of the returned designs.

6 CONCLUSION

We presented WAVEFORMER and WAVEFORMER BENCHMARK to study response-supervised circuit representations and their transfer to performance prediction. Conditioning on circuit structure, stimulus, and observation targets connects DC, complex AC, and transient curve learning to a common modeling interface. On held-out configurations of known topologies, response NRMSE decreases by 25.9–46.5% relative to DeepONet. Backbone transfer improves aggregate performance prediction under the evaluated fine-tuning protocol, with gains in DC and AC metrics but a small degradation in transient metrics. SPICE-verified `Basic_LDO` sizing demonstrates a design use of the resulting predictor. Together, these results support using response curves to connect behavioral modeling, transferable representations, and verified design decisions within one evaluation pipeline. The transient result also shows why transfer should be checked for each analysis. These findings motivate further study of when response supervision supports downstream tasks, including transfer to unseen topologies and broader design settings.

AI USE STATEMENT

The authors led the research, including the problem formulation, method design, experimental decisions, and assessment of the findings. Generative AI tools assisted with literature organization, experimental planning and result interpretation, code development, manuscript drafting and translation, and figure preparation. All AI-assisted content and outputs were reviewed and verified by the authors, who take full responsibility for the final text, methods, claims, and experimental results. The LLM configuration used in the AutoSizer baseline is reported in Section C.1.

REPRODUCIBILITY STATEMENT

Dataset construction and configuration-level splits are described in Section A. The architecture, normalization, and loss definitions are specified in Sections B and 4; pretraining and fine-tuning settings and model-selection criteria are provided in Section C.7. Section C documents baseline adaptations, hardware, evaluation metrics, and the SPICE-verified sizing protocol.

ETHICS STATEMENT

This work uses circuit graphs, parameterized netlists, and engineering data generated through SPICE simulation. It does not involve human subjects or personal data. Circuit designs returned by the optimization procedure are independently verified with SPICE.

REFERENCES

Cadence Design Systems, Inc. *Virtuoso Spectre Circuit Simulator and Accelerated Parallel Simulator User Guide*, October 2015. Product Version 15.1.

Zehao Dong, Weidong Cao, Muhan Zhang, Dacheng Tao, Yixin Chen, and Xuan Zhang. CktGNN: Circuit graph neural network for electronic design automation. In *Proc. ICLR*, 2023.

Paul R. Gray, Paul J. Hurst, Stephen H. Lewis, and Robert G. Meyer. *Analysis and Design of Analog Integrated Circuits*. John Wiley & Sons, 5 edition, 2009.

Kourosh Hakhamaneshi, Marcel Nassar, Mariano Phielipp, Pieter Abbeel, and Vladimir Stojanović. Pretraining graph neural networks for few-shot analog circuit modeling and design. *IEEE TCAD*, 42(7):2163–2173, 2023.

Yupeng Hu, Yapeng Li, Jindong Tu, Tinghuan Chen, Qi Sun, and Cheng Zhuo. Hierarchical behavioral learning-based dynamic electromigration analysis for signal networks. *IEEE TCAD*, 44(11): 4433–4437, 2025.

Haoning Jiang, Han Wu, Zhuoli Ouyang, Ziheng Wang, Tinghuan Chen, and Junmin Jiang. FD-MAGRPO: Functionality-driven multi-agent group relative policy optimization for analog-ldo sizing. *Proc. AAAI*, 40(27):22310–22317, 2026.

Savvas Karipidis, Valentina Davidoiu, Andi Buzo, Georg Pelz, and Thomas Noulis. Generative deep learning surrogate framework for continuous time-domain waveform synthesis of analog circuit transient response, August 2026. SSRN preprint 7352662, posted August 26, 2026.

Ken Kundert and Henry Chang. Verification of complex analog integrated circuits. In *Proc. CICC*, pp. 177–184. IEEE, 2006.

Jintao Li, Haochang Zhi, Ruiyu Lyu, Wangzhen Li, Zhaori Bi, Keren Zhu, Yanhan Zeng, Weiwei Shan, Changhao Yan, Fan Yang, Yun Li, and Xuan Zeng. AnalogGym: An open and practical testing suite for analog circuit synthesis. In *Proc. ICCAD*, 2024.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *Proc. ICLR*, 2021.

Zhuohua Liu, Kaiqi Huang, Qinxin Mei, Yuanqi Hu, and Wei W. Xing. Exploiting function-family structure in analog circuit optimization. In *Proc. DAC*. ACM, 2026.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3:218–229, 2021.

Asal Mehradfar, Xuzhe Zhao, Yilun Huang, Emir Ceyani, Yankai Yang, Shihao Han, Hamidreza Aghasi, and Salman Avestimehr. FALCON: An ML Framework for Fully Automated Layout-Constrained Analog Circuit Design. In *Proc. NeurIPS*, 2025.

Laurence W. Nagel. SPICE2: A computer program to simulate semiconductor circuits. Technical Report ERL-M520, University of California, Berkeley, 1975.

Souradip Poddar, Youngmin Oh, Yao Lai, Hanqing Zhu, Bosun Hwang, and David Z. Pan. INSIGHT: A universal neural simulator framework for analog circuits with autoregressive transformers. In *Proc. DAC*, 2025.

Synopsys, Inc. *HSPICE User Guide: Basic Simulation and Analysis*, June 2016. Version L-2016.06.

Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Proc. NeurIPS*, volume 33, pp. 7537–7547, 2020.

The ngspice Team. *ngspice: A Circuit Simulator Based on SPICE3f5, User's Manual*, 2023. URL `https://ngspice.sourceforge.io/docs/ngspice-manual.pdf`. Version 42.

Peng Xu, Jintao Li, Tsung-Yi Ho, Bei Yu, and Keren Zhu. Performance-driven analog layout automation: Current status and future directions. In *Proc. ASPDAC*, pp. 679–685, 2024.

Peng Xu, Yapeng Li, Tinghuan Chen, Tsung-Yi Ho, and Bei Yu. Ckt2Vec: Efficient electrical encoding for analog circuit representations in vector space. IEEE TCAD manuscript, 2026a.

Peng Xu, Yapeng Li, Tinghuan Chen, Tsung-Yi Ho, and Bei Yu. KCLNet: Electrically equivalence-oriented graph representation learning for analog circuits. In *Proc. AAAI*, volume 40, pp. 1355–1363, 2026b.

Xiaomeng Yang, Jian Gao, Yanzhi Wang, and Xuan Zhang. ZeroSim: Zero-shot analog circuit evaluation with unified transformer embeddings. In *Proc. ICCAD*, 2025.

Enis Yazici. A neural surrogate for accelerated multi-topology analog circuit simulation. *Results in Engineering*, 30:111112, 2026.

Xi Yu, Dmitrii Torbunov, Soumyajit Mandal, and Yihui Ren. AutoSizer: Automatic sizing of analog and mixed-signal circuits via large language model (LLM) agents. In *Proc. ICML*, volume 306 of *Proceedings of Machine Learning Research*, 2026.

Jinling Zhang, Shaowei Chen, Yuhan Chen, Yigong Zhang, and Shuai Zhao. Dynamic behavior modeling of analog circuits based on deep operator network. In *Proc. ONCON*, 2023.

A Benchmark Details

A.1 Dataset Construction

We construct the dataset by sampling parameter configurations from AnalogGym's netlists and declared parameter spaces, then running the supported SPICE analyses with its testbenches. Each topology–analysis pair defines one registered target. The training, validation, and test partitions contain 23,568, 5,037, and 5,090 parameter configurations, respectively. The partitions are disjoint by configuration, with all 30 topologies represented in each partition. Table 7 summarizes the dataset.

Table 7: Benchmark statistics and construction choices.

Field	Value
Construction source	AnalogGym netlists, declared parameter spaces, and testbenches
Circuit categories / topologies	6 / 30
Parameter instances	33,695
Train / validation / test	23,568 / 5,037 / 5,090
Analysis types	DC, AC, transient
Topology–analysis targets	90
Target convention	88 single-ended; 2 PLL differential
Processed response length	128 queries
Native trajectories	Simulator coordinates and values retained before processing
Source sample / source role / query	10 / 6 / 6 dimensions
Output channels (DC, AC, transient)	1, 2, 1

A.2 Data Format

Let $\mathbf{Q}_a$ collect the six-dimensional query features associated with the physical coordinates in Q_a. Each example is stored as

$$\left(G, U_a, \mathbf{Q}_a, Y_a, a, m, M_a^{(0)}, \mathrm{id}_{\mathrm{topology}}, \mathrm{id}_{\mathrm{instance}}\right). \tag{9}$$

For analysis a, let $o_a = 1$ for the real-valued DC and transient outputs and $o_a = 2$ for the real and imaginary components of the complex AC output. The channel vector $\mathbf{y}_a(q) \in \mathbb{R}^{o_a}$ contains the single component $y_a(q)$ for DC/transient and $[\mathrm{Re}\, y_a(q), \mathrm{Im}\, y_a(q)]^\top$ for AC. Row t of Y_a is $\mathbf{y}_a(q_t)^\top$. For T response queries, the query, response, and validity tensors have shapes

$$\mathbf{Q}_a \in \mathbb{R}^{T\times 6}, \qquad Y_a \in \mathbb{R}^{T\times o_a}, \qquad M_a^{(0)} \in \{0,1\}^{T\times o_a}. \tag{10}$$

Table 8 describes the fields. Variable-length axes are padded when assembling a batch, with masks identifying valid entries.

For a training batch of size B, normalized predictions, targets, and validity masks have shapes

$$\widehat{\bar{Y}}_a, \bar{Y}_a \in \mathbb{R}^{B\times T\times o_a}, \qquad M_a^{(0)} \in \{0,1\}^{B\times T\times o_a}. \tag{11}$$

The released schema records the parser order of source-sample, source-role, and query channels.

Graph attributes. Device nodes carry a categorical device type and the applicable numerical fields, such as transistor geometry and multiplicity or a passive value. Net nodes include structural/testbench roles and the target sign. Matching nodes have a group type; matching incidences carry the slot label. Physical device–net incidences carry the electrical terminal and both message directions.

Raw and processed responses. Raw records store native simulator coordinates and values, simulator status, and target registration. Processed records store 128 response queries, validity masks, and preprocessing metadata linking each record to its raw trace. Split manifests are defined at the parameter-instance level and are shared by all three analyses.

Table 8: Benchmark data format. S is the number of input sources, K is the padded source length, T is the number of response queries, and o_a is the number of real-valued output components.

Field	Shape / type	Semantics
G	attributed graph	Device, net, and matching nodes; directed terminal/matching incidences; numerical device parameters; injected target roles.
U_a.sample	$S \times K \times 10$	Ten numeric fields per sampled source record, in the fixed parser order; source padding is masked.
U_a.role	$S \times 6$	Source-role descriptor identifying each source's semantic function in the testbench.
U_a.mask	$S \times K$	Valid source samples.
$\mathbf{Q}_a$	$T \times 6$	Six-dimensional response-query features for the selected analysis.
Y_a	$T \times o_a$	Real DC/transient values or AC real–imaginary pairs.
a	categorical	One of DC, AC, or transient.
m	node role(s)	One marked net, or an ordered positive/negative net pair for a differential target.
$M_a^{(0)}$	$T \times o_a$	Valid response entries, including failure and padding masking.
IDs	two integers/strings	Topology and parameter-instance identifiers used to group records.

B Architecture and Training Details

This section details the shared encoders, analysis-specific response heads, and joint training objective in Section 4. The circuit and testbench encoders form a shared backbone with parameters η in Equation (2). Each analysis has a complete DeepONet head with parameters ψ_a, comprising its own base and Fourier-residual branches, trunks, and affine output map. The pretraining model has approximately 1.76 million parameters. The equations omit the analysis index where unambiguous.

B.1 Target-Aware Circuit Encoding

Device nodes carry device classes and numerical simulation parameters; net nodes carry connectivity and testbench roles; matching nodes connect each declared matching group. Physical and matching incidences carry messages in both directions while preserving direction and terminal labels. Signed node features in Figure 3 identify positive and negative targets for differential measurements.

Let $H^{(0)}$ be type-specific projections of the device, net, and matching payloads. The matrix H^ℓ collects the node embeddings h_i^ℓ, and L is the number of encoder layers. For layer ℓ and head ν, relation-aware attention is

$$e_{ij}^{\ell,\nu} = \frac{(W_Q^{\ell,\nu} h_i^\ell)^\top (W_K^{\ell,\nu} h_j^\ell)}{\sqrt{d_\nu}} + b_{\mathrm{dir}}^{\ell,\nu}(d_{ij}) + b_{\mathrm{term}}^{\ell,\nu}(r_{ij}) + b_{\mathrm{slot}}^{\ell,\nu}(s_{ij}) + b_{\mathrm{dist}}^{\ell,\nu}(\delta_{ij}), \tag{12}$$

$$\alpha_{ij}^{\ell,\nu} = \frac{M_{ij}^G \exp(e_{ij}^{\ell,\nu})}{\sum_k M_{ik}^G \exp(e_{ik}^{\ell,\nu})}, \tag{13}$$

$$\widetilde{h}_i^{\ell,\nu} = \sum_j \alpha_{ij}^{\ell,\nu} W_V^{\ell,\nu} h_j^\ell. \tag{14}$$

Here M_{ij}^G is the binary graph-attention mask; the softmax is evaluated only for valid query nodes with at least one allowed key. The attention output passes through a multi-head projection, residual connection, normalization, and feed-forward block. Null terminal and slot labels are learned explicitly for relations to which those fields do not apply. After the final layer, if h_+^L and h_-^L are the marked target embeddings (with a learned null for h_-^L on a single-ended target), let $h_{\mathrm{pool}} = \mathrm{MaskedPool}\{h_i^L\}_{i \in V}$. The circuit representation is

$$h_G = \mathrm{MLP}_G\left[h_{\mathrm{pool}}; h_+^L; h_-^L\right]. \tag{15}$$

B.2 Function-Aware Testbench Encoding

A testbench can contain multiple independent sources, and each source may be sampled on its own physical support. For source s, let $U_s = (u_{s,1}, \ldots, u_{s,K_s})$ be its valid sample records, $\xi_{s,k}$ the physical coordinate of sample k, and ζ_s a descriptor of the source's semantic role. Each sample is embedded as

$$h_{s,k}^{U,(0)} = W_u u_{s,k} + W_\zeta \zeta_s + e_U(\xi_{s,k}), \tag{16}$$

where W_u and W_ζ project the numerical sample and source-role descriptor, respectively, and $e_U(\xi_{s,k})$ embeds its coordinate. A shared temporal Transformer processes the valid sequence of each source and pools it into a source representation h_s^U. The testbench representation is then

$$h_U = \mathrm{SourcePool}\big(\{h_s^U\}_{s=1}^S\big)\,. \tag{17}$$

The permutation-invariant SourcePool makes h_U independent of source order. The role descriptor ζ_s distinguishes sources with identical samples but different functions in the testbench.

Equation 16 is applied independently to each declared source. Let H_s^U be the matrix of sample embeddings for source s, and let M_s^U be its sample mask from $U_a.\mathrm{mask}$. One temporal layer and the subsequent pooling give

$$\widetilde{H}_s^U = \mathrm{LN}\big(H_s^U + \mathrm{MHA}(H_s^U, H_s^U, H_s^U; M_s^U)\big)\,, \tag{18}$$

$$(H_s^U)' = \mathrm{LN}\Big(\widetilde{H}_s^U + \mathrm{FFN}(\widetilde{H}_s^U)\Big)\,, \tag{19}$$

$$h_s^U = \frac{\Big(\sum_k M_{s,k}^U (H_{s,k}^U)'\Big)^\top}{\sum_k M_{s,k}^U + \epsilon_U}. \tag{20}$$

Here $(H_{s,k}^U)'$ is row k of the output matrix, and $\epsilon_U > 0$ stabilizes the masked mean. The same temporal weights are used across sources and analyses. LN, MHA, and FFN denote layer normalization, multi-head attention, and a feed-forward network. Source pooling applies attention with a learned query vector and a source-level validity mask.

B.3 Multiscale Response Decoding

The response interface maps the representation and finite physical query list $Q_a = (q_1, \ldots, q_T)$ to the registered observable:

$$\widehat{Y}_a = H_{\psi_a}^{\mathrm{response}}(h_a, Q_a). \tag{21}$$

Writing $h = h_a$, for $o_a = \dim_{\mathbb{R}} \mathcal{Y}_a$ real-valued output components the branch coefficients are

$$\mathcal{B}(h) = [b_1, \ldots, b_{o_a}], \qquad \mathcal{B}_F(h) = [b_1^F, \ldots, b_{o_a}^F], \qquad b_c \in \mathbb{R}^p, \quad b_c^F \in \mathbb{R}^{p_F}. \tag{22}$$

The base trunk is $t(q) \in \mathbb{R}^p$, and the residual trunk is $t_F(\gamma(q)) \in \mathbb{R}^{p_F}$. Here $t(q)$ evaluates the query features associated with physical coordinate q; Section A distinguishes these features from the physical query list. The component-wise inner products and affine fusion are given in Equation (5).

Writing $h = h_a = [h_{G,a}; h_{U,a}]$, the base and residual coefficient matrices are

$$\mathbf{B} = \mathcal{B}(h) \in \mathbb{R}^{p \times o_a}, \qquad \mathbf{B}_F = \mathcal{B}_F(h) \in \mathbb{R}^{p_F \times o_a}. \tag{23}$$

For the physical query list $Q_a = (q_1, \ldots, q_T)$, the trunk matrices stack the query embeddings as

$$\mathbf{T} = \begin{bmatrix} t(q_1)^\top \\ \vdots \\ t(q_T)^\top \end{bmatrix} \in \mathbb{R}^{T \times p}, \qquad \mathbf{T}_F = \begin{bmatrix} t_F(\gamma(q_1))^\top \\ \vdots \\ t_F(\gamma(q_T))^\top \end{bmatrix} \in \mathbb{R}^{T \times p_F}. \tag{24}$$

The two DeepONet products are $\mathbf{Z} = \mathbf{T}\mathbf{B}$ and $\mathbf{Z}_F = \mathbf{T}_F\mathbf{B}_F$, both in $\mathbb{R}^{T \times o_a}$. The final prediction is

$$\widehat{Y}_a = [\mathbf{Z}; \mathbf{Z}_F]_{\mathrm{channel}} A_a^\top + \mathbf{1}_T \beta_a^\top\,, \tag{25}$$

where channel concatenation has size $2o_a$, $A_a \in \mathbb{R}^{o_a \times 2o_a}$, and $\beta_a \in \mathbb{R}^{o_a}$ is the output bias. The vector $\mathbf{1}_T$ contains T ones. For each instance and analysis, the shared circuit and testbench encoders produce h_a from that analysis's target and excitation, and its head computes the two branch outputs once; trunk evaluation and the two matrix products scale with T.

B.4 Masked Value and Shape Supervision

The loss in Equation (6) combines robust value fitting with slope, curvature, and response-feature errors. All terms use valid target entries and training-only normalization.

Training-only robust normalization. Output normalization is fitted separately for each topology, analysis, and real-valued channel. Fixing a topology and analysis, let $\mathcal{V}_j^{\mathrm{train}}$ contain all valid training values of channel j, and let $\mathcal{S}_j^{\mathrm{train}}$ contain the maximum-minus-minimum range of each valid training curve in that channel. With Q_p denoting the p-quantile, define

$$
\begin{aligned}
c_j &= \mathrm{Q}_{0.5}(\mathcal{V}_j^{\mathrm{train}}),\\
s_j &= \max\left\{ \frac{\mathrm{Q}_{0.95}(\mathcal{V}_j^{\mathrm{train}}) - \mathrm{Q}_{0.05}(\mathcal{V}_j^{\mathrm{train}})}{3.29}, \mathrm{Q}_{0.5}(\mathcal{S}_j^{\mathrm{train}}), \epsilon_{s,j} \right\},\\
\bar{x}_j &= \mathcal{N}_j(x_j) = \frac{x_j - c_j}{s_j}.
\end{aligned}
\tag{26}
$$

Including typical per-curve ranges accounts for narrow peaks or transitions whose amplitudes may be underestimated by pooled pointwise quantiles. The floor $\epsilon_{s,j}$ is 1 mV for the voltage targets. Predictions are returned to physical units by $\widehat{x}_j = c_j + s_j \widehat{\bar{x}}_j$. AC real and imaginary channels have separate statistics. Source values use training-set medians and interquartile ranges, separately for their real and imaginary channels. Absolute source and query-coordinate features use analysis-specific training medians and interquartile ranges; relative coordinates remain in $[0, 1]$. Source scaling is shared across instances, preserving differences in stimulus amplitude and offset. All statistics remain fixed for validation and test.

Masked differences on nonuniform coordinates. Within an analysis-specific batch, i, t, and c index instances, queries, and real-valued output components. Write the normalized prediction and target as $\widehat{\bar{Y}}_{a,i,t,c}$ and $\bar{Y}_{a,i,t,c}$. Let $\widetilde{q}_{i,t} \in [0, 1]$ be the local scan coordinate: DC sweep values and transient times are scaled linearly from the first to the last query, while AC uses the logarithm of frequency before this scaling. These coordinates express shape relative to each scan window, with logarithmic frequency allocating comparable coordinate intervals to successive frequency decades. For a normalized response X_i, the divided differences are

$$
(D_1 X_i)_{t,c} = \frac{X_{i,t+1,c} - X_{i,t,c}}{\widetilde{q}_{i,t+1} - \widetilde{q}_{i,t}}, \tag{27}
$$

$$
(D_2 X_i)_{t,c} = \frac{2\big((D_1 X_i)_{t+1,c} - (D_1 X_i)_{t,c}\big)}{\widetilde{q}_{i,t+2} - \widetilde{q}_{i,t}}. \tag{28}
$$

Their validity masks are

$$
M^{(1)}_{a,i,t,c} = M^{(0)}_{a,i,t,c} M^{(0)}_{a,i,t+1,c} \mathbf{1}[\widetilde{q}_{i,t+1} > \widetilde{q}_{i,t}], \tag{29}
$$

$$
M^{(2)}_{a,i,t,c} = M^{(1)}_{a,i,t,c} M^{(1)}_{a,i,t+1,c}. \tag{30}
$$

Differences are evaluated only where the required target entries and coordinate intervals are valid; boundary points are not extrapolated.

Robust value and derivative errors. For scalar residual e, the Smooth-L1 penalty is

$$
\mathcal{H}(e) = \begin{cases} e^2/(2\delta_H), & |e| < \delta_H, \\ |e| - \delta_H/2, & |e| \geq \delta_H, \end{cases} \qquad \delta_H = 0.5. \tag{31}
$$

For a nonnegative error tensor R_i and binary mask M_i, write

$$
\langle R_i \rangle_{M_i} = \frac{\sum_{t,c} M_{i,t,c} R_{i,t,c}}{\max\Big(1, \sum_{t,c} M_{i,t,c}\Big)}. \tag{32}
$$

These operators define the per-curve fitting, slope, and curvature errors in Section 4.4. The derivative targets include the reference curve's sharp transitions and changes in slope, such as the approach to a plateau or the reversal near a transient peak.

Response features. For each normalized curve X_i, the feature map ϕ_a extracts its start and end values, signed endpoint change, maximum absolute slope, and a curvature-weighted transition centroid. The endpoints and their signed difference emphasize boundary levels and net change over the query window; the maximum slope singles out brief, fast-changing segments that contribute few terms to an interval average. For channel c, these features are

$$\begin{aligned} g_{i,c}^{\mathrm{start}}(X_i) &= X_{i,1,c}, & g_{i,c}^{\mathrm{end}}(X_i) &= X_{i,T,c}, \\ g_{i,c}^{\mathrm{change}}(X_i) &= X_{i,T,c} - X_{i,1,c}, & g_{i,c}^{\mathrm{slope}}(X_i) &= \max_t |(D_1 X_i)_{t,c}|. \end{aligned} \tag{33}$$

Here the endpoints and maximum are taken over the valid support. Let $\mathcal{K}_i$ contain the valid second-difference positions, and define

$$\begin{aligned} w_{i,t}^K(X_i) &= \log\left(1 + \frac{1}{o_a}\sum_c |(D_2 X_i)_{t,c}|\right), \\ \pi_{i,t}^K(X_i) &= \frac{\exp(w_{i,t}^K(X_i))}{\sum_{u\in\mathcal{K}_i}\exp(w_{i,u}^K(X_i))}, \qquad g_i^{\mathrm{knee}}(X_i) = \sum_{t\in\mathcal{K}_i} \pi_{i,t}^K(X_i)\widetilde{q}_{i,t+1}. \end{aligned} \tag{34}$$

The weights include a uniform baseline, making g_i^{knee} a coarse coordinate centroid. For a dominant transition, it summarizes transition location; multiple events contribute to a common centroid. The centroid penalty is active when the target has appreciable curvature: $\chi_i = \mathbf{1}[\max_{t\in\mathcal{K}_i} w_{i,t}^K(\bar{Y}_{a,i}) > \log(1.05)]$. For compactness, $\widehat{g}$ and g below denote features of $\widehat{\bar{Y}}_{a,i}$ and $\bar{Y}_{a,i}$, respectively. The four feature-group errors are

$$\begin{aligned} e_i^{\mathrm{end}} &= \frac{1}{2o_a}\sum_c \left[\mathcal{H}(\widehat{g}_{i,c}^{\mathrm{start}} - g_{i,c}^{\mathrm{start}}) + \mathcal{H}(\widehat{g}_{i,c}^{\mathrm{end}} - g_{i,c}^{\mathrm{end}})\right], \\ e_i^{\mathrm{change}} &= \frac{1}{o_a}\sum_c \mathcal{H}(\widehat{g}_{i,c}^{\mathrm{change}} - g_{i,c}^{\mathrm{change}}), \\ e_i^{\mathrm{slope}} &= \frac{1}{o_a}\sum_c \mathcal{H}\Big(\mathrm{asinh}(\widehat{g}_{i,c}^{\mathrm{slope}}) - \mathrm{asinh}(g_{i,c}^{\mathrm{slope}})\Big), \\ e_i^{\mathrm{knee}} &= \mathcal{H}(\widehat{g}_i^{\mathrm{knee}} - g_i^{\mathrm{knee}}), \qquad \ell_{\mathrm{feat},i}^a = \frac{e_i^{\mathrm{end}} + e_i^{\mathrm{change}} + e_i^{\mathrm{slope}} + \chi_i e_i^{\mathrm{knee}}}{3 + \chi_i}. \end{aligned} \tag{35}$$

All features come from the response targets themselves. For AC, channel averages include both real and imaginary components.

Training configurations and optimization. For $r \in \{\mathrm{fit}, \mathrm{slope}, \mathrm{curv}, \mathrm{feat}\}$, let $\mathcal{B}_{a,r}$ contain batch instances with valid support. Each loss term averages valid curves equally:

$$\mathcal{L}_r^a = \frac{1}{\max(1, |\mathcal{B}_{a,r}|)} \sum_{i\in\mathcal{B}_{a,r}} \log(1 + \ell_{r,i}^a). \tag{36}$$

The logarithm reduces the influence of exceptionally large curve errors. The coefficient tuple $(\lambda_{\mathrm{fit},a}, \lambda_{\mathrm{slope},a}, \lambda_{\mathrm{curv},a}, \lambda_{\mathrm{feat},a})$ is $(1, 0.2, 0.05, 0.1)$ for shape-aware training and $(1, 0, 0, 0)$ for value-only training. The response model in Table 2 uses shape-aware training for all three analyses. The coefficients are fixed empirical choices: value fitting anchors response levels, slope matching emphasizes local changes, and the feature term emphasizes curve summaries. The smaller curvature coefficient is consistent with the higher sensitivity of second differences to sampling and interpolation errors. The configurations compared in Table 9 are specified in Section C.3. At each update, the three analysis losses contribute equally to the shared encoders and their respective response heads:

$$(\eta^\star, \{\psi_a^\star\}_{a\in\mathcal{A}}) = \arg\min_{\eta, \{\psi_a\}_{a\in\mathcal{A}}} \underbrace{\frac{1}{3}\sum_{a\in\mathcal{A}} \mathcal{L}_{\mathrm{response}}^a(\eta, \psi_a)}_{\mathcal{L}_{\mathrm{pretrain}}}. \tag{37}$$

C Experimental Protocols and Additional Results

C.1 Experimental Environment, Baselines, and Tasks

Experimental environment. Experiments run on a Linux server with Intel Xeon Platinum 8368 CPUs and NVIDIA A100 (80 GB) GPUs, using Python 3.9, PyTorch 2.6, and CUDA 12.4. Model inference runs on GPU; circuit simulation on CPU.

Benchmarks. As described in Section 3, WaveFormer Benchmark is derived from AnalogGym (Li et al., 2024) and contains 33,695 instances across 30 topologies and six circuit categories. DC, AC, and transient response curves provide the pretraining supervision under the fixed parameter-instance split. Downstream evaluation covers held-out response curves, performance prediction from the pretrained representations, and sizing guided by the trained performance predictor. The performance evaluation also includes DC operating points extracted from predicted curves. `Basic_LDO` sizing optimizes five device-group variables to minimize area, DC line variation, and transient excursion while maximizing AC gain under DC/AC/transient feasibility constraints. Each sizing method receives 20 SPICE calls during optimization, followed by independent SPICE verification of its Pareto designs. Efficiency is measured under isolated execution.

Baselines. Response baselines include DeepONet (Lu et al., 2021), LSTM behavioral modeling (Hu et al., 2025), and the graph encoders PGNN, KCLNet, and Ckt2Vec (Hakhamaneshi et al., 2023; Xu et al., 2026b;a). We adapt the three graph encoders to response prediction with a common query-conditioned decoder. We compare operating-point prediction against PGNN-DC and direct metrics against autoregressive INSIGHT (Poddar et al., 2025) and topology-conditioned ZeroSim (Yang et al., 2025). Sizing baselines are LLM-guided AutoSizer (Yu et al., 2026), function-grouped multi-agent FD-MAGRPO (Jiang et al., 2026), FALCON's graph-surrogate-guided gradient search (Mehradfar et al., 2025), and TabPFN-based Bayesian CPN (Liu et al., 2026). AutoSizer uses GPT-5.6-Sol (ultra reasoning) instead of Gemini-2.5-Flash. Efficiency baselines are HSPICE, Spectre, APS, and ngspice (Synopsys, Inc., 2016; Cadence Design Systems, Inc., 2015; The ngspice Team, 2023).

Evaluation metrics. Response metrics are topology-macro symmetric L_2, Pearson correlation, NMAE, and NRMSE. Operating-point metrics include topology-macro and pooled MAE/RMSE, error quantiles, R^2, accuracy within 5 mV, and paired wins; direct metrics use Primary, quantile, and analysis-wise NRMSE plus failure-cell counts. Sizing uses normalized hypervolume (HV), IGD+ against the union of verified fronts, feasible-query rate, and verified Pareto-set size. Efficiency uses mean per-case minimum runtime, CPU peak resident memory, GPU peak tensor allocation, and process-level GPU memory. Instance-level symmetric L_2 is

$$E_{\mathrm{symL2}}(\widehat{Y}_a, Y_a) = \frac{2\|M_a^{(0)} \odot (\widehat{Y}_a - Y_a)\|_2}{\|M_a^{(0)} \odot \widehat{Y}_a\|_2 + \|M_a^{(0)} \odot Y_a\|_2 + \epsilon_E}. \tag{38}$$

Here $\epsilon_E > 0$ stabilizes the denominator. Response metrics use flattened valid entries, combining real and imaginary channels for AC. Empty response masks contribute neither a response loss nor a response-error observation.

C.2 Response Prediction Metrics

We evaluate the pretrained response model on held-out configurations by comparing predicted and SPICE reference curves at valid query coordinates.

Let I_0 index the valid flattened entries of one response, including both real and imaginary channels for AC. For NMAE and NRMSE, predictions and targets are normalized by their channel-specific training statistics $\mathcal{N}_j$ before flattening into entries $\widehat{\bar{y}}_n$ and $\bar{y}_n$, $n \in I_0$. The normalized error metrics are

$$\mathrm{NMAE} = \frac{1}{|I_0|} \sum_{n \in I_0} |\widehat{\bar{y}}_n - \bar{y}_n|, \qquad \mathrm{NRMSE} = \sqrt{\frac{1}{|I_0|} \sum_{n \in I_0} \left(\widehat{\bar{y}}_n - \bar{y}_n\right)^2}. \tag{39}$$

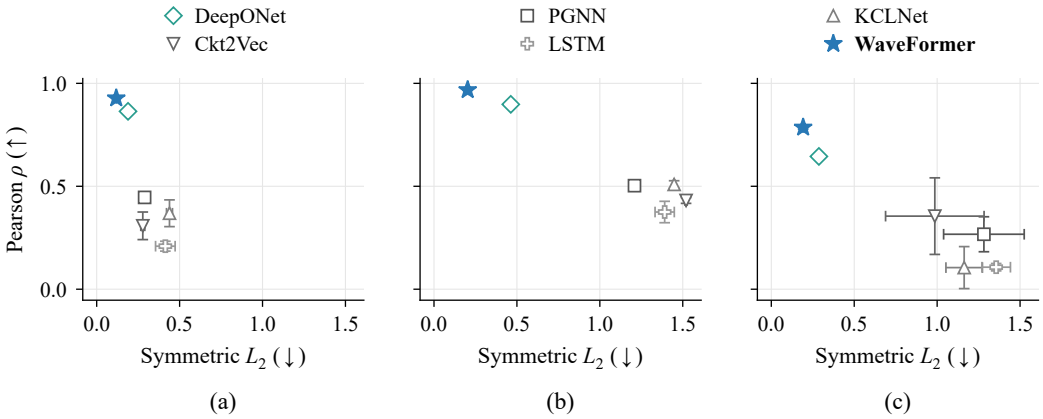


Figure 6: Response error and shape fidelity for (a) DC, (b) AC, and (c) transient prediction, with identical axis ranges. Horizontal and vertical error bars show ± 1 sample standard deviation around the means in Table 2. Lower symmetric L_2 and higher Pearson ρ are better.

Pearson correlation is computed on the same valid flattened vector. For per-instance metric $E_{i,\tau}$ on instance i of topology τ, the topology-macro result is

$$E_{\text{macro}} = \frac{1}{30} \sum_{\tau=1}^{30} \frac{1}{|\mathcal{I}_\tau|} \sum_{i \in \mathcal{I}_\tau} E_{i,\tau}. \tag{40}$$

In Figure 6, WAVEFORMER combines lower symmetric L_2 with higher Pearson correlation than all five baselines in each analysis. The common axes show both the error and shape-fidelity differences across tasks.

C.3 COMPONENT COMPARISONS ACROSS ANALYSES

We examine the circuit encoder, testbench encoder, response decoder, and masked shape-aware loss through component substitutions. The circuit-encoder variant uses a non-attentional message-passing network (MPNN), retaining the graph structure, relation attributes, and target readout. The testbench-encoder variant replaces hierarchical attention with a shared pointwise MLP and mean pooling within and across sources, preserving source values, roles, and physical coordinates. The decoder variant uses an MLP that directly receives the circuit and source representations, query coordinates, and the same fixed Fourier features. The loss variant retains value fitting alone, removing slope, curvature, and feature supervision.

Each variant retains the shared-encoder, three-head organization and uses the same split, training budget, and joint validation-based model-selection rule. The full model and the three architectural variants use shape-aware training, and the loss variant uses value fitting alone. Table 9 compares topology-macro symmetric L_2, Pearson correlation, NMAE, and NRMSE to assess response error, shape fidelity, and normalized amplitude error.

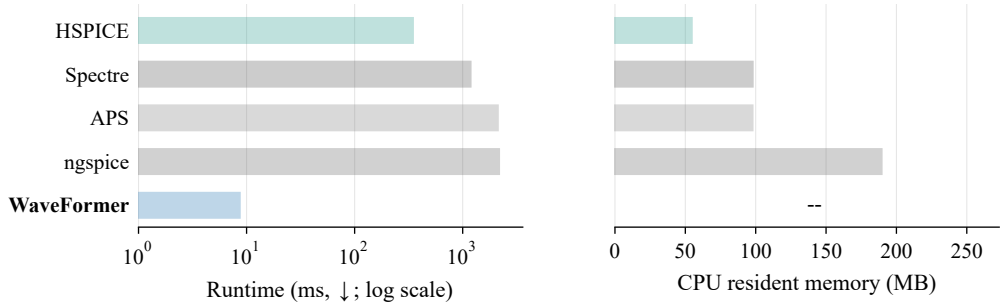
The full model has the best reported DC and transient scores across all four metrics. On DC, removing shape supervision lowers correlation from 0.928 to 0.875, the largest correlation change among the four substitutions. For transient responses, architecture substitutions raise symmetric L_2 from the full-model mean of 0.192 to 0.255–0.273, while removing shape supervision gives 0.196.

AC favors a different decoder: the MLP variant achieves the best values on all four metrics, including symmetric L_2 of 0.182 versus 0.203 for the full model. Removing shape supervision increases AC correlation but also increases all three error measures. These results distinguish improvements in amplitude accuracy from those in correlation and show how component effects vary across analyses.

Table 9: Component substitutions for circuit response prediction.

Metric	WaveFormer (Full)	w.o. Circuit attention	w.o. Testbench attention	w.o. Multiscale decoder	w.o. Shape supervision
(a) Response accuracy and shape fidelity (symmetric L_2, Pearson ρ)					
DC symmetric $L_2 \downarrow$	**0.118**	0.126	0.132	0.130	0.128
DC Pearson $\rho \uparrow$	**0.928**	0.924	0.921	0.925	0.875
AC symmetric $L_2 \downarrow$	0.203	0.259	0.208	**0.182**	0.206
AC Pearson $\rho \uparrow$	0.968	0.965	0.964	**0.987**	0.980
Tran. symmetric $L_2 \downarrow$	**0.192**	0.273	0.256	0.255	0.196
Tran. Pearson $\rho \uparrow$	**0.786**	0.655	0.695	0.637	0.785
(b) Normalized amplitude errors (NMAE, NRMSE)					
DC NMAE $\downarrow$	**0.178**	0.283	0.211	0.214	0.255
DC NRMSE $\downarrow$	**0.192**	0.298	0.226	0.229	0.272
AC NMAE $\downarrow$	0.065	0.081	0.066	**0.060**	0.074
AC NRMSE $\downarrow$	0.137	0.165	0.140	**0.131**	0.143
Tran. NMAE $\downarrow$	**0.083**	0.146	0.137	0.126	0.086
Tran. NRMSE $\downarrow$	**0.123**	0.196	0.180	0.174	0.126

Notes. w.o. denotes the component substitution described in the text. Bold marks the best reported value in each row. The full-model reference repeats the means in Table 2; ablation variants use one common seed.



WaveFormer GPU: mean peak tensors 19.3 MB; process 557.8 MB.

Figure 7: Response prediction: inference cost under isolated execution. (a) Mean per-case minimum runtime on a log axis with a common 1 ms bar baseline; (b) peak CPU resident memory on a linear axis starting at zero.

C.4 Inference Efficiency and Memory

Timing protocol. The timing set contains one preselected test configuration from each of the 30 topologies, evaluated under DC, AC, and transient analysis. Methods process these cases sequentially under isolated execution on the hardware in Section C.1. HSPICE, Spectre, APS, and ngspice run on CPU; WaveFormer performs single-instance inference on GPU (Synopsys, Inc., 2016; Cadence Design Systems, Inc., 2015; The ngspice Team, 2023). For each case, simulator execution is repeated five times. Model inference uses ten warm-up calls followed by 100 timed calls, with device synchronization around each call. The reported runtime is the arithmetic mean of the minimum successful runtime for each case. Model timing covers the forward pass with weights and inputs already resident on the device; simulator timing covers execution of the analysis. Data preparation, model loading, and training are outside the model-inference measurement.

Memory measurements. The model's 19.3 MB entry in Table 4 is the mean per-case peak tensor allocation on the GPU; its 557.8 MB entry measures total process-level GPU memory. Simulator memory entries measure peak resident memory of the simulator process and its children.

C.5 Performance Prediction Metrics

Response-derived operating points. DC operating-point evaluation extracts the registered output voltage at 25 °C from the predicted DC response and compares it with the SPICE reference

and PGNN-DC (Hakhamaneshi et al., 2023). Only registered responses supporting this operating-point measurement enter the paired comparison. Topology-macro MAE and RMSE average the corresponding errors within each topology and then across topologies. Pooled statistics combine all paired predictions. Accuracy@5 mV is the fraction with absolute error at most 5 mV, and paired wins is the fraction with smaller absolute error than the competing method.

Direct circuit metrics. The performance predictor in Section C.7 is trained on response-pretrained representations to predict circuit metrics jointly and is compared with INSIGHT and ZeroSim (Poddar et al., 2025; Yang et al., 2025). For metric j, let s_j^{metric} be its range over valid training labels pooled across topologies, bounded below by a positive numerical floor. The same scale is used for all methods. For topology g, the cell score is

$$E_{g,j} = \frac{1}{s_j^{\mathrm{metric}}} \sqrt{\frac{1}{|\mathcal{I}_{g,j}^{\mathrm{test}}|} \sum_{i \in \mathcal{I}_{g,j}^{\mathrm{test}}} \left(\widehat{p}_{i,j} - p_{i,j}\right)^2}, \tag{41}$$

where $\mathcal{I}_{g,j}^{\mathrm{test}}$ contains test instances with a valid label for metric j on topology g. Inapplicable or non-finite labels are masked. A cell is excluded when its topology-specific training range is at most 10^{-12} times its training standard deviation; this training-derived eligibility rule is shared across methods.

Primary NRMSE averages $E_{g,j}$ over eligible metrics within each topology and then over topologies. The median, P90, P95, and P99 are quantiles across eligible cells; DC, AC, and transient scores restrict this aggregation to the corresponding metric category. A failure cell satisfies $E_{g,j} > \max(0.1, 2E_{g,j}^{\mathrm{null}})$, where the null predictor returns the topology-specific training mean.

C.6 Circuit Sizing Protocol

Design space and objectives. The `Basic_LDO` task varies five coupled device groups: the first and second transconductance groups, the pass device, the load, and the bias network. Each coordinate lies in $[0, 1]$, with 0, 0.5, and 1 corresponding to the low, nominal, and high parameter settings. Within a group, device lengths, widths, and multiplicities vary together while preserving matching relations. Parameter values are interpolated piecewise linearly between the settings in Table 10. Integer multiplicities are rounded, and the remaining circuit parameters retain their nominal values.

Table 10: Device-group parameter settings for `Basic_LDO` sizing.

Device group	Length [nm]	Width [μm]	Multiplicity
First transconductance group	$(3000, 4000, 5000)$	$(4.125, 5.5, 6.875)$	$(6, 8, 10)$
Second transconductance group	$(750, 1000, 1250)$	$(6, 8, 10)$	$(5, 7, 9)$
Pass device	$(150, 150, 187.5)$	$(6.75, 9, 11.25)$	$(696, 928, 1160)$
Load	$(1500, 2000, 2500)$	$(4.5, 6, 7.5)$	$(6, 8, 10)$
Bias network	$(750, 1000, 1250)$	$(3.75, 5, 6.25)$	$(2, 3, 4)$

Notes. Each tuple gives the settings at group coordinates 0, 0.5, and 1.

Let A denote the device-area proxy $\sum_d W_d L_d M_d$ over the devices in these five groups, where W_d, L_d, and M_d are width, length, and multiplicity. Let ΔV_{DC} be the DC output range, G_{AC} the low-frequency gain in dB, and ΔV_{tr} the transient output range. The four minimization objectives are

$$\mathbf{f} = \left[\frac{A}{A_0}, \frac{\Delta V_{\mathrm{DC}}}{\Delta V_{\mathrm{DC},0}}, \frac{G_{\mathrm{AC},0} - G_{\mathrm{AC}}}{10}, \frac{\Delta V_{\mathrm{tr}}}{\Delta V_{\mathrm{tr},0}} \right], \tag{42}$$

where subscript 0 denotes the nominal design. The nominal scales are $A_0 = 1987.8\,\mu\mathrm{m}^2$, $\Delta V_{\mathrm{DC},0} = 40.358$ mV, $G_{\mathrm{AC},0} = 57.030$ dB, and $\Delta V_{\mathrm{tr},0} = 1.306$ V. Hypervolume is computed in this normalized objective space with reference point $(1.25, 1.50, 0.60, 1.15)$.

Feasibility. DC outputs must lie between 1.50 and 1.65 V, with a range no greater than 0.06 V. AC gain must be at least 51.3 dB, unity-gain bandwidth at least 459 kHz, and phase margin at least 75°. Transient outputs must lie between 0.35 and 1.90 V, settle to a final value between 1.50 and 1.70 V, and have a range no greater than 1.50 V. The additional limits are 1.40 V overshoot, 0.25 V undershoot, $100\,\mu$s settling time within 2%, and 0.15 V peak-to-peak tail ripple. SPICE simulation results determine satisfaction of these constraints and membership in the feasible Pareto set.

Surrogate-guided search. WAVEFORMER evaluates candidates with an ensemble of three performance predictors trained on response-pretrained representations in independent downstream runs. An online calibration model fits a robust affine correction followed by a radial-basis kernel regression to the SPICE residuals observed during the search. The outer optimizer uses the cross-entropy method: each of eight iterations samples 768 candidates from a clipped diagonal Gaussian over the design coordinates. The initial mean is the nominal design, and the initial standard deviation is 0.22. The best 20% of candidates update the sampling mean and standard deviation with smoothing factor 0.2 and minimum standard deviation 0.04. Candidate ranking prioritizes feasibility, followed by nondominated rank and crowding distance; predicted uncertainty supplies conservative constraint margins and exploratory evaluations.

Every method receives 20 SPICE evaluations during optimization, including its initial evaluations. For WAVEFORMER, selected candidates are simulated throughout the search, and their measured residuals update the calibration model. After optimization, each reported Pareto design is independently re-simulated for verification. The reported front contains verified feasible designs. Let $\mathcal{P}$ be a method's verified front in normalized objective space and $\mathcal{P}_{\mathrm{ref}}$ the nondominated union of the five verified fronts. Then

$$\mathrm{IGD}^{+}(\mathcal{P}) = \frac{1}{|\mathcal{P}_{\mathrm{ref}}|} \sum_{\mathbf{u}\in\mathcal{P}_{\mathrm{ref}}} \min_{\mathbf{v}\in\mathcal{P}} \sqrt{\sum_{k=1}^{4} \max(v_k - u_k, 0)^2}. \tag{43}$$

The feasible-query rate is the fraction of the 20 optimization evaluations satisfying all constraints; Pareto-set size counts verified nondominated designs.

C.7 DOWNSTREAM PERFORMANCE PREDICTOR

Transfer from response pretraining. The shared circuit and testbench encoders are pretrained with the three response heads using Equation (37). For performance prediction, we transfer both encoders and all learnable parameters in their fusion, and replace the three response heads with a randomly initialized joint metric-prediction head. The transferred backbone retains its shared parameters across analyses and is fine-tuned end to end with the metric head. Both stages use the same configuration-level training, validation, and test partitions in Section A, with preprocessing statistics fitted only on training data.

Architecture. For each analysis, the shared backbone processes the observation-marked circuit and its excitation to obtain $h_a = [h_{G,a}; h_{U,a}]$. Projected analysis conditions are added to the corresponding circuit–testbench contexts, while projected device-parameter features provide additional decoder inputs. Each of the 202 metrics has a learned query conditioned on its identity, analysis category, and corresponding analysis context. The joint metric head uses a Transformer decoder with self-attention among metric queries and cross-attention to the three circuit–testbench contexts and parameter features, masking padded parameter entries.

Layer normalization and a shared two-layer MLP with GELU activation map each decoded query to a standardized scalar prediction, which is converted to physical units using the training statistics below. The shared circuit encoder and the metric decoder use two layers, four attention heads, and hidden dimension 212; the decoder feed-forward dimension is 424. The model jointly outputs 202 metrics: 103 DC, 45 AC, and 54 transient metrics. Supervision masks select the metrics applicable to each circuit.

For a performance task τ, the output interface is

$$\widehat{\mathbf{p}}_\tau = H_{\omega_\tau}^{\mathrm{performance}}(r_\tau), \tag{44}$$

where r_τ combines the fine-tuned circuit–testbench contexts, analysis conditions, and device-parameter features, and the output vector contains the metrics required by the task.

Metric normalization. For metric j, let μ_j^{metric} and $\sigma_j^{\mathrm{metric}} > 0$ be the mean and standard deviation of valid training labels pooled across topologies, with a positive numerical floor on the standard deviation. The standardized target and conversion back to physical units are

$$\bar{p}_{i,j} = \frac{p_{i,j} - \mu_j^{\mathrm{metric}}}{\sigma_j^{\mathrm{metric}}}, \qquad \widehat{p}_{i,j} = \mu_j^{\mathrm{metric}} + \sigma_j^{\mathrm{metric}} \widehat{\bar{p}}_{i,j}. \tag{45}$$

Standardization defines the output coordinates, while the training-label range s_j^{metric} in Section C.5 scales the prediction error. Each metric has its own range pooled across training topologies, shared by training, validation, and testing. All normalization statistics remain fixed during fine-tuning and evaluation.

Fine-tuning objective. Let $\mathcal{T}$ be the set of training topologies, $\mathcal{J}_g$ the eligible metrics for topology g, and $\mathcal{I}_{g,j}^{\mathrm{train}}$ the instances with valid labels in that cell. The objective assigns equal weight to topologies and to their eligible metrics:

$$\mathcal{L}_{\mathrm{metric}} = \frac{1}{|\mathcal{T}|} \sum_{g \in \mathcal{T}} \frac{1}{|\mathcal{J}_g|} \sum_{j \in \mathcal{J}_g} \frac{1}{|\mathcal{I}_{g,j}^{\mathrm{train}}|} \sum_{i \in \mathcal{I}_{g,j}^{\mathrm{train}}} \left(\frac{\widehat{p}_{i,j} - p_{i,j}}{s_j^{\mathrm{metric}}} \right)^2 . \tag{46}$$

The training-only scales and eligibility rules are those in Section C.5. In standardized coordinates, each squared-error term is equivalently $\left[(\widehat{\widetilde{p}}_{i,j} - \widetilde{p}_{i,j})\sigma_j^{\mathrm{metric}} / s_j^{\mathrm{metric}}\right]^2$. Mini-batch estimates use inverse-frequency weights derived from valid training-label counts to preserve equal weighting of topologies and their eligible metrics. Missing or inapplicable labels are masked, and the training-derived cell eligibility mask is fixed for optimization, validation, and testing.

Optimization and model selection. The joint response model is pretrained for 5,000 AdamW updates with batch size 30, initial learning rate 3×10^{-4} decayed to zero by a cosine schedule, weight decay 10^{-6}, and gradient-norm clipping at 5. Response validation is performed after each full pass through the training set.

The selection score equally averages the validation normalized waveform MAE of the three analyses:

$$S_{\mathrm{val}} = \frac{1}{3} \sum_{a \in \mathcal{A}} \mathrm{MAE}_{\mathrm{norm},a}^{\mathrm{val}}. \tag{47}$$

Here $\mathrm{MAE}_{\mathrm{norm},a}^{\mathrm{val}}$ is the validation normalized waveform MAE for analysis a, evaluated without topology-macro aggregation. The checkpoint with the lowest S_{val} supplies all three heads for response evaluation and the shared backbone for downstream fine-tuning.

The joint performance predictor is fine-tuned end to end with AdamW for 5,000 updates, using the same batch size, weight decay, and clipping threshold. The learning rate increases linearly to 3×10^{-4} over 250 updates and then follows a cosine schedule to 3×10^{-5}. Validation is performed every 100 updates, and the model with the lowest validation Primary NRMSE is used for testing.

Pretraining comparison. The comparison in Figure 4(d) uses the same joint predictor architecture, data partitions, fine-tuning objective and budget, and model-selection rule. The pretrained model initializes the shared circuit encoder, testbench encoder, and learnable fusion from response prediction; the model without pretraining initializes them randomly. Both models optimize the full shared backbone and the joint metric head using scalar performance labels. The pretrained variant additionally receives the response supervision and pretraining updates described above; this comparison does not match the total training cost of the two variants.

D Comparison with Related Work

Supervised pretraining and circuit representations. PGNN pretrains graph neural networks on DC node-voltage prediction and adapts the learned representations to circuit-property prediction and design (Hakhamaneshi et al., 2023). It establishes circuit behavior as a supervision signal for representation learning. KCLNet uses KCL-inspired electrical-equivalence constraints, while Ckt2Vec encodes device I–V curves through spectral features (Xu et al., 2026b;a). WaveFormer conditions circuit representations on sampled source functions and observation targets, extending response supervision to DC transfer characteristics, complex AC frequency responses, and transient waveforms. We evaluate the learned shared circuit–testbench backbone as an initialization for scalar performance regression, with the predictor architecture and fine-tuning protocol held fixed.

Analog performance prediction and benchmarks. Analog performance predictors map design vectors or circuit graphs to scalar specifications. Existing methods exploit graph representations, dependencies among measurements, or shared models across topologies, while predicting task-specific metrics (Dong et al., 2023; Poddar et al., 2025; Yang et al., 2025). AnalogGym provides open netlists, parameter spaces, testbenches, and evaluation infrastructure for analog synthesis (Li et al., 2024). We use these resources to connect response supervision, scalar performance labels, and design evaluation under a common data interface and a fixed configuration-level split.

Neural operators and circuit response modeling. DeepONet approximates nonlinear operators by combining an input-function encoder with coordinate-conditioned decoding (Lu et al., 2021). Fourier features and Fourier neural operators provide frequency-based representations for function and operator learning (Tancik et al., 2020; Li et al., 2021). Circuit-behavioral formulations of DeepONet and LSTM learn excitation-dependent responses (Zhang et al., 2023; Hu et al., 2025). Recent surrogates further explore prediction across circuit topologies and generative transient synthesis (Yazici, 2026; Karipidis et al., 2026). More specifically, Zhang et al. (2023) study the zero-state input–output response of one second-order band-pass circuit. Yazici (2026) train a topology-conditioned surrogate over 21 heterogeneous analog circuits for transient-waveform prediction. Karipidis et al. (2026) synthesize continuous transient responses for a 12-parameter charge-pump circuit with a generative decoder. WAVEFORMER conditions a DeepONet-based response head on circuit-graph and source representations and augments it with a Fourier-residual path. Training this head on response curves supervises the joint dependence on circuit structure, excitation, and observation targets. Our benchmark evaluates this formulation on DC, complex AC, and transient responses with a common input and query interface.

Learning and evaluation scope. The response task supervises a shared circuit–testbench backbone, which is subsequently fine-tuned with a joint metric head. The performance predictor then guides a fixed sizing procedure, with SPICE used to verify the returned designs. These stages allow response accuracy, the incremental benefit of backbone pretraining, and design utility to be assessed separately. Our configuration-level split tests new parameter configurations of known topologies; the `Basic_LDO` experiment evaluates one fixed-budget design setting.

Table 1 in Section 2 summarizes the capabilities reported in these studies and compares them with those evaluated for WAVEFORMER. Section 5.1 describes the baselines used in the response-prediction experiments.