CHIP CAPACITANCE PREDICTION BY A JOINTLY ROUTED TRANSFORMER

Anonymous authors
Paper under double-blind review

ABSTRACT

Accurate parasitic capacitance estimation is essential for chip design and verification. Lookup-table-based parasitic extraction (LUT-based PEX) reuses precomputed capacitance tables to avoid weeks to months of physical simulations, but sparse coverage limits accuracy for unobserved chip layout geometries. To address this limitation, we propose a jointly routed Transformer for chip capacitance prediction, conditioned on layout geometry, material context, and available capacitance observations. Our key idea is to train a shared Transformer to jointly learn residual corrections to an interpolation estimate and adaptive weights for combining neural and numerical corrections. Notably, the model extrapolates beyond the observed geometry range and enables transfer to an unseen manufacturing process without parameter updates, aided by our structured tokenizer. Moreover, we curate and augment a unified dataset and evaluation suite for capacitance prediction from sparse observations, advancing AI research in LUT-based PEX. Our method outperforms commercial-style multilinear interpolation in all 15 evaluated settings and reduces normalized extrapolation error by 84–86% on public tables. Encouragingly, these gains extend to downstream chip analysis: timing estimates using our predicted capacitances more closely match the full-table reference in all 16 settings across eight fixed layouts.

1 INTRODUCTION

Accurate parasitic capacitance estimation is essential for chip design and verification because capacitance affects signal delay and coupling between wires (Yu et al., 2021). Parasitic extraction (PEX) supplies these estimates for static timing analysis (STA) and circuit simulation. Field solvers compute capacitance from electrostatics (Nabors & White, 1991), but characterizing many geometries is expensive. Lookup-table-based PEX (LUT-based PEX) reuses precomputed values through lookup and interpolation (OpenROAD Project, n.d.), as illustrated in Figure 1. Sparse tables reduce characterization costs but limit prediction accuracy within and beyond the observed geometry range, motivating better interpolation and extrapolation from available observations.

Existing learning-based methods predict capacitance from geometry using multilayer perceptrons (Kasai et al., 2019), convolutional and graph networks (Yang et al., 2023; Liu et al., 2024), and point-cloud and attention-based Transformers (Guo et al., 2024; Huang et al., 2026). Residual and hybrid approaches combine learned models with physical or numerical methods (Tsai et al., 2025b; Abouelyazid et al., 2022). However, direct surrogates encode physical responses in model parameters, making generalization sensitive to training coverage. Recent work on interconnect geometry exploration reports rising errors near training thickness boundaries, requiring broader coverage to maintain accuracy (Tsai et al., 2025a). Rule tables already contain valuable process-specific physical calculations. We therefore provide observed geometry–capacitance pairs at inference, conditioning queries on the current table's sparse observations and material context, following the principle of learning from context sets (Garnelo et al., 2018; Kim et al., 2019). Here, extrapolation is relative to the current observations, rather than necessarily to the source training distribution. Appendix A discusses related work in detail.

We propose a jointly routed Transformer for chip capacitance prediction. Our key idea is to jointly learn residual corrections to an interpolation estimate and adaptive weights for combining neural and numerical corrections. A structured tokenizer encodes geometry, material context, and available capacitance observations, allowing a shared Transformer to condition both prediction and input-dependent routing (Rosenbaum et al., 2018) on each table. The neural residual allows predictions

Anonymous code and reproducibility artifacts are available at
`https://anonymous.4open.science/r/residual-gating-capacitance-9727/`

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.

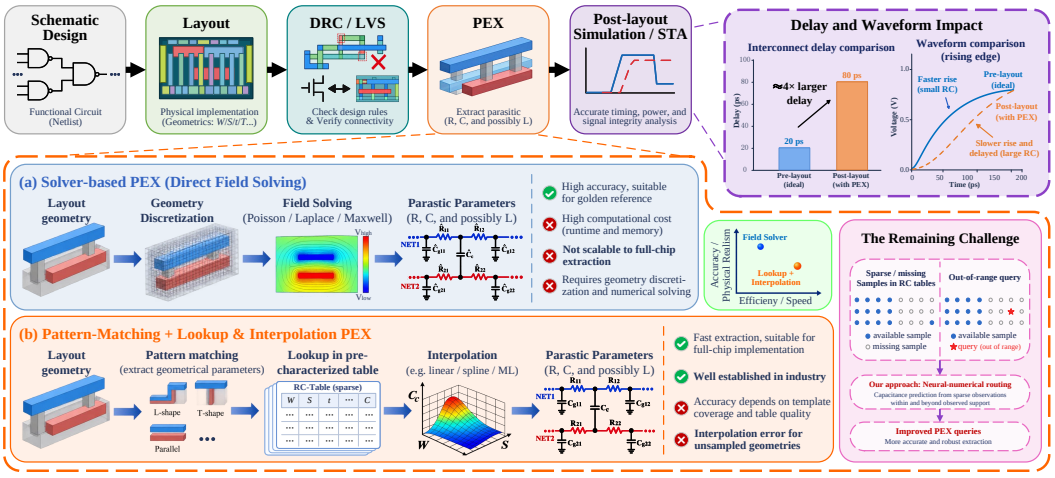


Figure 1: Parasitic extraction by (a) field solving or (b) characterized patterns. We predict capacitance from sparse observations and audit its downstream timing effects.

beyond clamped interpolation boundaries. Notably, observation conditioning supports transfer to an unseen manufacturing process without parameter updates.

Motivated by public benchmarks for AI in chip design (Jiang et al., 2024), we curate a unified LUT-based PEX dataset and evaluation suite combining rule tables from four manufacturing technologies, an independent finite-element reference, and a source-only transfer protocol. Controlled observation masks enable separate evaluation of interpolation and extrapolation. At 30% missingness, our method outperforms rank-matched commercial-style multilinear interpolation in all 15 evaluated settings and reduces normalized extrapolation error by 84–86% on public tables. Encouragingly, incorporating predictions into extraction rules improves agreement with full-table capacitance and timing estimates in all 16 downstream settings across eight fixed layouts. **Our main contributions are summarized as follows:**

- **A generalizable neural framework for LUT-based PEX.** To our knowledge, we are the first to introduce a neural framework that enhances LUT-based PEX through both geometry extrapolation and cross-process transfer without parameter updates.
- **A unified dataset and evaluation suite.** We curate and augment capacitance tables from four manufacturing technologies with independently solved reference grids, establishing a unified benchmark for interpolation, extrapolation, and source-only cross-process transfer.
- **Downstream validation.** Experiments on eight fixed routed layouts demonstrate improved agreement with full-table capacitance and timing estimates in all 16 evaluated settings.

2 Terminology

Capacitance tables and material stacks. A capacitance table maps sampled wire geometries to capacitance components. Each row corresponds to a geometry x_i and a response vector y_i; X denotes all candidate geometries and K the number of response components. A material stack describes the conductor and dielectric layers of a process. The context c encodes structural and material information associated with these geometries. A table stores sampled values of the capacitance response; the prediction task estimates responses at supplied, unobserved query geometries.

Support, queries and masks. The support S is the set of observed row indices, with available responses y_S. The query set $\mathcal{Q}$ contains the remaining indices whose responses must be predicted. A mask determines which responses are hidden; query coordinates remain available. *Support-conditioned* means that a prediction depends on the current table's observed geometry–response pairs. The missingness rate is the fraction of candidate rows hidden by the mask.

Support rank and prediction regimes. Support rank counts the independent geometric directions spanned by the observed coordinates, which can be fewer than the number of stored coordinates. An *in-support* query lies inside the observed support's convex hull in its affine span; an *out-of-support* query lies outside that supported region. These terms refer to the current observations, not the support of a training distribution or the extent of the original reference table. Random, contiguous-block and boundary-extrapolation masks specify how observations are withheld.

Anchor, residual and gate. The anchor b_i is a deterministic interpolation estimate computed from observed support. A residual $\Delta_{i,e}$ is a correction relative to this anchor, supplied by a neural or numerical branch e. The gate assigns nonnegative weights $\pi_{i,e,k}$ that sum to one across applicable branches for each query i and capacitance component k. An *anchor-only* control returns b_i; the full model adds the gated residual corrections.

Source-only transfer. The source stack supplies the training data, and the target stack supplies the tables evaluated after training. Source-only transfer keeps model parameters fixed while using the target table's available observations as support; it does not perform target-domain parameter updates.

3 METHOD

3.1 TASK AND OVERALL ARCHITECTURE

We formulate capacitance estimation from sparse observations as conditional query prediction. Each episode supplies support and query geometries $X = \{x_i\}_{i=1}^{N}$, with $x_i \in \mathbb{R}^d$ ($d = 3$ stored coordinates), a structural/material context c, and K capacitance components ($K \in \{2, 3\}$). Only the support $S \subseteq \{1, \dots, N\}$ has characterized values y_S; the model predicts responses at the query indices $\mathcal{Q} = \{1, \dots, N\} \setminus S$. Queries may lie inside or outside the observed hull. In the experiments, masks partition characterized tables into support and queries so that withheld responses provide evaluation targets. Thus, the learned mapping is $F_\theta : (X, c, S, y_S) \mapsto \hat{y}_\mathcal{Q}$. All candidate coordinates are known, but query labels are absent from every input and support computation. Training-derived coordinate scales ξ and family-specific component scales σ are fixed before evaluation.

Figure 2 summarizes the architecture: a structured tokenizer and shared Transformer feed a residual head and a component-wise gate. Both heads use the same query representation, so learning the correction and deciding how much to trust it are coupled. In physical units,

$$\hat{y}_{i,k} = \left[b_{i,k} + \sum_{e \in \mathcal{E}_i} \pi_{i,e,k} \Delta_{i,e,k} \right]_+, \qquad \sum_{e \in \mathcal{E}_i} \pi_{i,e,k} = 1, \qquad \pi_{i,e,k} \geq 0, \tag{1}$$

where $[z]_+ = \max(0, z)$, b_i is an observation-derived anchor, $\Delta_{i,e}$ is a correction to that anchor, and $\mathcal{E}_i$ contains applicable branches. The network learns *what correction is needed* and *how strongly to use it* through the same query-prediction objective. We use the term residual gating for this anchor-centered soft mixture within the routing architecture.

Our design follows three observations about sparse characterization. First, the table's own support provides a useful numerical reference, so we learn departures from that reference. Second, local interpolation and learned extrapolation fail differently as support changes, motivating complementary correction branches. Third, support quality varies by query and capacitance component, motivating local routing instead of one fixed mixture per table.

3.2 STRUCTURED GEOMETRY–OBSERVATION TOKENIZATION

The tokenizer assigns one numerical token to each candidate geometry and one to each available dielectric region. Keeping geometry, observation status and capacitance in the same token preserves their physical correspondence.

Geometry tokens and label removal. Let $m_i = \mathbf{1}[i \in S]$. Define $\tilde{y}_i = y_i/\sigma$ for $i \in S$ and $\tilde{y}_i = 0$ otherwise; the zero is an input placeholder, distinguished from a measured zero by m_i. The geometry token is

$$t_i^{\text{geo}} = [\log(1 + x_i/\xi)\,;\, c\,;\, \tilde{y}_i\,;\, m_i\,;\, b_i/\sigma\,;\, (1, 0)] \in \mathbb{R}^{17+2K}. \tag{2}$$

Here division and logarithms are componentwise. The 11-dimensional context c comprises a four-way structural-family indicator (OVER, UNDER, OVERUNDER, DIAGUNDER; see Appendix B for the structures), the target metal and its two context-layer indices divided by 16, and three material-summary values (target-conductor bottom height, thickness, and local relative permittivity) with an availability flag. The material block is zero when unavailable. These family indicators encode interconnect arrangement, rather than a process-identity vocabulary. The anchor field exposes the current table's numerical reference to the encoder, so the learned representation can focus on its correction.

Material tokens. For a dielectric region with bottom/top coordinates z_r^-, z_r^+ and relative permittivity ϵ_r, the corresponding token is

$$t_r^{\text{mat}} = \left[\log(1 + z_r^-/\xi_1), \log(1 + z_r^+/\xi_1), \log(1 + \epsilon_r)\,;\, c\,;\, 0_K\,;\, 0\,;\, 0_K\,;\, (0, 1)\right]. \tag{3}$$

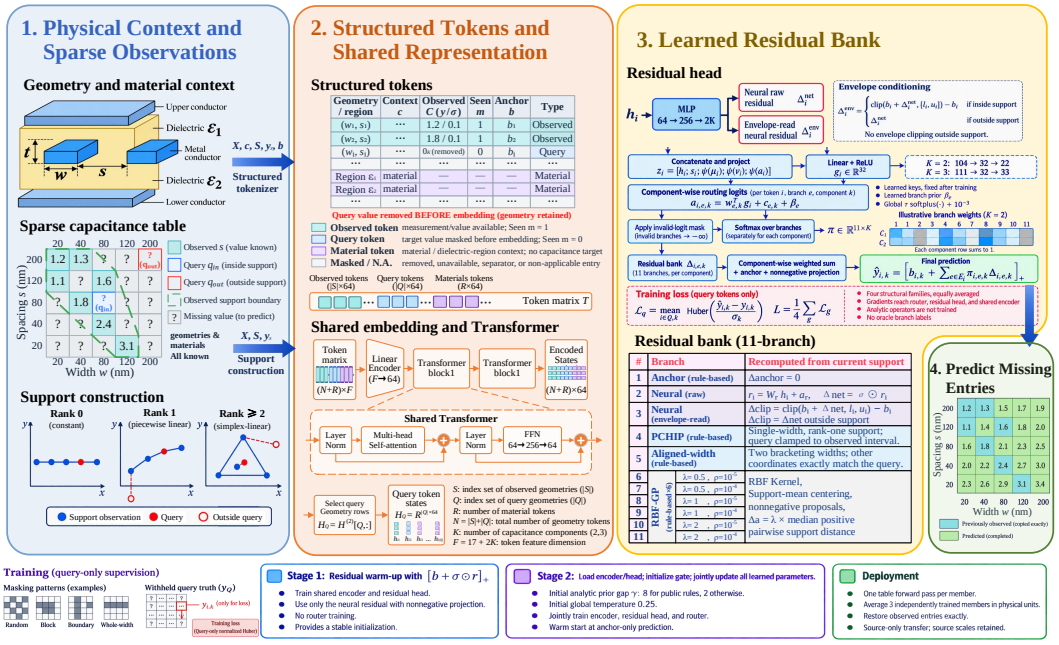


Figure 2: Support-conditioned neural–numerical routing. Blue: learned modules; gold: numerical computations; green: prediction assembly. One Transformer supplies the residual and routing state; observed values are preserved.

The type flags distinguish geometry from material despite their common input width. Material tokens make the dielectric profile directly accessible through attention, while the summary fields provide a compact table-level context. Tables without region descriptions simply have no material tokens.

Embedding and information boundary. Stacking N geometry and R material tokens gives $T \in \mathbb{R}^{(N+R)\times(17+2K)}$, projected row-wise to width 64. Query labels are zeroed *before* embedding; the attention mask excludes only padding. Explicit physical coordinates replace sequence-index positional embeddings, and predictions are read from the corresponding geometry tokens. Appendix C illustrates the token fields.

3.3 Support-conditioned Transformer encoder

$$
\begin{aligned}
U^{(l)} &= H^{(l-1)} + \mathrm{MHSA}_l(\mathrm{LN}(H^{(l-1)})),\\
H^{(l)} &= U^{(l)} + \mathrm{FFN}_l(\mathrm{LN}(U^{(l)})), \quad l = 1, 2.
\end{aligned}
\tag{4}
$$

Let $H^{(0)} = TW_{\mathrm{in}}^{\top} + b_{\mathrm{in}}$. We use two pre-normalized Transformer blocks Vaswani et al. (2017), each with four attention heads of width 16 and a $64 \to 256 \to 64$ feed-forward network, as in Eq. (4). Bidirectional attention combines local observations, geometric trends and material context into $h_i = H_i^{(2)}$. Query geometries can interact, but their labels remain unavailable. With physical coordinates and no index embedding, the encoder is permutation equivariant at inference. The feed-forward activation is ReLU and stage-2 dropout is zero; Appendix C gives the attention and padding equations.

3.4 Support anchor and hybrid residual representation

Geometric reference and local envelope. The deterministic support module uses only (X, S, y_S) to construct b_i, a local interval $[\ell_i, u_i]$, and an inside-support indicator I_i. Observed coordinates are range-normalized and reduced to independent axes. For support rank zero, the anchor copies the constant observation; for rank one, it uses piecewise linear interpolation with endpoint clamping. At higher rank it uses simplex-linear interpolation in reduced coordinates, with nearest-observation fallback outside the reduced hull or upon triangulation failure. Simplex interpolation and a complete-grid multilinear rule agree on observed nodes but can differ inside a cell; we evaluate the anchor separately from the commercial baseline. The inside indicator additionally checks the original affine span. Inside support, $[\ell_i, u_i]$ is the componentwise range of the bracketing or simplex vertices.

Outside support it records the nearest observation, but is not imposed on the neural extrapolation branch.

Learned residual branches. The encoder's pointwise linear head predicts a normalized correction $r_i = W_r h_i + a_r$. In physical units, the two neural branches are

$$\Delta_{i,\mathrm{net}} = \sigma \odot r_i, \qquad \Delta_{i,\mathrm{clip}} = \begin{cases} \mathrm{clip}(b_i + \Delta_{i,\mathrm{net}}, \ell_i, u_i) - b_i, & I_i = 1, \\ \Delta_{i,\mathrm{net}}, & I_i = 0. \end{cases} \tag{5}$$

These branches share all parameters. They give the router access to both a flexible learned correction and its locally range-preserving version, without training two separate networks. Restricting the clipped reading to supported queries keeps the model's ability to extrapolate. The local envelope is an inductive bias from observed values, not a general physical bound on capacitance.

Analytic correction operators. Eight additional operators provide complementary local and smooth interpolation biases: one shape-preserving cubic (PCHIP) Fritsch & Butland (1984), one aligned-width linear rule, and six RBF Gaussian-process predictors Rasmussen & Williams (2006) with different fixed kernel scales and ridges. Together with the zero-correction anchor and the two neural branches, they form $E = 11$ slots. Each analytic operator is recomputed from the current support, and enters as $\Delta_{i,e} = f_e(x_i; S, y_S) - b_i$. PCHIP applies to single-width tables. The aligned-width rule interpolates between two observed widths only when their other coordinates exactly match the query. Unsupported or nonfinite proposals are masked out. Appendix D gives the interpolation formulas and boundary conventions; Appendix C specifies the branch settings and tensor dimensions. Figure 6 illustrates the correction dictionary.

The operator types are fixed, but their predictions change with the support; the neural branches also change with θ. Expressing every proposal as a correction to the same anchor lets the gate compare them on a common scale.

3.5 Support-aware component routing

Routing state. The gate receives both the learned state and an explicit description of the available evidence. Let $s_i \in \mathbb{R}^{23+d+4K}$ concatenate table context with support density, distances, bracketing extent, support rank and region, directional coverage, envelope widths, aligned-width availability, and local variation/slope/curvature estimated from observed values. These features are *query-label-free*; they intentionally use known support capacitances.

$$z_i = [h_i; s_i; \psi(\mu_i); \psi(v_i); \psi(a_i)], \qquad g_i = \mathrm{ReLU}(W_g z_i + a_g) \in \mathbb{R}^{32}. \tag{6}$$

To summarize branch agreement, define $\bar{\Delta}_{i,e} = \Delta_{i,e}/\sigma$ and compute, componentwise over applicable branches, its mean μ_i, standard deviation v_i and range a_i. With $\psi(z) = \mathrm{sgn}(z)\log(1 + |z|)$, Eq. (6) constructs the routing state. Agreement statistics include both neural branches, allowing the routing loss to reach the residual head through this path as well.

Component-wise attention over correction branches. For branch e and capacitance component k, the output projection defines the logits and weights in Eq. (7).

$$a_{i,e,k} = w_{e,k}^{\top} g_i + c_{e,k} + \beta_e, \qquad \pi_{i,e,k} = \frac{\exp(a_{i,e,k}/\tau)}{\sum_{e' \in \mathcal{E}_i} \exp(a_{i,e',k}/\tau)}, \quad e \in \mathcal{E}_i. \tag{7}$$

Here g_i is a query-dependent state and $w_{e,k}$ is a learned branch/component key, fixed after training. Unlike token self-attention, this routing compares correction branches. Aggregate disagreement keeps the gate compact, but does not identify which individual branch makes an extreme proposal; Appendix E examines this limitation.

Inapplicable logits are masked before softmax. A learned global temperature $\tau = \mathrm{softplus}(t) + 10^{-3}$ controls routing sharpness; β_e is a learned branch prior shared across components. Different components can therefore prefer different corrections at the same geometry. Because the anchor is the zero-correction branch, allocating mass to it continuously attenuates the other corrections while retaining the skip path in Eq. (1). Nonnegative normalized weights make this a convex mixture of branch values, followed by the nonnegativity projection.

3.6 Masked-query training and joint optimization

Training episodes. At each update we sample one episode from each of the four structural families, first choosing a process and then a training table within that process. We sample random missingness, a contiguous block, or boundary extrapolation, adding whole-width masks for tables with multiple widths. The missing rate is sampled from $\{10, 30, 50, 70\}\%$, with a fresh mask seed.

The tokenizer retains all candidate geometries and exposes only y_S. Supervision uses withheld capacitance responses. Changing the mask changes both the conditioning observations and whether each query requires interpolation or extrapolation; the objective is continuous query prediction.

Final-output objective. Both the residual head and the router are optimized through the assembled prediction. For an episode,

$$L_g = \frac{1}{K|\mathcal{Q}|} \sum_{i \in \mathcal{Q}} \sum_{k=1}^{K} \rho\left(\frac{\hat{y}_{i,k} - y_{i,k}}{\sigma_k} \right), \qquad \rho(v) = \begin{cases} v^2/2, & |v| \le 1, \\ |v| - 1/2, & |v| > 1. \end{cases} \tag{8}$$

The reported stage-2 objective is $L = \frac{1}{4} \sum_g L_g$. Group-DRO Sagawa et al. (2020) is available in the implementation, but its update rate is zero in this configuration. Huber loss is a robust, normalized training objective; evaluation separately reports NRMSE. Gradients from Eq. (8) flow through the mixture weights, the learned corrections and their shared encoder. Analytic support operators need no learned weights or surrogate risk labels. Nonnegativity, exact support retention and the optional envelope reading provide structural constraints; no additional physics penalty is included in this stage-2 objective.

Residual initialization followed by joint fitting. Stage 1 trains the same tokenizer/encoder and a residual prediction head using $[b + \sigma \odot r]_+$, before introducing the correction bank. Stage 2 loads that encoder and head, initializes the gate, and updates all network parameters jointly. The new gate output projection is initially zero; its priors are zero for the anchor/neural branches and $-\gamma$ for the eight analytic corrections. We use $\gamma = 8$ for public rules and $\gamma = 2$ for the other domains, with τ initially 0.25. This suppresses analytic corrections before routing is learned while retaining gradients to the pretrained residual. A randomly initialized residual head has zero output, but the warm-started head is already trained; the full stage-2 model is therefore not claimed to start exactly at the anchor. AdamW uses learning rate 2×10^{-4}, weight decay 10^{-4} and unit-norm gradient clipping for 24,000 public-domain or 12,000 reference/transfer stage-2 updates. Validation ablations support residual initialization followed by joint optimization; Appendix E reports these comparisons, training-budget and branch-prior sensitivity, and anchor-only controls.

3.7 Inference, transfer and inspectable deployment

Inference uses the same support construction, tokenization, encoder and two heads without accessing query labels or taking gradient steps. Each trained member predicts the whole supplied query set in one table forward pass; three independently seeded members are averaged in physical units. Observed entries are then copied back exactly. Source-only transfer uses this same graph and source training scales, conditioning on the target table's available observations without target-domain optimization. The explicit bank exposes branch use and permits evaluated pruning by omitting operators and masking their logits; this also changes agreement features, so accuracy must be rechecked. The public reduced-bank diagnostic changes NRMSE by at most 0.5% relative (Appendix G). Each ensemble member has 105,604 parameters for $K = 2$ or 106,384 for $K = 3$.

4 Experiments

4.1 Dataset construction

We assemble a capacitance prediction benchmark from 1,713 public rule tables across four technologies and a finite-element reference with a separate 27-table evaluation collection; fitting uses 16 earlier reference tables. The public tables provide one-dimensional geometry variation; the reference adds two-dimensional width–distance grids and an ASAP7 source-only transfer setting. We construct support–query episodes by withholding responses under random, contiguous-block and boundary-extrapolation masks, retaining query coordinates and reserving withheld values for supervision or scoring. This construction tests prediction as the available observations and their geometric coverage change. Public rules provide extraction references, while the numerical grids provide independently solved capacitances. Appendix H details the data sources, grid construction, cohort sizes, splits, masking procedure and separation from the downstream audit.

4.2 Experimental setup

4.2.1 Baselines

We compare the capacitance query-prediction step, following the PEX-oriented baseline definition: commercial multilinear interpolation at the table's dimension (**CMI-1D** or **CMI-2D**; CMI-4D requires four independent axes and is inapplicable here, so it is omitted); an **OpenRCX-style** separation-axis piecewise linear rule; and general numerical and statistical interpolators (**PCHIP**,

Table 1: **Capacitance prediction at 30% missingness.** Mean-mask macro NRMSE (↓); bold: best method. CMI: 1D for public rules, 2D for FEM. Rnd./Ext.: random/boundary extrapolation.

	Public rules (rank 1)					FEM reference (rank 2)				
	In-support		Out-of-support			In-support		Out-of-support		
Method	Rnd.	Block	Rnd.	Block	Ext.	Rnd.	Block	Rnd.	Block	Ext.
CMI (rank-matched)	0.1286	0.2812	0.5531	0.9546	1.1867	0.0495	0.1502	0.1344	0.9790	0.6404
OpenRCX-LI	0.1286	0.2812	0.5531	0.9546	1.1867	0.0450	0.1597	0.2220	0.9767	0.6691
PCHIP	0.1070	0.1891	0.5531	0.9546	1.1867	0.0389	0.0997	0.2220	0.9767	0.6691
IDW	0.1911	0.2733	0.6235	1.5155	1.9906	0.1337	0.9631	3.7321	5.2144	3.8558
RBF-GP	0.4011	0.2490	0.9120	3.5596	4.6800	0.0116	2.8414	1.0237	18.3431	17.6758
Ours	**0.0803**	**0.0678**	**0.0910**	**0.1391**	**0.1620**	**0.0103**	**0.0366**	**0.0895**	**0.1203**	**0.1137**
Error reduction vs. CMI	37.6%	75.9%	83.5%	85.4%	86.3%	79.2%	75.6%	33.4%	87.7%	82.2%

IDW, **RBF-GP**). Masked characterization removes grid nodes, so a bracketing cell can be incomplete. Each multilinear rule therefore brackets the query between the nearest *observed* nodes on every axis, which is the natural generalization of the rule to a sparse table, and reports the width of the cell it had to use. A corner of that cell that is itself unobserved is filled by interpolating along whichever axis brackets it, which on a complete grid reduces to the textbook rule exactly (verified to 4×10^{-16} on affine fields at ranks 1, 2 and 4). The same missing-corner procedure serves every baseline, every baseline that cannot answer falls back to the same floor, and all methods are scored on one population, so none benefits from abstaining. Appendix E.1 compares matched-budget **ungated residual** and **binary-gated Transformers**; Appendix E.2 adds a validation-selected **local affine predictor** that permits extrapolation.

4.2.2 Regimes

We report interpolation and extrapolation relative to current support separately. A query is *in-support* if it lies inside the observed convex hull in its affine span; it is *out-of-support* otherwise. The evaluated multilinear controls use boundary clamping for queries beyond their observed range. The support partition of the protocol already draws this line geometrically; we collapse its five regions into these two, and aggregate their error and truth-energy accumulators with the same metric recipe.

4.2.3 Metrics and protocol

Masks are **random**, **contiguous-block** and **boundary-extrapolation** at 10/30/50/70% missingness with **five mask seeds**, plus every applicable whole-width fold. The random/block/extrapolation surface comprises 60 populated cohort–regime– protocol–rate cells and is reported in Appendix I.

$$V_m = \operatorname*{mean}_{(g,k)\,:\,\mathrm{sy}_{g,k}>0} \sqrt{\frac{\sum_{x\in(g,k)} (\hat{y}_k(x) - y_k(x))^2}{\sum_{x\in(g,k)} y_k(x)^2}}, \tag{9}$$

Reported scores are the **mean-mask macro NRMSE**, computed exactly as follows. Every scored query contributes its squared error and the square of its truth to one accumulator cell keyed by (method, training seed, process, family, protocol, missing rate, mask seed, support region), per capacitance component. Within one mask seed the score is given by Eq. (9), a flat mean over every defined (process/family group, component) pair of a pooled normalized root-mean-square error, and the reported score is the mean of V_m over the mask seeds. A cell with no component of positive truth energy is absent rather than zero. An inapplicable baseline proposal uses the shared fallback; all methods are therefore scored on the **same query population**. Public paired intervals resample associated groups with **2,000 replicates**, keeping every mask of a group in the same replicate. Configuration is chosen on the **validation cohort only**: among the candidate configurations, the one whose *largest* validation cell is smallest, over both regimes, both local missingness protocols and all four missing rates. The confirmation cohorts are opened once, afterwards, with that choice already fixed.

4.3 Prediction within and beyond observed support

Accurate interpolation and stronger extrapolation. Table 1 combines all five populated conditions per cohort. On public rules, in-support NRMSE is 0.0803/0.0678 for random/block masks, 25.0%/64.2% below the strongest external baseline (PCHIP). Beyond observed support, errors fall by 84–86% relative to CMI-1D. Here the anchor equals CMI-1D, making the improvement attributable to the learned correction. Paired bootstrap intervals exclude zero for both in-support comparisons against PCHIP and all three out-of-support comparisons against CMI (Appendix I.3). On the independent FEM reference, our method beats every external baseline in all five condi-

tions. The clearest advantage is under block missingness: in-support/out-of-support error falls by 75.6%/87.7% relative to CMI-2D.

4.4 Source-only transfer across material stacks

Table 2 tests whether support-conditioned corrections remain useful with frozen source parameters. On ASAP7, block-mask in-support NRMSE is 0.0531 versus 0.0974 for the strongest external baseline; out-of-support block and extrapolation errors fall by 85.8% and 83.0% relative to CMI. These improvements support the design choice to condition routing on the observations available in each target table. RBF-GP wins in-support random missingness on ASAP7. Across Tables 1–2, we improve on CMI in 15/15 conditions and the strongest numerical/statistical comparator listed there in 14/15. Appendix E.2 additionally tests a validation-selected numerical predictor that permits extrapolation. Matched-budget controls in Appendix E.1 favor full routing over the ungated residual in 14/15 conditions and the binary gate in 13/15, with larger gains on the FEM reference than on public rules. The complete 10–70% missingness surface yields 60/60 improvements over CMI (Appendix I).

Table 2: **Source-only transfer to ASAP7, 30% missingness.** Mean-mask macro NRMSE (↓); bold: best; underline: ours second-best.

	To ASAP7 (8 tables)				
	In-support		Out-of-support		
Method	Rnd.	Block	Rnd.	Block	Ext.
CMI-2D	0.0488	0.1473	0.1330	0.9583	0.6297
OpenRCX-LI	0.0444	0.1554	0.2173	0.9562	0.6574
PCHIP	0.0384	0.0974	0.2173	0.9562	0.6574
IDW	0.1331	0.9024	3.1647	4.7812	3.5543
RBF-GP	**0.0113**	2.5309	0.8194	16.0145	15.5718
Ours	<u>0.0159</u>	**0.0531**	**0.0877**	**0.1358**	**0.1073**
Reduction vs. CMI	67.4%	64.0%	34.1%	85.8%	83.0%

4.5 Downstream RC/STA fidelity on fixed routed layouts

We insert query predictions into the masked rule tables used by OpenRCX and rerun extraction and STA, keeping the routed database, constraints and toolchain fixed. Golden replay reproduces the original parasitics, and exports preserve every unmasked line. The audit rewrites 84 tables, including 55 from the original training split; the public ensemble was therefore retrained with all 84 excluded from training, validation and test. Public confirmation results and this audit use that held-out model family on distinct table populations.

Table 3 reports the capacitance and timing results for each evaluated condition. Our predictions improve capacitance NRMSE and worst-slack fidelity in all 16 cells, by factors of 1.8–13.9 and 2.1–56, respectively. Nonzero TNS deviations improve by factors of 2.1–25. For GF180/IBEX at 50% missingness, capacitance NRMSE falls from 0.0199 to 0.0061, worst-slack deviation from 0.1328 to 0.03701 ns, and TNS deviation from 168.6 to 50.22 ns. Since the anchor reproduces Linear, this downstream improvement also isolates the learned correction.

Endpoint-level replay on GF180/IBEX provides a more detailed check. For the preselected seed 0, slack MAE falls by 55.1%, with smaller error at 2,037 of 2,039 constrained endpoints. Seed 1 also improves critical-path agreement, while seed 2 has slightly higher MAE despite lower RMSE. Figure 3 shows the seed-0 spatial maps alongside Table 3; Appendix J.1 reports all three seeds, and Appendix J.2 extends the spatial audit to other layouts. These results establish fidelity to full-table extraction on unchanged layouts; they do not imply timing closure or faster circuits.

5 Conclusion

We study capacitance prediction conditioned on sparse physical observations, with particular attention to queries beyond their geometric support. A shared Transformer learns neural corrections and component-wise routing among numerical and neural proposals relative to an observation-derived anchor. The method improves over rank-matched CMI in all 15 headline conditions. Matched-budget learned controls show larger gains from full routing on the FEM reference than on public rules, with two transfer conditions favoring a binary gate. Writing predictions into extraction rules lowers capacitance and worst-slack deviations in all 16 downstream settings. These results support observation-conditioned residual prediction and a condition-dependent benefit of hybrid routing; broader extrapolation ranges and other physical responses remain to be tested. Appendix K discusses the implications and limits of the empirical evidence.

References

Mohamed Saleh Abouelyazid, Sherif Hammouda, and Yehea Ismail. Accuracy-based hybrid parasitic capacitance extraction using rule-based, neural-networks, and field-solver methods. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 41(12):5681–5694, 2022. doi: 10.1109/TCAD.2022.3161199.

Table 3: **Downstream capacitance and timing fidelity.** Errors against full-table extraction; WS/TNS: maximum absolute deviation across three seeds. Bold: lower error.

Process	Design	Mask	Method	Cap. NRMSE	$\lvert\Delta\text{WS}\rvert$ (ns)	$\lvert\Delta\text{TNS}\rvert$ (ns)
asap7	AES	30%	Linear	0.0185	0.000155	0.003715
			Ours	**0.0025**	**0.00003533**	**0.000235**
		50%	Linear	0.0187	0.0005771	0.003178
			Ours	**0.0027**	**0.0001836**	**0.0009669**
	IBEX	30%	Linear	0.0183	0.009746	0.03004
			Ours	**0.0024**	**0.0009348**	**0.005607**
		50%	Linear	0.0213	0.01083	0.03004
			Ours	**0.0022**	**0.0001934**	**0.001202**
gf180	AES	30%	Linear	0.0018	0.00191	0.344
			Ours	**0.0009**	**0.00074**	**0.153**
		50%	Linear	0.0222	0.05479	10.05
			Ours	**0.0068**	**0.01525**	**2.875**
	IBEX	30%	Linear	0.0014	0.00441	5.54
			Ours	**0.0008**	**0.00214**	**2.68**
		50%	Linear	0.0199	0.1328	168.6
			Ours	**0.0061**	**0.03701**	**50.22**
ihp-sg13g2	AES	30%	Linear	0.0175	0.005925	0
			Ours	**0.0016**	**0.001105**	0
		50%	Linear	0.0353	0.04597	0
			Ours	**0.0025**	**0.002026**	0
	IBEX	30%	Linear	0.0062	0.01169	0
			Ours	**0.0006**	**0.001394**	0
		50%	Linear	0.0198	0.04383	0
			Ours	**0.0018**	**0.004394**	0
sky130hs	AES	30%	Linear	0.0054	0.00169	0.003441
			Ours	**0.0007**	**0.0006721**	**0.001301**
		50%	Linear	0.0182	0.01598	0.02583
			Ours	**0.0021**	**0.00267**	**0.005426**
	IBEX	30%	Linear	0.0046	0.003377	0.6307
			Ours	**0.0006**	**0.000625**	**0.1274**
		50%	Linear	0.0141	0.01304	3.592
			Ours	**0.0022**	**0.003031**	**0.7839**

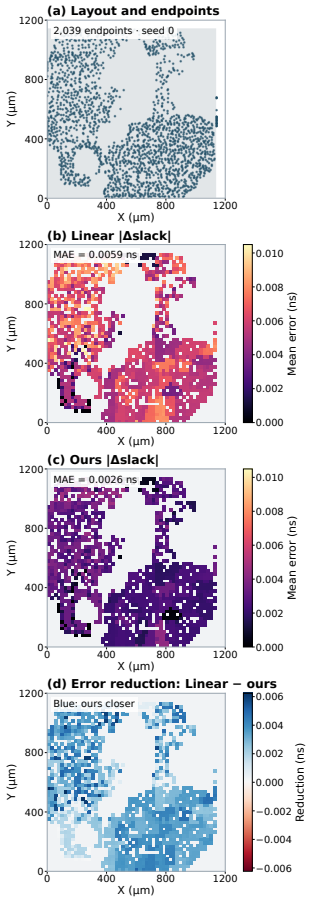


Figure 3: **Spatial timing fidelity.** GF180/IBEX, 50% block mask, seed 0. Top to bottom: layout, Linear error, ours, error reduction. Gray: empty bins; blue: improvement.

F. N. Fritsch and J. Butland. A method for constructing local monotone piecewise cubic interpolants. *SIAM J. Sci. Stat. Comput.*, 5(2):300–304, 1984.

Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and S. M. Ali Eslami. Conditional neural processes. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1704–1713. PMLR, 2018.

C. Geuzaine and J.-F. Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *Int. J. Numer. Methods Eng.*, 79(11):1309–1331, 2009.

Kairong Guo et al. PCT-Cap: Point cloud transformer for accurate 3d capacitance extraction. In *2024 2nd International Symposium of Electronics Design Automation*, 2024. doi: 10.1109/ISEDA62518.2024.10617673.

T. Gustafsson and G. D. McBain. scikit-fem: A Python package for finite element assembly. *J. Open Source Software*, 5(52):2369, 2020.

J. Huang, H. R. Rodriguez, D. Yang, Z. Ye, Y. Lin, and W. Yu. AttentionCap: Transformer based capacitance matrix learning toward full-chip extraction. *arXiv:2606.08161*, 2026.

X. Jiang, Z. Chai, Y. Zhao, Y. Lin, R. Wang, and R. Huang. CircuitNet 2.0: An advanced dataset for promoting machine learning innovations in realistic chip design environment. In *Proc. ICLR*, 2024.

Ryosuke Kasai, Toshiki Kanamoto, Masashi Imai, Atsushi Kurokawa, and Koutaro Hachiya. Neural network-based 3d ic interconnect capacitance extraction. In *2019 2nd International Conference on Communication Engineering and Technology*, pp. 168–172, 2019. doi: 10.1109/ICCET.2019.8726919.

Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, S. M. Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. Attentive neural processes. In *International Conference on Learning Representations*, 2019.

Lihao Liu, Fan Yang, Li Shang, and Xuan Zeng. GNN-Cap: Chip-scale interconnect capacitance extraction using graph neural network. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 43(4):1206–1217, 2024. doi: 10.1109/TCAD.2023.3331942.

Keith Nabors and Jacob K. White. FastCap: A multipole accelerated 3-d capacitance extraction program. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 10 (11):1447–1459, 1991. doi: 10.1109/43.97624.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. RouteLLM: Learning to route LLMs with preference data. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=8sSqNntaMr.

OpenROAD Project. Parasitics extraction (OpenRCX). https://openroad.readthedocs.io/en/latest/main/src/rcx/README.html, n.d. Accessed September 25, 2026.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Clemens Rosenbaum, Tim Klinger, and Matthew Riemer. Routing networks: Adaptive selection of non-linear functions for multi-task learning. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=ry8dvM-R-.

S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *Proc. ICLR*, 2020.

C.-Y. Tsai, S. Kim, and S.-K. Lim. Capacitance extraction via machine learning with application to interconnect geometry exploration. In *Proc. ICCAD*, 2025a.

Jiun-Cheng Tsai, Hsuan-Ming Huang, Wei-Min Hsu, Pei-Ting Lee, Jen-Hang Yang, Heng-Liang Huang, Yen-Ju Su, and Charles H.-P. Wen. ResCap: Fast-yet-accurate capacitance extraction for standard cell design by physics-guided machine learning. In *Proceedings of the 30th Asia and South Pacific Design Automation Conference*, pp. 1243–1250, 2025b. doi: 10.1145/3658617.3697737.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

D. Yang, H. Li, W. Yu, Y. Guo, and W. Liang. CNN-Cap: Effective convolutional neural network-based capacitance models for interconnect capacitance extraction. *ACM Trans. Design Autom. Electron. Syst.*, 28(4):56:1–56:22, 2023. doi: 10.1145/3564931.

Wenjian Yu, Mingye Song, and Ming Yang. Advancements and challenges on parasitic extraction for advanced process technologies. In *Proceedings of the 26th Asia and South Pacific Design Automation Conference*, pp. 841–846, 2021. doi: 10.1145/3394885.3431626.

A Related Work

Prediction from observation context. Conditional Neural Processes learn to predict from observed input–output pairs and can use a new context set at test time (Garnelo et al., 2018). Attentive Neural Processes let each query attend to relevant context points (Kim et al., 2019). Our model shares the use of observations as inference inputs, producing deterministic capacitance estimates conditioned on geometry, material context and available responses. Its specific design integrates a support-derived anchor, numerical correction branches and component-wise routing. We evaluate extrapolation relative to each episode's observed hull as well as interpolation within it. The contribution lies in this architecture and its capacitance evaluation; observation-conditioned prediction itself is established by these prior works.

Learned routing and adaptive combinations. Routing Networks learn input-dependent compositions of function blocks (Rosenbaum et al., 2018). RouteLLM learns to select between language models to balance response quality and inference cost (Ong et al., 2025). These works establish learned routing as a way to adapt computation to an input. Our router instead assigns soft weights to corrections for each query and capacitance component. Its candidates combine deterministic numerical estimators with two readings of a shared neural residual, all relative to one anchor. All applicable branches are evaluated in the full model; routing combines their outputs rather than providing conditional-computation savings. Joint training uses prediction loss without labels identifying a preferred branch.

Numerical estimators and residual correction. PCHIP provides shape-preserving one-dimensional interpolation (Fritsch & Butland, 1984), and Gaussian-process regression provides kernel-based prediction from observed values (Rasmussen & Williams, 2006). These estimators supply explicit numerical proposals in our model; their formulas and boundary conventions appear in Appendix D. In capacitance modeling, ResCap learns residuals to a physics-guided initial estimate (Tsai et al., 2025b). Our numerical anchor and analytic proposals are reconstructed from the current observations. The network learns both a residual and its combination with these proposals under changing support. The local envelope provides an observation-based inductive bias; it is not a guaranteed physical bound, and it does not constrain the neural branch outside support.

Learned capacitance models and extraction. OpenRCX reuses process-specific rules derived from characterized wire patterns (OpenROAD Project, n.d.). CNN-Cap learns capacitance from convolutional geometry representations (Yang et al., 2023). AttentionCap learns capacitance matrices with a customized Transformer and process-node embeddings, and demonstrates adaptation to an unseen node through fine-tuning (Huang et al., 2026). Our setting makes the current geometry–capacitance observations explicit inputs and keeps model parameters fixed during target inference. The benchmarks compare predictors supplied with the same observed support and query coordinates. They do not establish performance relative to geometry-based extractors on those methods' original tasks. Export to OpenRCX separately tests how query prediction errors affect extraction and timing on unchanged layouts.

B Structural families and geometric context

The four-way structural-family indicator in Section 3.2 identifies the arrangement of the target conductor and its surrounding conductors. Figure 4 illustrates the four configurations used to interpret these labels. The drawings are schematic cross sections: orange marks the target, blue marks neighboring conductors in the target layer, green marks a conductor in an upper layer, and gray marks grounded context. Dielectric regions and outer simulation boundaries are omitted for clarity. The family name specifies a relative arrangement, while dimensions, layer indices and material properties are supplied separately by geometry and context.

OVER. The target and its same-layer neighbors lie above a grounded context conductor. The configuration includes lateral interaction between same-layer conductors and coupling to the lower context. "Over" refers to the target's position relative to this lower context.

UNDER. The target and its same-layer neighbors lie below a grounded context conductor. This configuration retains lateral interactions and places the additional context above the target. "Under" refers to this relative vertical ordering.

OVERUNDER. Grounded context conductors are present both below and above the target layer. The target therefore interacts with its lateral neighbors and with context on both sides vertically. The upper and lower context layers are distinct parts of the arrangement.

DIAGUNDER. An upper-layer signal conductor is laterally displaced from the target, giving both horizontal and vertical separation. Here "diagonal" describes their relative positions in cross section,

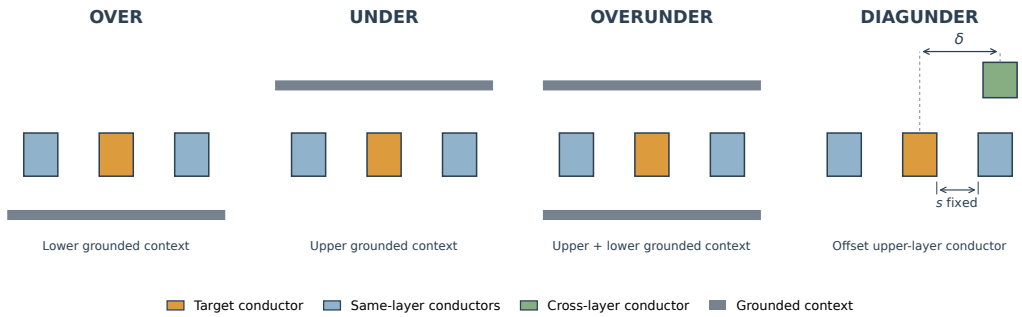


Figure 4: **Four structural families.** OVER places the target above lower grounded context; UNDER places it below upper grounded context; OVERUNDER includes both. DIAGUNDER includes an upper-layer conductor with lateral offset δ from the target. In the reference configuration, its same-layer gap s is fixed while the cross-layer offset varies. Colors identify conductor roles; the cross sections are not to scale and do not depict the complete simulation domain.

not a slanted conductor. In the illustrated reference family, the same-layer gap remains fixed as the cross-layer offset changes. This distinguishes offset cross-layer coupling from the planar upper context shown for UNDER.

Role in conditioning. The indicator tells the shared network which arrangement a query belongs to; it does not encode a process identity or replace the observed capacitances. The target and context-layer indices, continuous geometry and available material descriptions supply the remaining physical context. The same family can occur in different process stacks with different dimensions and response values. These four schematics describe the families used in this study and do not exhaust commercial extraction pattern categories.

C Network specification and operator settings

Table 4 specifies each member of the released ensembles. Public models use $K = 2$; the reference and transfer models use $K = 3$. Each reported domain has its own three-seed ensemble. The following details complete the equations in Section 3.

Table 4: Exact per-member network dimensions and stage-2 settings.

Quantity	Public ($K = 2$)	Reference/transfer ($K = 3$)
Geometry/material token width $17 + 2K$	21	23
Shared embedding / encoder width	64	64
Encoder layers / heads / head width	2 / 4 / 16	2 / 4 / 16
Feed-forward hidden width	256	256
Residual head	$64 \to 2$	$64 \to 3$
Support/context features $23 + d + 4K$	34	38
Agreement features $3K$	6	9
Router input / hidden / output	$104/32/22$	$111/32/33$
Correction slots E	11	11
Trainable parameters	105,604	106,384
Initial prior gap γ / temperature τ	$8/0.25$	$2/0.25$
Stage-2 updates / dropout	$24{,}000/0$	$12{,}000/0$

Support descriptors. The descriptor s_i concatenates: observation density, log nearest distance, log bracketing extent, inside-support indicator and K log envelope widths; a five-way indicator (outside, missing width, dense rank-one, dense higher-rank, sparse/degenerate); support rank, log typical neighbor distance and aligned-width availability; d directional coverage fractions; $3K$ local variation, slope and slope-difference channels; and the 11 context fields. Distances use observed-range-normalized coordinates. Local response statistics use up to the three nearest observed values divided by σ. The typical distance is the median nearest-neighbor distance among distinct support points; the dense indicator compares local extent with twice that distance. Thus no experimental mask name or hidden response is supplied to the gate.

Analytic operators. The aligned-width operator requires two observations with identical non-width coordinates to the query and widths bracketing it; its weight is the ordinary linear fraction

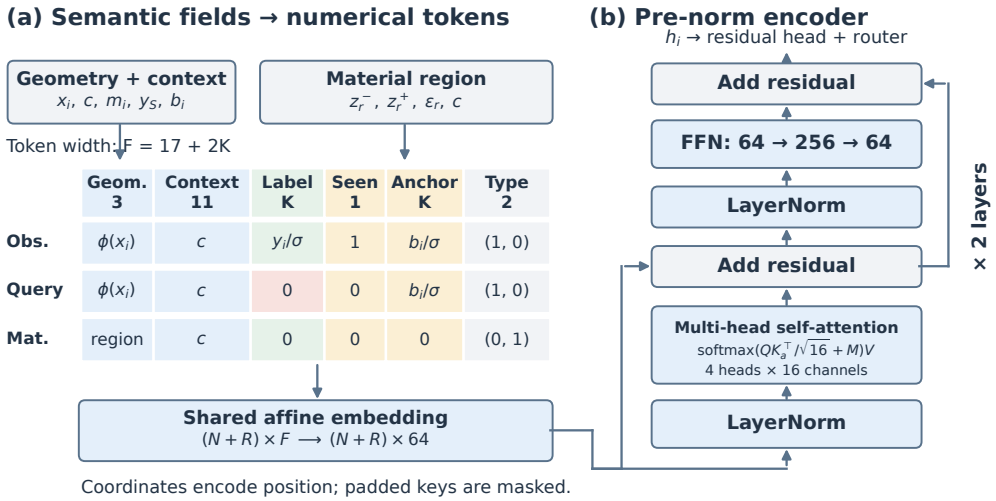


Figure 5: Structured tokenization and the shared encoder. Each geometry row contains six feature blocks, including an explicit observation indicator; $\phi(x_i) = \log(1 + x_i/\xi)$. A query retains its geometry and support-derived anchor while its label field is zero. Material rows encode region extent and permittivity. All valid tokens participate in attention; two pre-normalized encoder blocks produce one hidden state per token. Hidden query labels enter only the training loss.

between those widths. The six GP operators use reduced support coordinates a_j, query a_i, and

$$f_{\lambda,\rho}(x_i) = \left[\bar{y} + k_i^\top (K_\lambda + \rho I)^{-1}(Y_S - \mathbf{1}\bar{y}^\top)\right]_+, \qquad (K_\lambda)_{jv} = \exp\left(-\frac{\|a_j - a_v\|^2}{2L_\lambda^2}\right), \quad (10)$$

where $\bar{y}$ is the observed component mean, k_i is the corresponding query-to-support kernel vector, and L_λ is λ times the median positive pairwise support distance. We use $\lambda \in \{0.5, 1, 2\}$ and $\rho \in \{10^{-6}, 10^{-4}\}$. Zero-rank support yields a constant proposal; exact observations are restored. PCHIP is evaluated on sorted rank-one support with the query clamped to its observed interval. Analytic operators are support-conditioned computations, not extra pretrained networks.

Numerical conventions. All network-side anchors, residuals, envelopes and agreement statistics are expressed in units of σ; Eq. (1) writes their physical-unit equivalent. The agreement standard deviation uses a 10^{-12} variance floor for finite derivatives. Invalid branches receive logit -10^9 and contribute nothing to agreement statistics. The model's learned keys are the rows of the router output projection; descriptor-dependent keys are evaluated separately in Appendix F. This distinguishes the reported network from those architectural variants.

Attention and padding. For each head, Q, K_a and V are learned projections of the normalized token states, and

$$A = \mathrm{softmax}\left(QK_a^\top/\sqrt{16} + M\right), \qquad O = AV. \quad (11)$$

$M_{ij} = -\infty$ for a padded key and zero otherwise. Head outputs are concatenated and projected back to width 64; the feed-forward activation is ReLU. The reported stage-2 models use zero dropout.

Routing sensitivity. The coupling can be seen directly from the router's local sensitivity. Holding branch proposals fixed, when the pre-projection output is positive,

$$\frac{\partial \hat{y}_{i,k}}{\partial a_{i,e,k}} = \frac{\pi_{i,e,k}}{\tau}\left(\Delta_{i,e,k} - \sum_{e' \in \mathcal{E}_i} \pi_{i,e',k}\Delta_{i,e',k}\right). \quad (12)$$

An update therefore depends on how a proposal changes the current mixture, weighted by the final prediction error through the chain rule. This trains routing for the actual query-prediction task without requiring a separate target that names the best branch.

Implementation entry points. In the released study, `unified/basis.py` constructs support and analytic corrections; `unified/episodes.py` builds tokens and padded batches; `unified/model.py` implements the encoder, residual head, gate and priors. Training and warm-start logic are in `unified/train.py`; inference, observed-value restoration and pruning

are in `unified/score.py`. The `cmi.py`, `score_baselines.py`, `regime.py`, `aggregate.py` and `paired.py` modules provide baselines and metrics. The selection, branch and runtime diagnostics are in `scripts/final_selection.py`, `score_elements.py`, `diagnose_cells.py` and `benchmark_interpolation.py`; `run_sta_visualization.py` reproduces the endpoint audit.

D CLASSICAL ESTIMATION

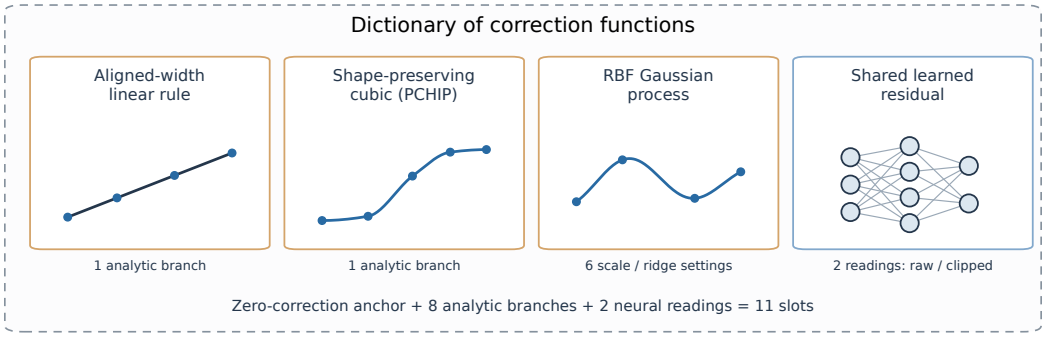


Figure 6: Dictionary of correction functions (schematic). The analytic branches comprise one aligned-width linear rule, one shape-preserving cubic interpolant, and six RBF-GP predictors. The shared neural residual supplies raw and locally envelope-constrained readings. Together with the zero-correction anchor, these form 11 slots. Analytic predictions enter as $\Delta_{i,e} = f_e(x_i; S, y_S) - b_i$; inapplicable branches are masked before component-wise routing. Dots denote observed support; curves illustrate operator behavior, not measured results.

We state the numerical estimators using the observed index set S and component index k. These explicit controls do not identify proprietary commercial interpolation algorithms. In particular, the simplex-linear anchor below is distinct from the multidimensional CMI baseline.

Linear interpolation and the aligned-width rule. For a one-dimensional query bracketed by observed coordinates $x_a < x_b$,

$$b_k(x) = (1-t)y_{a,k} + ty_{b,k}, \qquad t = \frac{x - x_a}{x_b - x_a}. \tag{13}$$

Outside the observed interval, Linear extends the nearest endpoint value. The aligned-width correction uses Eq. (13) with width as its coordinate, but is applicable only when both bracketing observations match all non-width query coordinates. Its residual is $f_{\mathrm{width,k}}(x) - b_k(x)$. For support rank $r > 1$, let $a(x)$ denote the support-normalized independent coordinates and $v_0, \ldots, v_r$ the vertices of an enclosing simplex. Then

$$b_k(x) = \sum_{q=0}^{r} \alpha_q y_{v_q,k}, \qquad a(x) = \sum_{q=0}^{r} \alpha_q a(x_{v_q}), \qquad \sum_{q=0}^{r} \alpha_q = 1, \quad \alpha_q \geq 0. \tag{14}$$

Outside the reduced support hull, or if triangulation fails, the anchor copies the nearest observed value in these coordinates. Rank-zero support gives a constant estimate.

Shape-preserving cubic interpolation (PCHIP). PCHIP (Fritsch & Butland, 1984) uses shape-preserving support slopes $d_{j,k}$ in a piecewise cubic Hermite interpolant. On $[x_a, x_b]$, with $\Delta x = x_b - x_a$ and t as in Eq. (13),

$$p_k(t) = h_{00}(t)y_{a,k} + h_{10}(t)\Delta x\, d_{a,k} + h_{01}(t)y_{b,k} + h_{11}(t)\Delta x\, d_{b,k}, \tag{15}$$

where

$$\begin{aligned} h_{00}(t) &= 2t^3 - 3t^2 + 1, \quad & h_{10}(t) &= t^3 - 2t^2 + t, \\ h_{01}(t) &= -2t^3 + 3t^2, & h_{11}(t) &= t^3 - t^2. \end{aligned} \tag{16}$$

The slopes preserve the local monotonicity of the observed data. Our PCHIP control clamps queries to the observed interval, extending endpoint values outside it; the network branch uses sorted rank-one, single-width support.

Inverse-distance weighting (IDW). The IDW baseline predicts

$$f_{\mathrm{IDW},k}(x) = \frac{\sum_{j\in S} w_j(x) y_{j,k}}{\sum_{j\in S} w_j(x)}, \qquad w_j(x) = \max\big(\|a(x) - a(x_j)\|, 10^{-12}\big)^{-2}. \tag{17}$$

IDW is an external baseline, not an additional slot in the correction dictionary.

RBF Gaussian-process prediction. The mean-centered RBF-GP estimator (Rasmussen & Williams, 2006) is

$$f_{\mathrm{GP},k}(x) = \bar{y}_k + \mathbf{k}(x,S)^\top (\mathbf{K} + \rho I)^{-1}(y_{S,k} - \bar{y}_k \mathbf{1}), \qquad \bar{y}_k = \frac{1}{|S|}\sum_{j\in S} y_{j,k}, \tag{18}$$

with $\mathbf{K}_{jv} = \kappa(x_j, x_v)$, $\mathbf{k}(x,S)_j = \kappa(x, x_j)$, and

$$\kappa(x,x') = \exp\left(-\frac{\|a(x) - a(x')\|^2}{2\ell^2}\right). \tag{19}$$

Distances use support-normalized independent coordinates. The network's six GP branches use $\ell = \lambda D_S$, where D_S is the median positive pairwise support distance, $\lambda \in \{0.5, 1, 2\}$ and $\rho \in \{10^{-6}, 10^{-4}\}$. Their proposals are projected to nonnegative values before subtracting the anchor, as specified in Appendix C. All support computations exclude query responses, and observed entries are restored exactly at inference.

E WHY RESIDUAL INITIALIZATION AND JOINT ROUTING MATTER

Table 5: **Design choices behind neural–numerical routing.** Validation NRMSE (↓). Each panel is a separate controlled comparison, retaining its recorded validation protocol macro rather than the regime-specific confirmation scores of Tables 1–2. Panel A controls architecture, data and budget; B varies stage-2 updates; C varies the initial prior of the eight analytic corrections.

Comparison / setting	Cohort	Random	Block	Extrap.
A. Training stages; matched architecture, data and budget				
Anchor only	Public val.	0.2195	0.5576	0.9125
Joint training from scratch	Public val.	0.1827	0.2955	0.3767
Warm start; train gate only	Public val.	0.1215	0.2050	0.2755
Warm start; train jointly (ours)	Public val.	0.1000	0.1930	0.2755
B. Stage-2 budget; public validation				
3000 steps	Public val.	0.0832	0.1388	0.1945
6000 steps	Public val.	0.0756	0.1297	0.1864
12000 steps	Public val.	0.0734	0.1170	0.1632
24000 steps	Public val.	0.0712	0.1098	0.1532
C. Analytic-branch initialization; matched budget				
Prior $\beta_e = -8$	Source val.	0.0156	0.0443	0.1297
Prior $\beta_e = -2$	Source val.	0.0088	0.0336	0.1156

Table 5 connects the training procedure to its motivation. A zero-initialized residual competes with immediately usable numerical operators, making joint training from scratch a weak starting point. Initializing the residual first reduces random/block/extrapolation error from 0.1827/0.2955/0.3767 to 0.1215/0.2050/0.2755 even with a frozen trunk; subsequent joint training reaches 0.1000/0.1930/0.2755. This supports learning a useful correction before learning how strongly to use it.

Longer schedules continue to improve the measured validation scores (panel B), without establishing gains beyond the tested budget. The initial analytic-branch prior also matters: on the FreePDK15 source-validation cohort for transfer to ASAP7, changing it from -8 to -2 lowers all three errors (panel C). Thus routing depends on both the available corrections and their accessibility during optimization; adding branches alone does not ensure they are used well.

Anchor-only control. On the FEM reference, the anchor-only control isolates the contribution of residual gating. It already improves on CMI outside support, but residual gating reduces out-of-support block/extrapolation error from 0.6542/0.5663 to 0.1203/0.1137. For out-of-support random masks, the anchor's 0.0606 is better than our 0.0895. These internal controls complement the training-stage ablation in Table 5; Tables 1 and 2 compare the full model with external methods.

Routing limitations. On ASAP7 out-of-support random masks, the anchor scores 0.0595 compared with our 0.0877, so the learned mixture does not always improve on its numerical reference. Descriptor-dependent keys give mixed validation results, and the validation-selected variant worsens three of five confirmation conditions. The tested variants and numerical-selector definition are retained in Appendix F.

Computational cost. The complete three-member ensemble costs about 16× CMI per query on 13-node public tables and 3–4× on 128-node grids on a shared, single-thread host. The larger query batch amortizes support construction. Removing eight little-used analytic branches on public rules changes NRMSE by at most 0.5% relative and reduces measured scoring time by 6.1×. This is a separate pruning diagnostic; all headline accuracy comparisons use the full bank. Appendix G reports the costs and session variation.

E.1 Matched-budget learned controls

The earlier direct, residual and binary-gated checkpoints each received only 2,000 updates, and the public versions predate the downstream exclusions. They therefore do not provide a matched-budget test of the reported model. We separately train two simpler predictors to distinguish the benefit of cross-table learning from that of routing:

$$\hat{y}_i^{\text{residual}} = [b_i + \sigma \odot r_i]_+, \tag{20}$$

$$\hat{y}_{i,k}^{\text{binary}} = [b_{i,k} + \alpha_{i,k}\sigma_k r_{i,k}]_+, \qquad 0 \le \alpha_{i,k} \le 1. \tag{21}$$

The first is an ungated residual Transformer with the same tokens, encoder and residual head. The second uses a component-wise two-slot softmax over the anchor and raw neural residual; its gate reads the hidden state and the same support/context features. Neither includes analytic correction branches or the clipped neural reading, and the binary gate has no bank-disagreement features. The latter is a newly trained two-branch model, distinct from inference-time pruning of the full model to three remaining slots. These comparisons test the utility of routing under the reported data and training budget; they do not establish that an ungated neural predictor cannot represent a similar correction. The full-versus-binary comparison changes the analytic bank, clipped reading and agreement features together; it does not separately identify the effect of each of these additions.

For each setting and seed, both controls reuse the full model's fitting tables, training scales and 2,000-step residual initialization. They retain the same family-balanced episode sampler, random seed, loss, AdamW settings, gradient clipping and stage-2 updates: 24,000 for public rules and 12,000 for reference/transfer. The last checkpoint is used without separate hyperparameter search for the controls; thus these are controls under the reported training recipe, not an exhaustive optimization of each smaller architecture. All reported predictions average three independently trained members. Each public member has 101,506 parameters for the ungated model, 104,809 for the binary model and 105,604 for the full model; the corresponding three-output models have 101,699, 105,196 and 106,384 parameters.

The new evaluations retain the headline 30% masks, five mask seeds, query indices, support partition and macro-NRMSE aggregation. Full-model scores are read from the original reported ensemble records. These supplementary comparisons use the existing confirmation cohorts; they are not presented as a newly blinded benchmark. Source-only transfer controls are fitted on eight FreePDK15 tables and use the inherited source-selected training recipe.

Overall and reference-domain comparisons. Across the 15 confirmation conditions, the full ensemble improves on the ungated residual in 14 and on the binary gate in 13. Its advantage is largest on the two-dimensional FEM reference: it improves all five conditions against both controls, reducing NRMSE by 18.1–73.3% relative to the binary gate. For example, in-support random NRMSE is 0.0103 for the full model, 0.0387 for the binary gate and 0.0529 for the ungated residual. These differences support the utility of the full hybrid design under the reference-domain training recipe, rather than attributing all gains to cross-table learning alone.

Transfer to ASAP7 is less uniform. Full routing improves on the ungated residual in all five conditions, but the binary ensemble is better on out-of-support block masks (0.1291 versus 0.1358) and extrapolation masks (0.1037 versus 0.1073). The full branch bank is consequently not a uniformly better choice across the evaluated settings.

Public-rule attribution. The full ensemble has lower point estimates than the binary gate in all five public conditions, but the relative improvements range from only 0.6% to 5.6%. The ungated residual has a lower point estimate for in-support block masks (0.0645 versus 0.0678), while the full model is better in the other four conditions. Much of the reduction relative to numerical interpolation is therefore already achieved by a support-conditioned residual Transformer.

Table 6: **Matched-budget learning and routing controls.** Confirmation NRMSE at 30% missingness. Each row is a three-member ensemble with the same training data, residual initialization and update budget within its setting. Bold marks the lowest value; underlining marks our second-best value in each condition.

	In-support		Out-of-support		
Method	Random	Block	Random	Block	Extrap.
Public rules					
Transformer + ungated residual	0.0875	**0.0645**	0.0965	0.1455	0.1675
Anchor/neural binary gate	0.0850	0.0701	0.0954	0.1420	0.1629
Full routing (reported ensemble)	**0.0803**	<u>0.0678</u>	**0.0910**	**0.1391**	**0.1620**
FEM reference					
Transformer + ungated residual	0.0529	0.0885	0.1336	0.1867	0.1761
Anchor/neural binary gate	0.0387	0.0560	0.1303	0.1479	0.1388
Full routing (reported ensemble)	**0.0103**	**0.0366**	**0.0895**	**0.1203**	**0.1137**
Transfer to ASAP7					
Transformer + ungated residual	0.0588	0.0868	0.1172	0.1482	0.1172
Anchor/neural binary gate	0.0432	0.0640	0.1124	**0.1291**	**0.1037**
Full routing (reported ensemble)	**0.0159**	**0.0531**	**0.0877**	<u>0.1358</u>	<u>0.1073</u>

Table 7 qualifies the small routing gains. The full model's paired intervals exclude zero against the binary gate in the two in-support conditions, but not in any of the three out-of-support conditions. Against the ungated residual, they exclude zero in favor of the full model for in-support random and out-of-support block masks. These are pointwise comparisons conditional on the trained ensembles, not evidence of a uniform advantage across training seeds or a guarantee that the full branch bank is necessary. Together with the pruning diagnostic, the public evidence supports a strong learned residual and more limited gains from the extra routing machinery.

Training-seed checks and interpretation. Replaying all original full-model members reproduces the reported ensemble NRMSEs exactly in all 15 conditions. Comparing corresponding individual seeds, the full model beats the ungated residual in 12/15 public, 15/15 reference and 13/15 transfer seed–condition pairs; against the binary gate the counts are 9/15, 13/15 and 13/15. These descriptive counts are not independent statistical tests. They also need not match ensemble win counts, because averaging predictions precedes computing ensemble NRMSE. The reference advantage is stronger than the public advantage, but neither seed counts nor pooled ensemble scores establish uniform dominance. In particular, these controls support a condition-dependent benefit of the full hybrid design, not the necessity of every analytic operator in every domain.

Table 7: **Paired public-group uncertainty for the routing controls.** Difference is full routing minus the named comparator, so negative favors the full model. Pointwise 95% intervals use 2,000 paired group-bootstrap samples, retaining all masks of each dependency group. The intervals condition on the trained ensembles and are not multiplicity-adjusted.

Regime / mask	Comparator	ΔNRMSE	Paired 95% interval
In-support / random	Ungated residual	-0.0072	[-0.0099, -0.0050]
In-support / random	Binary gate	-0.0047	[-0.0068, -0.0029]
In-support / block	Ungated residual	+0.0033	[-0.0009, +0.0065]
In-support / block	Binary gate	-0.0023	[-0.0045, -0.0010]
Out-of-support / random	Ungated residual	-0.0055	[-0.0117, +0.0005]
Out-of-support / random	Binary gate	-0.0044	[-0.0115, +0.0045]
Out-of-support / block	Ungated residual	-0.0064	[-0.0120, -0.0008]
Out-of-support / block	Binary gate	-0.0029	[-0.0079, +0.0017]
Out-of-support / extrapolation	Ungated residual	-0.0055	[-0.0111, +0.0006]
Out-of-support / extrapolation	Binary gate	-0.0009	[-0.0060, +0.0039]

E.2 An extrapolation-capable numerical control

To distinguish learned gains from endpoint-clamping conventions, we add a local affine predictor that continues its fitted trend beyond the support hull. Coordinates are normalized by the observed range and projected onto the independent axes of the observed support. For each query, the predictor fits an affine function to its m nearest observed points, using at least $r + 1$ points when available, where r is the support rank:

$$(a_i, B_i) = \arg\min_{a,B} \sum_{j \in N_m(i)} \|y_j - a - B(z_j - z_i)\|_2^2 + 10^{-8}\|B\|_F^2, \qquad \hat{y}_i = [a_i]_+. \tag{22}$$

The intercept is unpenalized; rank-zero support returns its mean response. The predictor preserves observed entries exactly. It does not clip predictions to endpoint values or a local response envelope.

We select $m \in \{2, 4, 8, 16\}$ by the mean of the five defined regime/protocol validation NRMSEs at 30% missingness, using the original five mask seeds. The public validation pool has the same downstream exclusions as the reported public model; the transfer validation pool contains FreePDK15 only. The selected counts are 4 for public rules and 8 for reference and transfer. Selection is recorded before scoring confirmation. Confirmation uses exactly the query indices, support regions, truth-energy accumulators and normalization scales of the headline comparisons.

Table 8: **Local affine extrapolation versus the support anchor and full model.** Confirmation NRMSE at 30% missingness; lower is better. The affine predictor can extrapolate outside support and its neighbor count is selected only on validation. Bold marks the lowest value; underlining marks our second-best value in each condition.

	In-support		Out-of-support		
Method	Random	Block	Random	Block	Extrap.
Public rules					
Support anchor	0.1286	0.2812	0.5531	0.9546	1.1867
Local affine (validation-selected)	0.2062	0.4324	0.4766	0.5588	0.6503
Full routing (reported ensemble)	**0.0803**	**0.0678**	**0.0910**	**0.1391**	**0.1620**
FEM reference					
Support anchor	0.0208	0.1286	**0.0606**	0.6542	0.5663
Local affine (validation-selected)	0.0389	0.1579	0.0621	0.3374	0.2376
Full routing (reported ensemble)	**0.0103**	**0.0366**	0.0895	**0.1203**	**0.1137**
Transfer to ASAP7					
Support anchor	0.0205	0.1246	0.0595	0.6374	0.5580
Local affine (validation-selected)	0.0381	0.1542	**0.0594**	0.3288	0.2364
Full routing (reported ensemble)	**0.0159**	**0.0531**	0.0877	**0.1358**	**0.1073**

The full model is better than this extrapolation-capable control in 13 of 15 conditions. On public out-of-support block and extrapolation masks, the comparison changes from 0.5588/0.6503 to 0.1391/0.1620. Thus the gains in these conditions do not depend solely on comparing against endpoint extension. However, local affine prediction is better on out-of-support random masks for both the FEM reference (0.0621 versus 0.0895) and transfer to ASAP7 (0.0594 versus 0.0877). The existing anchor failures therefore remain material; the learned mixture does not uniformly improve extrapolation.

F ROUTING DIAGNOSTICS AND ALTERNATIVE GATES

What the numerical branches provide. Table 9 retains the numerical-operator diagnostic, whose scope is narrower than a full-network oracle. The scoring script constructs the anchor and eight analytic corrections without loading neural predictions. Its label-informed selector chooses a single proposal per query by minimizing the sum of absolute errors in training-normalized component units; undefined corrections revert to the anchor in this diagnostic. It is evaluated using macro NRMSE, a different objective. Thus it is neither an NRMSE lower bound nor an oracle over all eleven branches or component-wise convex mixtures. Individual operators are scored only where applicable, so the best-operator column can also have a different population.

On public out-of-support random masks, the network's 0.0910 is well below the numerical selector's 0.4297, illustrating the contribution unavailable to this diagnostic.

Table 9: **Numerical-operator diagnostics at 30% random missingness.** NRMSE (↓). Best operator* is scored on its applicability subset; parentheses identify the operator. The label selector reads query truth, uses only numerical branches and is not a deployable method or a bound on the full network. Other columns use the full query population. Bold marks the lowest displayed NRMSE; underlining marks the second-lowest displayed Ours value. Populations and label access differ as described above.

Cohort	Regime	Best operator*	Label selector	CMI	**Ours**
Reference	out-of-support	0.0606 (anchor)	**0.0524**	0.1344	0.0895
Reference	in-support	0.0208 (anchor)	**0.0027**	0.0495	<u>0.0103</u>
ASAP7	out-of-support	0.0595 (anchor)	**0.0518**	0.1330	0.0877
ASAP7	in-support	0.0205 (anchor)	**0.0027**	0.0488	<u>0.0159</u>
Public rules	out-of-support	0.5531 (anchor)	0.4297	0.5531	**0.0910**
Public rules	in-support	0.1070 (PCHIP)	**0.0761**	0.1286	<u>0.0803</u>

Descriptor-dependent keys. We tested five-channel descriptor keys with pointwise or per-condition objectives and a six-channel variant whose envelope descriptor also covers outside-support queries. Table 10 groups all reported validation rows by cohort; absent variant/cohort combinations are not inferred. The descriptor-key variants give mixed results on the source-validation cohort for

transfer to ASAP7. On the reference and public cohorts, none of the tabulated variants improves over the reported gate. Boldface reflects numerical comparisons directly.

The source-validation minimax rule ranked a descriptor-key configuration first for transfer to ASAP7. Its one-time confirmation evaluation worsened three of five conditions, including out-of-support random NRMSE (0.107 versus 0.088). We retain the headline configuration fixed before these exploratory variants, and do not claim that validation selection identified this failure in advance.

Table 10: **Descriptor-key validation diagnostics at 30% missingness.** NRMSE (↓), grouped by cohort. Bold marks the lowest NRMSE among all gates in each cohort/condition; underlining marks our second-best value. All available rows are retained. The transfer row uses only FreePDK15 source-validation tables, not ASAP7 labels.

		In-support		Out-of-support		
Cohort	Gate / objective	Random	Block	Random	Block	Extrap.
public rules	5-channel / pointwise	0.0755	0.1042	0.1289	0.1881	0.2210
	5-channel / per-condition	0.0751	0.0682	0.1228	0.1472	0.1617
	reported gate	**0.0676**	**0.0648**	**0.1006**	**0.1293**	**0.1471**
2-D reference	5-channel / pointwise	0.0243	0.0537	0.0470	0.1867	0.1155
	5-channel / per-condition	0.0085	0.0315	0.0274	0.1452	0.1033
	reported gate	**0.0049**	**0.0314**	**0.0193**	**0.1120**	**0.0903**
Source → ASAP7	5-channel / pointwise	0.0207	0.0576	0.0301	0.1531	**0.1066**
	5-channel / per-condition	0.0060	**0.0294**	0.0150	**0.1208**	0.1177
	6-channel / per-condition	0.0187	0.0533	0.0357	0.1644	0.1375
	reported gate	**0.0051**	<u>0.0363</u>	**0.0127**	0.1733	0.1300

G LATENCY AND EVALUATED BRANCH PRUNING

Table 11 combines the original method comparison and the later platform breakdown. Both use interleaved execution on a shared host, with one BLAS thread and one PyTorch thread. One episode is one table's query set: six queries for the 13-node public workload and 64 for the 128-node reference workload. The full three-seed ensemble includes support construction, numerical branches and Transformer passes.

The platform session gives ratios of 15.77–16.41× CMI for public rules and 3.06× for the ASAP7 reference grid. Repeating the original benchmark unchanged gives 16.07×/3.05×, compared with 15.96×/4.03× in its recorded session. Hence we report a 3–4× reference range; absolute times and fine differences depend on shared-host load. These measure query-prediction costs, not field-solver speedups.

Table 11: **Single-thread query-prediction cost.** Panel A retains the original full-method timing comparison; panel B pairs CMI with the full ensemble by platform in a separate session. The latter retains median query latency and P95 episode latency without repeating equivalent batch/table columns. CMI ratios use unrounded timings; the two sessions should not be pooled. Bold marks the minimum per table class and metric in panel A, and the lower CMI/ours latency in each platform/workload pair in panel B. Underlined Ours timings rank second within the corresponding comparison.

A. Original benchmark session; 50 timed rounds after warm-up

Table class	Method	ms/episode	ms/query	CMI ratio
Public (13 nodes)	IDW	**0.890**	**0.148**	**0.62**×
	RBF-GP	1.208	0.201	0.84×
	CMI-1D	1.432	0.239	1.00×
	Ours	22.855	3.809	15.96×
Reference (128 nodes)	IDW	**3.576**	**0.056**	**0.13**×
	RBF-GP	4.140	0.065	0.15×
	OpenRCX-LI	10.377	0.162	0.38×
	CMI-2D	27.043	0.423	1.00×
	PCHIP	113.651	1.776	4.20×
	Ours	109.036	1.704	4.03×

B. Platform session; medians, same workload within each table class

Platform	Nodes	ms/query		P95 ms/episode		CMI ratio
		CMI	**Ours**	CMI	**Ours**	**Ours** / CMI
asap7	13	**0.072**	<u>1.141</u>	**0.603**	<u>6.958</u>	15.77×
gf180	13	**0.071**	<u>1.147</u>	**0.461**	<u>7.070</u>	16.27×
ihp_sg13g2	13	**0.070**	<u>1.149</u>	**0.459**	<u>7.066</u>	16.41×
sky130	13	**0.070**	<u>1.149</u>	**0.461**	<u>7.057</u>	16.37×
asap7	128	**0.131**	<u>0.400</u>	**8.767**	<u>26.953</u>	3.06×

Pruning diagnostic. On public rules, removing PCHIP, aligned-width interpolation and six GP branches changes in-support random/block NRMSE from 0.0803/0.0678 to 0.0804/0.0678, and out-of-support random/block/extrapolation from 0.0910/0.1391/0.1620 to 0.0912/0.1398/0.1628. The largest relative change is below 0.5%, while the measured scoring pipeline is 6.1× faster. This diagnostic follows observed gate usage and includes recomputation of agreement features; it is not used to replace the full-bank headline results or to claim the same saving on rank-two tables.

H Dataset organization and construction

The benchmark organizes characterized geometry–capacitance pairs into observation-conditioned prediction episodes. Its construction combines public extraction rules, independently solved numerical grids and controlled masks. Table 12 summarizes the data used by the different protocols. The contribution is this organization and evaluation interface together with the numerical reference; the public rule values originate from existing process resources.

Table 12: **Data and evaluation cohorts.** Counts describe the indicated protocols, not additive counts of independent datasets. Public split counts are the original allocation before the downstream exclusions described below. Boundary cohorts supplement the headline confirmation evaluations.

Data / protocol	Geometry and responses	Size and use
Public rules	Rank-one variation; $K = 2$	1,713 tables across ASAP7, GF180, IHP-SG13G2 and Sky130; original train/validation/test allocation: 1,116/256/341; 340 scored test instances after duplicate weighting.
Numerical reference	Rank-two width–distance grids; $K = 3$	27 evaluation tables, each 8×16 geometries: 16 confirmation, 8 validation and 3 boundary-layer tables. Fitting uses 16 separate 9×17 tables; configuration selection uses 24 validation tables.
ASAP7 transfer	Target observations, frozen source parameters	8 new-geometry confirmation tables; 2 boundary-layer tables evaluated separately.
Downstream audit	Fixed routed layouts; masked extraction rules	84 rewritten tables excluded from model fitting; 16 process/design/mask settings on 8 fixed layouts.

H.1 Public rule-table organization

The public corpus contains 1,713 capacitance tables from ASAP7, GF180, IHP-SG13G2 and Sky130 extraction rules. A row stores a candidate wire geometry and its capacitance response vector. Tables carry the structural families OVER, UNDER, OVERUNDER and DIAGUNDER, together with the available layer and material context used by the tokenizer in Section 3.2. Public models predict two response components ($K = 2$).

The original partition assigns associated groups to training, validation and test, yielding 1,116, 256 and 341 tables, respectively. Duplicate weighting leaves 340 scored test instances. These original counts must be distinguished from the exclusions made for the downstream audit. Although three coordinates are stored, every public table has one constant coordinate and two identical varying coordinates. We verified this property across all 1,713 tables; observed support is rank one under the evaluated masks. The rank-matched multilinear control is therefore CMI-1D. The rule responses are extraction references, rather than independent field-solver truth.

H.2 Independent numerical reference and transfer

The numerical reference comprises 27 tables constructed from ASAP7 and FreePDK15 CTF material descriptions. Each table samples an 8×16 grid in wire width and distance, giving 128 candidate geometries. Capacitances are computed under unit conductor excitations using quadratic finite elements (Gustafsson & McBain, 2020) on Gmsh meshes (Geuzaine & Remacle, 2009). The resulting grids span two independent geometric axes and use three response components ($K = 3$); their rank-matched baseline is CMI-2D. The construction is a two-dimensional physical idealization.

Sixteen tables provide new-geometry confirmation, eight provide validation, and three form a boundary-layer cohort. These 27 tables contain no training partition: the headline reference confirmation and boundary results come from this collection, whereas fitting and model selection use the separate cohorts specified below. Source-only transfer trains on FreePDK15 tables and evaluates new geometries on ASAP7. The reported target is ASAP7, with eight confirmation tables and a separate two-table boundary cohort. Target observations enter the support computations and model inputs, but they never trigger target-domain parameter updates. Source training scales remain fixed. Model selection also uses source labels only: the 12 tables in the archived `to_asap7_val` cohort

are all FreePDK15. The earlier labels "To ASAP7 val." denote the transfer destination, not the process supplying validation labels. This applies to the branch-prior and alternative-gate comparisons in Tables 5 and 10. Appendix I.2 reports the small boundary cohorts separately from the headline results.

Fitting and selection cohorts. The original FEM collection contains 32 tables on 9×17 grids (153 geometries per table), split into 16 training, eight validation and eight original test tables across the two processes. Reference fitting uses the 16 training tables: four structural families at layer indices 2 and 3 in each process. The stage-1 residual checkpoints and stage-2 models use this same training partition. The reference model-selection cohort contains 24 tables: eight original validation tables, eight second-round validation tables, and eight validation tables from the 27-table collection. These are layer indices 4 and 6; the two layer-6 collections have different grids. Headline confirmation uses the 16 third-round tables at layer indices 7 and 8, while the three boundary tables use indices 1 or 9. Training and headline reference confirmation have no overlapping process/family/layer/context tuples.

For transfer to ASAP7, stage 1 and stage 2 fit only the eight FreePDK15 training tables. Stage 1 uses four original FreePDK15 validation tables; subsequent configuration comparisons use 12 FreePDK15 validation tables (four from each of the three validation collections above). Although the archived fold file also stores ASAP7 rows marked `test`, both fitting loops filter the `train` partition; those target labels do not enter updates. Confirmation uses eight ASAP7 tables at indices 7 and 8, and the two boundary tables use indices 1 and 9. Target observations are read only as inference context, and withheld target responses are used only for scoring. The provenance audit checks the cohort hashes in the archived validation receipts, rather than inferring label access from run names.

Table 13: **Fitting, selection and headline confirmation.** Counts are unique tables; validation can contain different grids for the same layer. The public counts reflect downstream exclusions and deduplication. FEM fitting grids have 153 nodes; the headline FEM/transfer confirmation grids have 128 nodes. Boundary cohorts are excluded from the last column.

Setting	Fitting	Selection	Confirmation	Selection-label source
Public rules	1,051	246	340	Public training-domain processes
FEM reference	16	24	16	ASAP7 and FreePDK15
Transfer to ASAP7	8	12	8	FreePDK15 only

H.3 Support–query episode construction

For each table, we retain the candidate coordinates X and context c, select observed indices S, and assign the remaining indices to $\mathcal{Q}$. Only y_S is exposed to the predictor. Query response fields are zeroed before embedding, with explicit observation indicators distinguishing unavailable responses from measured zeros. Anchors, envelopes, numerical corrections and routing descriptors are all reconstructed from the retained observations. Withheld responses enter only the training loss or evaluation metric.

Evaluation uses random, contiguous-block and boundary-extrapolation masks at 10/30/50/70% missingness, with five mask seeds and every applicable whole-width fold. Random masks remove individual observations; block masks remove contiguous coverage; boundary masks withhold geometries beyond the retained observed range. Whole-width folds withhold all rows at an applicable width. These masks expose different support configurations without changing the underlying reference responses. Training draws fresh masks and missingness rates as specified in Section 3.6.

After masking, each query is classified by its position relative to the observed convex hull and affine span. Queries inside this region contribute to the in-support regime; other queries contribute to the out-of-support regime. Boundary-extrapolation masks have no in-support queries. The query coordinates and reference responses still belong to the characterized dataset, so this protocol measures extrapolation beyond current observations, not unrestricted extrapolation beyond all characterized geometries. All methods are scored on the same query population. Validation selects configurations before the confirmation cohorts are evaluated; Section 4.2.3 defines selection and aggregation. Appendix I reports the 60 populated cohort–regime–protocol–rate conditions.

H.4 Separation from downstream extraction data

The downstream audit writes predictions into 84 masked rule tables and reruns PEX and STA on fixed routed layouts. Of these 84 tables, 55 belonged to the original training split. The public ensemble was therefore retrained with all 84 excluded from training, validation and test. Public confirmation results and the downstream audit use that held-out model family on distinct table pop-

ulations. Export preserves unmasked values exactly; the original full-table extraction supplies the downstream reference. Appendix J details the export checks and replay protocol.

This characterization benchmark differs from design-level corpora such as CircuitNet 2.0 (Jiang et al., 2024), which provide routability, timing and power data. Our table-level protocols test support-conditioned capacitance prediction; the separate layout audit tests its consequences for extraction and timing.

I Additional prediction evidence

I.1 The complete missingness surface

Table 14 places all 60 scored cohort–regime–protocol–rate conditions in one matrix, pairing our result with rank-matched CMI at each rate. We improve in all 60 cells. On the reference, the relative block-mask in-support gain grows from 24% at 10% missingness to 91% at 70%, consistent with greater benefit when contiguous masking removes local interpolation support. At 10% public missingness, the evaluated masks remove the same single node, so protocol scores coincide; these repeated values are not independent evidence.

Table 14: **Complete missingness surface.** Mean-mask macro NRMSE (↓); bold marks the better result in each Ours/CMI pair. CMI is 1D for public rules and 2D otherwise. All 60 populated conditions are included; boundary-extrapolation masks have no in-support queries.

		10% missing		30% missing		50% missing		70% missing	
Cohort	Regime / mask	**Ours**	CMI	**Ours**	CMI	**Ours**	CMI	**Ours**	CMI
Public	In / random	**0.0386**	0.1862	**0.0803**	0.1286	**0.0827**	0.1502	**0.0827**	0.1879
	In / block	**0.0386**	0.1862	**0.0678**	0.2812	**0.0849**	0.1692	**0.1089**	0.9221
	Out / random	**0.0670**	0.1550	**0.0910**	0.5531	**0.0792**	0.4511	**0.1259**	0.6195
	Out / block	**0.0670**	0.1550	**0.1391**	0.9546	**0.1143**	0.7181	**0.1186**	0.6984
	Out / extrap.	**0.0670**	0.1550	**0.1620**	1.1867	**0.1319**	0.8819	**0.1073**	0.7849
Reference	In / random	**0.0059**	0.0531	**0.0103**	0.0495	**0.0167**	0.0391	**0.0240**	0.0528
	In / block	**0.0534**	0.0703	**0.0366**	0.1502	**0.0343**	0.2481	**0.0370**	0.4008
	Out / random	**0.2077**	0.4236	**0.0895**	0.1344	**0.0855**	0.1639	**0.0557**	0.1151
	Out / block	**0.1832**	0.5769	**0.1203**	0.9790	**0.0803**	0.8870	**0.0887**	0.9009
	Out / extrap.	**0.1826**	0.4857	**0.1137**	0.6404	**0.0935**	0.7525	**0.1239**	0.7935
To ASAP7	In / random	**0.0102**	0.0523	**0.0159**	0.0488	**0.0254**	0.0386	**0.0378**	0.0520
	In / block	**0.0643**	0.0679	**0.0531**	0.1473	**0.0614**	0.2439	**0.0678**	0.3949
	Out / random	**0.1886**	0.4122	**0.0877**	0.1330	**0.0876**	0.1607	**0.0754**	0.1133
	Out / block	**0.1574**	0.5614	**0.1358**	0.9583	**0.0994**	0.8729	**0.1077**	0.8914
	Out / extrap.	**0.1677**	0.4737	**0.1073**	0.6297	**0.0929**	0.7429	**0.1308**	0.7862

I.2 Boundary-layer cohorts

Table 15 combines the reference and ASAP7 transfer boundary cohorts. The method leads all five reference conditions and four of five ASAP7 conditions; RBF-GP wins ASAP7 in-support random missingness. These cohorts contain only three and two tables, respectively, so conditions within a cohort do not establish independent-geometry confidence.

Table 15: **Boundary-layer prediction at 30% missingness.** NRMSE (↓), with cohort table counts in parentheses. Bold marks the best method; underlining marks our second-best value in each cohort/condition.

		In-support		Out-of-support		
Cohort	Method	Random	Block	Random	Block	Extrap.
Reference (3)	CMI-2D	0.0545	0.1504	0.1336	0.9052	0.6038
	OpenRCX-LI	0.0549	0.1498	0.1836	0.9033	0.6271
	PCHIP	0.0484	0.0947	0.1836	0.9033	0.6271
	IDW	0.1590	0.6366	0.5992	2.7570	2.0935
	RBF-GP	0.0149	1.4523	0.1017	7.4223	7.3902
	Ours	**0.0127**	**0.0351**	**0.0779**	**0.1453**	**0.1072**
ASAP7 (2)	CMI-2D	0.0550	0.1585	0.1418	1.0019	0.6655
	OpenRCX-LI	0.0549	0.1598	0.2036	1.0001	0.6913
	PCHIP	0.0482	0.1017	0.2036	1.0001	0.6913
	IDW	0.1608	0.6911	0.7007	3.0702	2.3259
	RBF-GP	**0.0149**	1.5683	0.1188	8.1893	8.1253
	Ours	0.0203	**0.0643**	**0.0752**	**0.1298**	**0.0908**

I.3 PUBLIC-RULE UNCERTAINTY AND PROCESS/MODE DETAIL

The paired bootstrap keeps all masks of each associated group together and uses 2,000 replicates. In-support differences (ours minus PCHIP) over 338 groups are -0.0267 $[-0.0344, -0.0177]$ for random and -0.1213 $[-0.1748, -0.0684]$ for block masks. Out-of-support differences against CMI-1D over 340 groups are -0.4622 $[-0.6010, -0.3955]$, -0.8155 $[-1.0006, -0.4973]$ and -1.0247 $[-1.3058, -0.5493]$ for random, block and extrapolation. Every comparison has bootstrap $P(\text{better}) = 1.000$.

Table 16 retains the process/family breakdown without replacing the main macro metric. ASAP7 has the highest pooled error in every populated mode; DIAGUNDER is especially difficult on ASAP7 and Sky130. The IHP DIAGUNDER cell has no positive truth energy and is absent, rather than zero. Best/worst summarize individual tables, average gives the mean table RMSE, and pooled weights queries equally; differences between average and pooled reflect their weighting conventions.

Table 16: **Public-rule errors by process and coupling mode.** 30% block missingness, shown as count:normalized RMSE. Counts denote tables for best/worst and scored queries for average/pooled, and repeat across statistics. "–" indicates no positive truth energy. Bold marks the lowest error across processes for each statistic and coupling mode.

Process	Statistic	OVER	UNDER	OVERUNDER	DIAGUNDER
asap7	best	1:0.0157	1:0.0196	2:0.0103	1:**0.0017**
asap7	worst	2:0.2832	2:0.3197	1:0.5653	1:0.8028
asap7	average	360:0.1027	280:0.1045	720:0.1051	48:0.2297
asap7	pooled	360:0.1034	280:0.1138	720:0.1212	48:0.1901
gf180	best	1:**0.0091**	2:0.0080	2:0.0120	1:0.0070
gf180	worst	1:0.3397	1:0.1090	1:0.1525	1:**0.1402**
gf180	average	3840:0.0654	2400:0.0413	1960:**0.0376**	2360:**0.0320**
gf180	pooled	3840:0.0683	2400:0.0493	1960:0.0472	2360:**0.0458**
ihp_sg13g2	best	2:0.0115	2:**0.0078**	1:**0.0048**	–
ihp_sg13g2	worst	1:**0.1125**	1:**0.0552**	1:0.1478	–
ihp_sg13g2	average	240:**0.0491**	160:**0.0257**	280:0.0420	–
ihp_sg13g2	pooled	240:**0.0556**	160:**0.0325**	280:0.0564	–
sky130	best	1:0.0248	2:0.0194	2:0.0169	1:0.0039
sky130	worst	2:0.1257	1:0.0666	1:**0.0931**	2:0.4100
sky130	average	160:0.0598	120:0.0433	160:0.0422	120:0.0959
sky130	pooled	160:0.0598	120:0.0442	160:**0.0457**	120:0.1274

J DOWNSTREAM AUDIT PROTOCOL AND ENDPOINT EVIDENCE

Table 3 reports all 16 design/process/mask conditions. Each run reuses the golden routed snapshot and constraints and reruns only extraction and the final timing report. Golden replay is checked for identical parasitics; export validation checks both exact preservation of unmasked lines and round-trip recovery of written values. Mask construction follows the deterministic table-evaluation recipe, but the 84 rewritten tables are distinct from the public confirmation cohort and excluded from model fitting.

Worst slack uses each platform's Liberty time unit converted to ns. The main timing columns take the maximum absolute deviation across the three seeds for each design and mask. Extraction wall time and peak memory measure the reporting flow and are not substitutes for query-prediction latency.

J.1 ENDPOINT, SPATIAL AND DECISION-LEVEL AUDIT

We replay the seven existing gf180/ibex SPEFs (one golden and Linear/ours for seeds 0, 1 and 2 at 50% block missingness). Each replay reads the same final connected ODB, the original SDC and the same TC Liberty library using the pinned OpenROAD/OpenSTA executable. The original routed-database and constraint hashes agree across the archived runs; SPEF hashes are checked before loading, and exported cell and port coordinates agree exactly across all seven replays. Resistance NRMSE is zero in each archived SPEF comparison. The replays reproduce every archived WS/TNS value within its reported rounding precision. Aggregating the original records recovers Table 3: Linear/ours capacitance NRMSE 0.0199/0.0061, maximum $|\Delta\text{WS}|$ 0.1328/0.03701 ns and maximum $|\Delta\text{TNS}|$ 168.6/50.22 ns. These are three-seed aggregates, whereas the spatial maps in Figure 3 depict seed 0 alone.

For endpoint i, let $E_i^m = |s_i^m - s_i^{\text{gold}}|$ and $G_i = E_i^{\text{Linear}} - E_i^{\text{ours}}$. The spatial maps average these errors in identical 50×50 bins, using shared color limits for the two methods and a symmetric scale for G_i. All endpoint coordinates refer to the unchanged physical database. In Figure 3, boxes denote placed cells and points denote constrained endpoints, not routed-metal geometry. Coordinates are in μm and errors in ns; the two error maps share color limits, and the difference map uses a symmetric scale. Gray bins contain no constrained endpoints.

Endpoint coverage and units. OpenSTA enumerates 3,859 endpoints in every replay: 2,039 have finite setup slack and 1,820 are unconstrained. No endpoint is absent from any variant. The unconstrained endpoints remain in the raw data with infinite slack and are explicitly excluded from the finite-error, ranking and sign statistics. For every constrained endpoint we use its minimum setup slack; internal pins are mapped to their instance's bounding-box center, and output ports to their physical pin-box center. All coordinates come from ODB in μm, and all slacks are in ns. The exported STA properties have 10^{-6} ns display resolution; a 2×10^{-6} ns tolerance defines ties only for the fraction of endpoints with improved error. Violation classification uses the original, unshifted threshold $s < 0$.

Decision definitions. A false negative (FN) is a golden violation classified as safe; a false positive (FP) is a golden-safe endpoint classified as violating. Sign accuracy counts both correctly safe and correctly violating endpoints. For Top-K agreement we retain one worst setup path per constrained endpoint, sort by slack with endpoint/startpoint names breaking ties, and compare either the endpoint sets (EP@K) or the full ordered data-pin sequences (Path@K). Each overlap is the intersection size divided by K. We also export startpoint–endpoint pair overlap; it agrees with full-pin-sequence overlap at the reported K values in this case. Spearman ρ and Kendall τ use all finite endpoint slacks.

Table 17: **Decision fidelity for all three seeds.** gf180/ibex, 50% block missingness; 2,039 constrained endpoints per row. MAE and RMSE are endpoint-slack errors in ns, Sign is slack-sign accuracy, EP@50 is endpoint overlap, Path@50 uses the full data-pin sequence, and ρ is Spearman correlation with golden slack. Bold marks the best value within each seed (lower MAE/RMSE/FN/FP; higher Sign/EP@50/Path@50/ρ); ties are bolded. Our second-best values are underlined. No seed is omitted.

Seed	Method	MAE	RMSE	Sign (%)	FN	FP	EP@50	Path@50	ρ
0	Linear	0.0059	0.0061	99.95	**0**	1	**1.00**	**1.00**	0.99999
0	**Ours**	**0.0026**	**0.0028**	**100.00**	**0**	**0**	**1.00**	**1.00**	**1.00000**
1	Linear	0.0879	0.0935	99.75	5	**0**	0.98	0.76	0.99733
1	**Ours**	**0.0261**	**0.0277**	**99.90**	**0**	<u>2</u>	**1.00**	**1.00**	**0.99960**
2	Linear	**0.0038**	0.0044	**100.00**	**0**	**0**	**1.00**	**1.00**	0.99984
2	**Ours**	<u>0.0040</u>	**0.0042**	**100.00**	**0**	**0**	**1.00**	**1.00**	**0.99999**

Seed 0 improves endpoint error at 99.90% of endpoints, and seed 1 at 100%. Seed 2 improves 51.10%, worsens 48.85%, and ties on 0.05%; its MAE rises from 0.00384 to 0.00396 ns while RMSE falls from 0.00443 to 0.00416 ns. The seed-1 endpoint MAE falls from 0.08787 to 0.02610 ns. At $K = 100$, seed-1 endpoint overlap increases from 0.74 to 0.98 and full-path overlap from 0.42 to 0.98. Seeds 0 and 2 have perfect overlap for both methods at $K = 10, 20, 50, 100$. These distinctions matter: restoring slack values, restoring their ordering and preserving their signs are related but not interchangeable claims.

Reproducible artifacts. The replay script saves input hashes, STA logs, raw endpoint/path exports, seed-wise CSVs and PDF/SVG/PNG figures; its entry point is listed in Appendix C. Additional scatter and bin-maximum maps expose spatial averaging. No retraining, rerouting, constraint shifts or ECO operations are used.

J.2 Spatial fidelity across additional process/design settings

Figures 7 and 8 extend Figure 3 to four additional archived settings: ASAP7/IBEX, IHP-SG13G2/IBEX, GF180/AES and Sky130HS/IBEX. All use 50% block missingness and seed 0, fixed across settings rather than selected separately by error. The plots are reconstructed from existing STA exports; no training, routing or timing rerun is introduced. Golden, Linear and ours have identical endpoint coverage and exported geometry within each setting. Only endpoints with finite slack in all three variants are plotted: 3,704, 3,704, 691 and 3,975, respectively. The archived worst slack and total negative slack are checked against the endpoint exports before plotting. ASAP7 values are converted from ps to ns; the other exports already use ns.

Each column follows the four-panel organization of Figure 3. Errors are averaged in 50×50 spatial bins, with a shared zero-based scale for Linear and ours within each setting. Error-reduction maps use a symmetric scale; gray bins contain no constrained endpoints. Color limits and physical extents vary across settings, so cross-setting comparisons must use the labeled units rather than color intensity alone. Panel annotations report endpoint-weighted MAE, not an unweighted average of occupied bins. These fixed-seed maps supplement the three-seed aggregate audit in Table 3.

K Discussion and limitations

What the results say about the design. The matched-budget controls distinguish a strong learned residual contribution from condition-dependent routing gains. The full hybrid model has a larger advantage on the FEM reference than on public rules, while a binary gate is better in two transfer conditions (Appendix E.1). Public rules isolate the correction because their anchor equals CMI; rank-two controls show that boundary handling also contributes. Strong block-mask results and weaker random-mask transfer suggest that the benefit depends on what the remaining support can already resolve, rather than on a uniform advantage of learned prediction.

Where the evidence stops. Extrapolation is measured relative to the observations retained by each mask. The scored query geometries come from characterized reference tables; the experiments do not establish accuracy arbitrarily far beyond those tables or outside the source training distribution. The FEM reference is a two-dimensional idealization, and source-only transfer covers two related material stacks. Public rules are extraction references, not independently solved physical truth. The downstream audit covers four open processes and two designs with fixed layouts and held-out rewritten tables; it measures extraction/STA agreement, not foundry signoff accuracy. Numerical interpolation remains cheaper, and gate failures show that a useful branch need not receive an appropriate weight. The failed descriptor-key variants further caution against treating richer routing inputs as a sufficient solution. The appendix retains these failures, small boundary cohorts and all missingness rates alongside the positive main results.

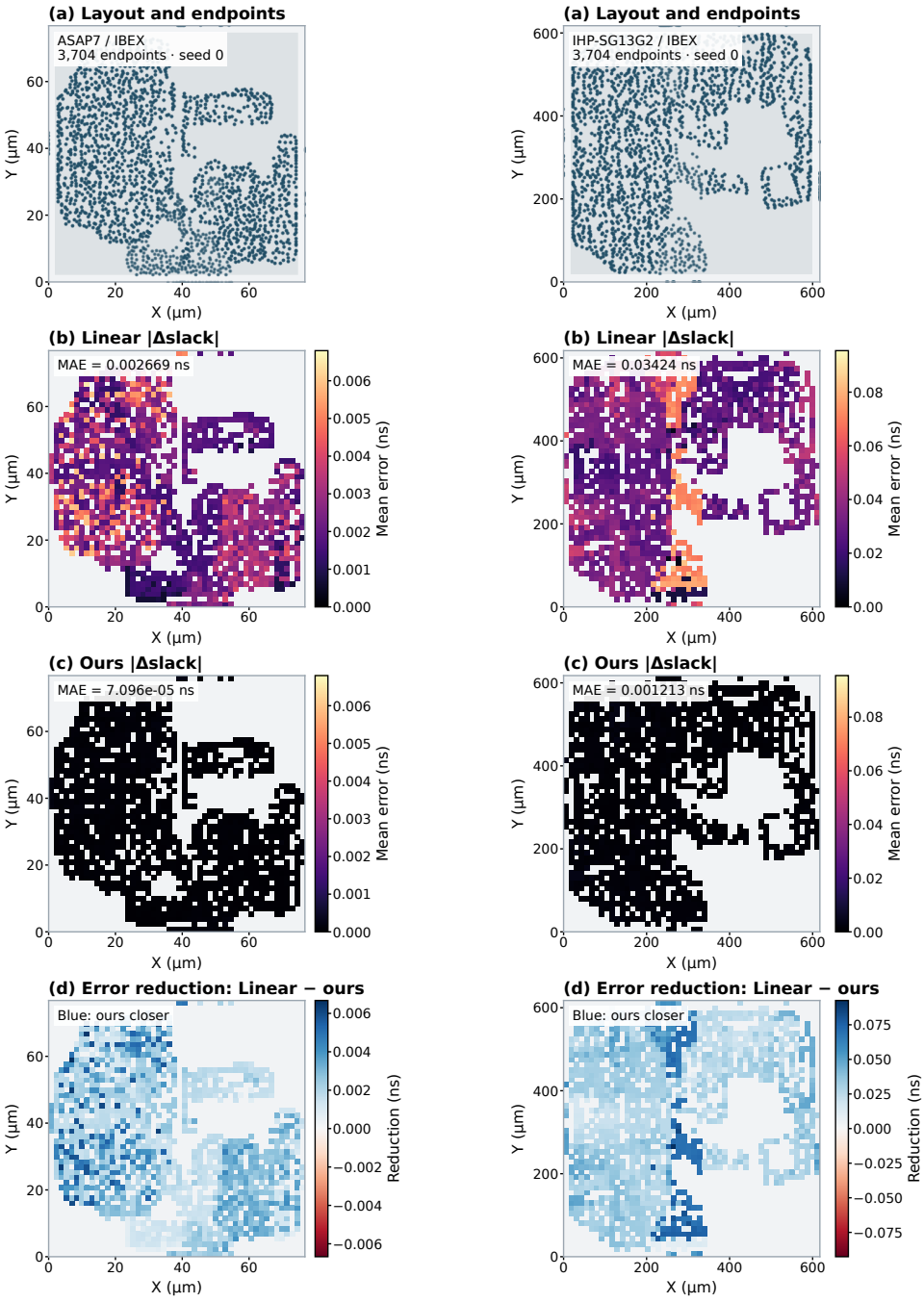


Figure 7: **Spatial timing fidelity on ASAP7/IBEX (left) and IHP-SG13G2/IBEX (right).** 50% block missingness, seed 0. Each column shows (a) placed cells and constrained endpoints, (b) Linear absolute slack error, (c) ours and (d) error reduction relative to Linear. Heatmaps average endpoint errors in 50×50 bins. The two error maps share a scale within each column; reduction uses a symmetric scale, with blue favoring ours. Gray bins are empty. Coordinates are in μm and all errors in ns, including the converted ASAP7 exports.

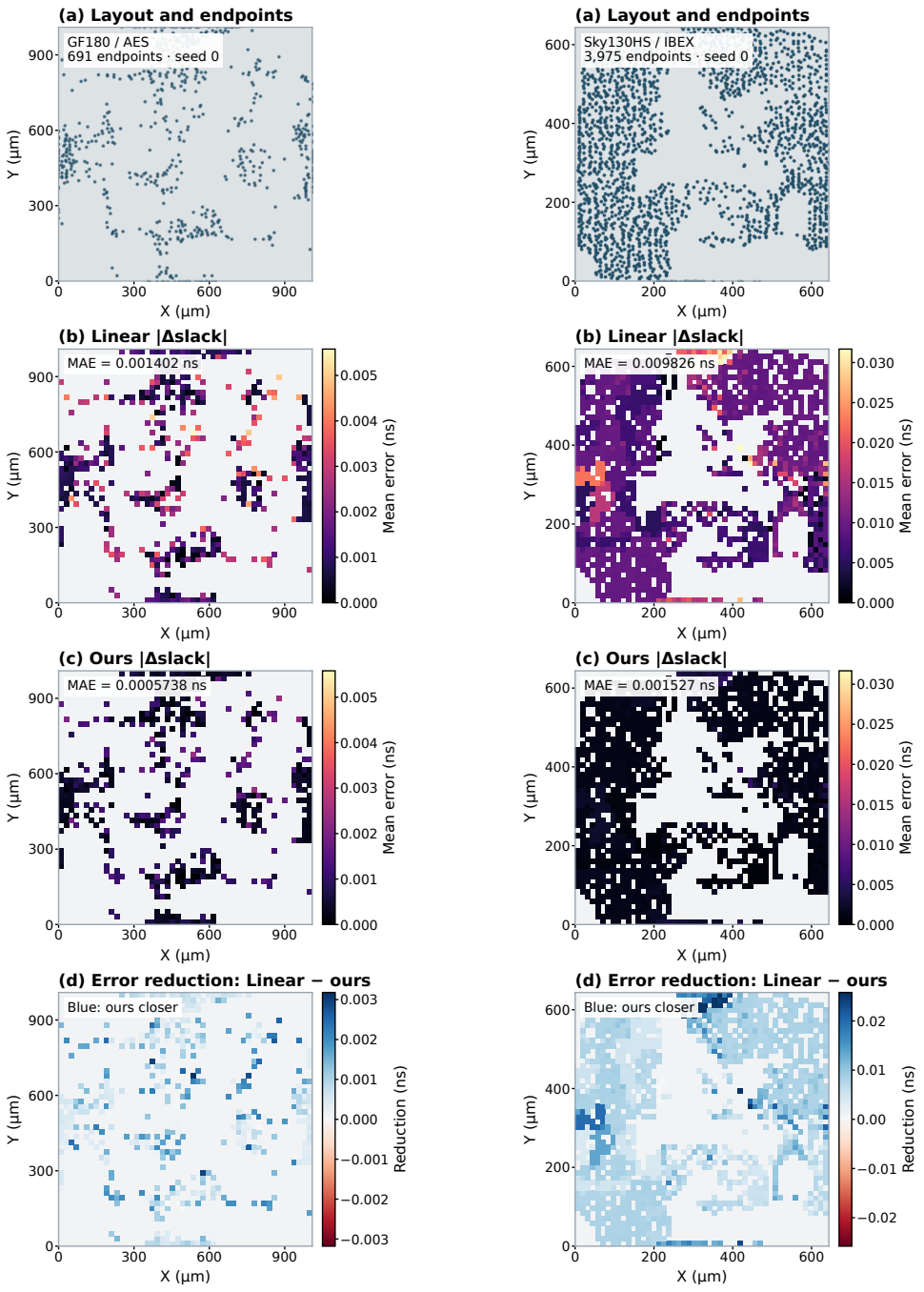


Figure 8: **Spatial timing fidelity on GF180/AES (left) and Sky130HS/IBEX (right).** 50% block missingness, seed 0. Rows show layout/endpoints, Linear error, ours and error reduction, following Figure 3. The maps include all 691 and 3,975 constrained endpoints, respectively; the sparse GF180/AES coverage leaves more empty bins. Within each column, Linear and ours share the same error scale; blue/red in the symmetric reduction map indicate improvement/degradation. Physical extents and color limits are specific to each setting.