DRAFT-SOLVER: DYNAMIC SPARSE ATTENTION VIA NODE DRAFTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning surrogate models for steady-state Partial Differential Equations (PDEs) on large unstructured meshes represents a challenging task because relevant interactions may span the entire domain, while accurate predictions often require additional computation only in localized, task-dependent regions. Dense attention captures long-range dependencies but becomes impractical on large discretizations, whereas sparse and multiscale approaches commonly prescribe where fine-scale processing should occur using geometric or physical priors. To address these limitations, we introduce DRAFT-Solver (**D**ynamic spa**R**se **A**ttention via node dra**FT**ing), a dual-scale architecture that learns its computational allocation directly from data. DRAFT-Solver combines global information exchange through softly constructed latent tokens with fine-grained self-attention over a learned, fixed-budget subset of nodes. Both token routing and node selection are trained end-to-end without predefined surface regions or manually specified masks. Across benchmarks spanning two- and three-dimensional aerodynamics, semiconductor-device simulation, and biomolecular electrostatics, DRAFT-Solver achieves competitive or improved performance relative to graph-, neural-operator-, and transformer-based baselines.

1 INTRODUCTION

Predicting physical fields around complex geometries is central to engineering design, including aerodynamic analysis and semiconductor device simulation (Li et al., 2020b; Jang et al., 2023; Qi et al., 2025). Traditional numerical solvers provide high-fidelity solutions but are often too expensive for large-scale design exploration (Zienkiewicz et al., 2005). Learning-based surrogates offer substantially faster inference (Li et al., 2022; Li et al., 2023; Serrano et al., 2023), but scaling them to industrial unstructured meshes remains challenging.

Realistic meshes may contain hundreds of thousands or millions of nodes (Zhou et al., 2026; Zhong et al., 2026), making dense self-attention prohibitively expensive. At the same time, prediction difficulty is spatially non-uniform: boundaries, wakes, material interfaces, and high-gradient regions often require greater modeling capacity than smoother parts of the domain (Qi et al., 2025; Zhao et al., 2026). Existing approaches improve scalability through local graph propagation, latent tokenization, or geometry-informed multiscale decompositions (Li et al., 2020a; Wu et al., 2024; Alkin et al., 2025). Together, these observations motivate combining efficient global communication with sample-dependent fine-scale computation directly on unstructured meshes, without prescribing where refinement should occur.

To address these challenges, we introduce **DRAFT**-Solver (**D**ynamic spa**R**se **A**ttention via node dra**FT**ing), hereafter DRAFT.

DRAFT is a dual-scale surrogate that learns computational allocation from data. Its *coarse* pathway softly aggregates mesh nodes into latent tokens, performs global self-attention in token space, and routes the updated information back to every node. On the other hand, its *fine* pathway applies additional self-attention to a learned, fixed-budget subset of nodes selected through differentiable hard routing. Both the token assignments and fine-node selection are learned from data without prescribing surface regions or geometric masks. Global physical parameters condition every DRAFT block through Feature-wise Linear Modulation (FiLM) (Perez et al., 2017; Peebles & Xie, 2023).

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.

We evaluate DRAFT on four benchmarks: AirFRANS (Bonnet et al., 2023) and ShapeNet Car (Chang et al., 2015) for aerodynamics, CMOS (Kovani et al., 2026) for semiconductor simulation, and SPACIER-M (Wu & Luo, 2025) for biomolecular electrostatics. We also analyze its learned computational allocation and the effect of error-ranking supervision across boundary- and bulk-driven systems.

Our main contributions are:

- **Coarse-grained token pathway.** For a mesh containing N nodes, DRAFT learns to aggregate node representations into $K \ll N$ latent tokens, performs global self-attention in token space, and routes the resulting information back to every node.
- **Fine-grained sparse pathway.** DRAFT learns a fixed-budget subset of $M < N$ nodes that receives additional node-level attention. Selection is optimized end-to-end through a straight-through estimator and can optionally be guided by error-ranking supervision, without predefined surfaces, partitions, or masks. Together, the two pathways have interaction cost $\mathcal{O}(NK + K^2 + M^2)$.
- **Cross-domain performance with learned allocation.** Across four benchmarks spanning aerodynamics, semiconductor-device simulation, and biomolecular electrostatics, DRAFT achieves the best reported AirFRANS and CMOS results, the lowest field-level errors on ShapeNet Car, and the best R^2 and Mean Absolute Error (MAE) on SPACIER-M among the considered methods. These results are obtained with a common architecture that learns both token assignments and fine-node allocation from data, without dataset-specific refinement regions, geometric partitions, or prescribed attention masks.

2 Related Works

PDE solution operators on irregular geometries. Neural operators have been extensively explored in recent years as a framework for PDE surrogate modeling, learning mappings between input conditions and physical solution fields (Kovachki et al., 2021). However, early spectral formulations primarily target regular discretizations (Li et al., 2020b). Surrogate models for irregular geometries have subsequently developed along two broad directions. Graph-based methods operate directly on mesh connectivity, preserving the native discretization but relying predominantly on local message passing (Pfaff et al., 2021; Brandstetter et al., 2022). Geometry-aware operator methods instead map irregular inputs to continuous or latent representations through learned coordinate deformations, graph-to-grid operators, or coordinate-based neural fields (Li et al., 2022; Li et al., 2023; Serrano et al., 2023; Deng et al., 2024). Whereas these approaches primarily address how physical fields are represented across varying geometries and discretizations, DRAFT focuses on the complementary problem of distributing computation when prediction difficulty is spatially non-uniform.

Scalable and adaptive PDE transformers. Existing PDE transformers improve scalability and spatial adaptivity along two complementary directions. One direction compresses mesh information into a smaller set of latent tokens, enabling efficient global interactions (Wu et al., 2024; Alkin et al., 2025; Yang et al., 2026); recent extensions further support industrial-scale meshes through memory-aware execution or precomputed geometric decompositions (Zhou et al., 2026; Zhong et al., 2026). A second direction distributes spatial resolution non-uniformly to emphasize physically informative regions. SpiderSolver uses boundary-informed spectral partitions (Qi et al., 2025), whereas MeshTok constructs multiscale tokens over fixed simulation grids, assigning finer tokens to regions with stronger spatial variations (Zhao et al., 2026). DRAFT combines efficient global token reasoning with learned node-level refinement directly on unstructured meshes: every node receives a coarse update, while a sample-dependent, fixed-budget subset receives additional attention. Unlike geometry-prescribed or grid-based refinement, DRAFT learns both global token assignments and fine-node selection end-to-end, with the latter optionally guided by per-node prediction errors.

3 Methodology

We consider mesh-based surrogate modeling for steady-state PDEs on unstructured meshes. Each simulation is represented by N mesh nodes and a feature matrix $\mathbf{X} \in \mathbb{R}^{N \times d_{\text{in}}}$, where d_{in} is the

number of input features per node. These features encode geometric and physical information, such as spatial coordinates, surface normals, and boundary indicators. When available, simulation-level PDE parameters are represented by $\mathbf{c} \in \mathbb{R}^{d_c}$, where d_c denotes the number of global conditioning variables. For each node, the target matrix $\mathbf{Y} \in \mathbb{R}^{N \times d_{\text{out}}}$ contains d_{out} physical quantities, such as velocity, pressure, electrostatic potential, or carrier densities. We learn a surrogate f_θ, parameterized by trainable weights θ, that produces the prediction $\widehat{\mathbf{Y}}$:

$$\widehat{\mathbf{Y}} = f_\theta(\mathbf{X}, \mathbf{c}), \qquad f_\theta : \mathbb{R}^{N \times d_{\text{in}}} \times \mathbb{R}^{d_c} \to \mathbb{R}^{N \times d_{\text{out}}}. \tag{1}$$

When global parameters are unavailable, $\mathbf{c}$ is omitted. The model is trained to approximate $\mathbf{Y}$ across varying geometries, mesh discretizations, and simulation conditions. Complete forward-pass pseudo-code is provided in Appendix A.

3.1 DRAFT: Dynamic spaRse Attention via node draFTing

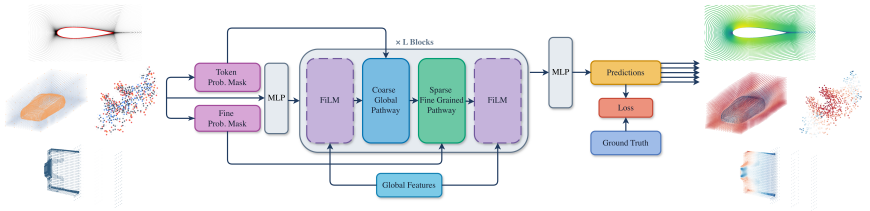

Figure 1: Overview of DRAFT. Each block combines global latent-token attention with fine-grained attention over a learned, fixed-budget node subset. Global PDE parameters are optionally injected through FiLM, and the model is trained end-to-end with optional ranking supervision.

Our architecture (Figure 1) consists of a stack of L DRAFT blocks. Input node features $\mathbf{X} \in \mathbb{R}^{N \times d_{\text{in}}}$ are projected by a multilayer perceptron (MLP) encoder into hidden representations $\mathbf{H}^{(0)} \in \mathbb{R}^{N \times h}$. Each block combines dual-scale attention, comprising coarse latent-token attention for global information propagation and fine-grained attention over a learned subset of nodes, with optional FiLM (Perez et al., 2017; Peebles & Xie, 2023) conditioning on global PDE parameters $\mathbf{c}$.

Given $\mathbf{H}^{(l)}$ representing the hidden representations at the l iteration and global PDE parameters $\mathbf{c}$, a DRAFT block applies FiLM-conditioned normalization before the dual-scale attention module and again before a feed-forward MLP that independently transforms each node representation, with residual connections around all three sublayers:

$$\begin{aligned} \mathbf{H}_{\text{in}}^{(l)} &= \text{FiLM}\left(\mathbf{H}^{(l)} \mid \mathbf{c}\right), \\ \mathbf{H}'^{(l)} &= \mathbf{H}^{(l)} + \text{DualScaleAttn}\left(\mathbf{H}_{\text{in}}^{(l)}\right), \\ \mathbf{H}^{(l+1)} &= \mathbf{H}'^{(l)} + \text{MLP}\left(\text{FiLM}\left(\mathbf{H}'^{(l)} \mid \mathbf{c}\right)\right). \end{aligned} \tag{2}$$

where $\text{DualScaleAttn}(\mathbf{H}_{\text{in}}^{(l)}) = \mathbf{H}_{\text{in}}^{(l)} + \Delta\mathbf{H}_{\text{coarse}}^{(l)} + \Delta\mathbf{H}_{\text{fine}}^{(l)}$ (Eqs. equation 4–equation 7; Appendix A.1). When global PDE parameters are unavailable, FiLM reduces to standard Layer Normalization. The final node representations are mapped to per-node predictions by an output decoder.

3.1.1 Coarse Global Pathway

The coarse pathway enables global information exchange by compressing the N node representations into $K \ll N$ latent tokens, applying self-attention in token space, and routing the updated token features back to the nodes.

The shared assignment matrix $\mathbf{A}$ is computed from the initial node representations:

$$\mathbf{A} = \text{softmax}_K\left(\text{MLP}_{\text{assign}}(\mathbf{H}^{(0)})\right) \in \mathbb{R}^{N \times K}, \tag{3}$$

where each row defines a probability distribution over the K tokens. Matrix $\mathbf{A}$ is computed once and reused across all DRAFT blocks.

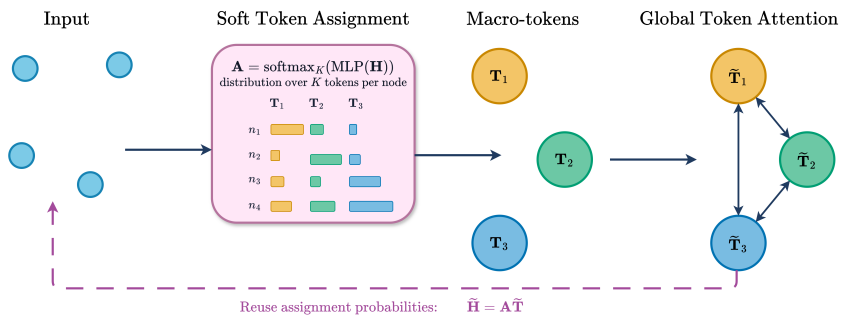

Figure 2: Coarse global pathway. Nodes are softly aggregated into latent tokens, globally updated in token space, and routed back using the same assignment probabilities.

At block ℓ, the FiLM-normalized node representations $\mathbf{H}_{\text{in}}^{(\ell)} \in \mathbb{R}^{N\times h}$ are aggregated into $\mathbf{T}^{(\ell)} \in \mathbb{R}^{K\times h}$, whose rows are normalized weighted averages representing the K latent tokens: Specifically,

$$\begin{aligned}
\mathbf{T}^{(\ell)} &= \left(\mathbf{A}^\top \mathbf{H}_{\text{in}}^{(\ell)}\right) \oslash \left(\mathbf{A}^\top \mathbf{1}_N + \epsilon\right), \\
\tilde{\mathbf{T}}^{(\ell)} &= \text{TokenAttention}^{(\ell)}\left(\mathbf{T}^{(\ell)}\right), \\
\mathbf{H}_{\text{coarse}}^{(\ell)} &= \mathbf{H}_{\text{in}}^{(\ell)} + \text{MLP}_{\text{coarse}}^{(\ell)}\left(\mathbf{A}\tilde{\mathbf{T}}^{(\ell)}\right).
\end{aligned} \tag{4}$$

Here, the operator $\oslash$ denotes element-wise division, with the K-dimensional denominator broadcast across the h feature dimensions and ϵ ensures numerical stability. This reduces global interaction complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(NK + K^2)$. Detailed tensor operations and pseudocode are provided in Appendix A.2.

3.1.2 Fine-Grained Sparse Pathway

The fine pathway complements global token communication by applying additional attention to a learned, fixed-budget subset of nodes, concentrating computation in regions insufficiently resolved by the coarse representation.

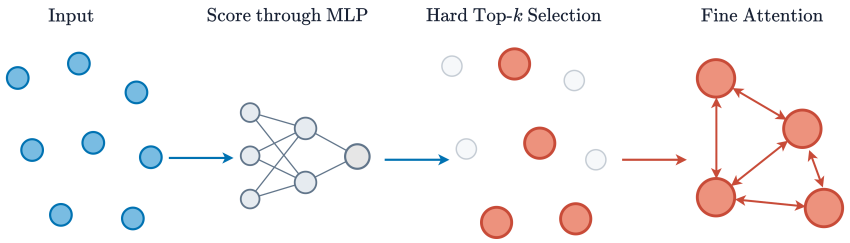

Figure 3: Fine-grained sparse pathway. A learned scoring network selects a fixed-budget subset of nodes for additional self-attention, using hard routing in the forward pass and a straight-through estimator during optimization.

Node-importance scores are computed from the initial representations, and the highest-scoring nodes are selected according to the budget $b \in [0, 1]$:

$$\mathbf{s} = \text{MLP}_{\text{score}}\left(\mathbf{H}^{(0)}\right), \qquad M = \max(1, \lfloor bN \rfloor), \qquad \mathcal{I} = \text{TopMIndices}(\mathbf{s}, M). \tag{5}$$

The resulting selection is reused across all DRAFT blocks. To optimize the discrete routing decision, we use a straight-through estimator (STE) (Bengio et al., 2013; van den Oord et al., 2017), with a sigmoid surrogate for the backward pass:

$$\mathbf{m} = \mathbf{m}^{\text{hard}} + \sigma(\mathbf{s}) - \text{sg}(\sigma(\mathbf{s})), \qquad m_i^{\text{hard}} = \mathbb{1}[i \in \mathcal{I}], \tag{6}$$

where σ is the element-wise sigmoid function, sg denotes the stop-gradient operation, and $\mathbb{1}$ is the indicator function. Thus, the forward pass uses hard top-M routing, while gradients propagate through the sigmoid surrogate.

At block ℓ, the selected coarse representations receive an additional sparse attention update:

$$\begin{aligned}\Delta\mathbf{H}_{\mathcal{I}}^{(\ell)} &= \mathrm{FineAttention}^{(\ell)}\left(\mathrm{Gather}_{\mathcal{I}}\left(\mathbf{H}_{\mathrm{coarse}}^{(\ell)}\right)\right),\\ \mathbf{H}_{\mathrm{dual}}^{(\ell)} &= \mathbf{H}_{\mathrm{coarse}}^{(\ell)} + \mathrm{Scatter}_{\mathcal{I}}\left(\mathbf{m}_{\mathcal{I}} \odot \Delta\mathbf{H}_{\mathcal{I}}^{(\ell)}\right).\end{aligned} \tag{7}$$

Here, $\mathrm{Gather}_{\mathcal{I}}$ selects the rows indexed by $\mathcal{I}$, mapping $\mathbb{R}^{N\times h}$ to $\mathbb{R}^{M\times h}$. Conversely, $\mathrm{Scatter}_{\mathcal{I}}$ maps the resulting updates back to their original rows in $\mathbb{R}^{N\times h}$, assigning zero updates outside $\mathcal{I}$. The symbol $\odot$ denotes element-wise multiplication. The attention cost is reduced from $\mathcal{O}(N^2)$ to $\mathcal{O}(M^2)$. For further details consult Appendix A.3.

3.1.3 FiLM Conditioning

Global PDE parameters $\mathbf{c}$ condition each DRAFT block through Feature-wise Linear Modulation (FiLM) (Perez et al., 2017). The conditioning vector is projected into feature-wise scaling and shifting parameters:

$$[\boldsymbol{\gamma}, \boldsymbol{\beta}] = \mathrm{Linear}_{\mathrm{FiLM}}(\mathbf{c}) \in \mathbb{R}^{2h}, \qquad \boldsymbol{\gamma}, \boldsymbol{\beta} \in \mathbb{R}^{h}. \tag{8}$$

Given a hidden representation $\mathbf{H}$, the conditioned representation is

$$\mathrm{FiLM}(\mathbf{H} \mid \mathbf{c}) = (1 + \boldsymbol{\gamma}) \odot \mathrm{LN}(\mathbf{H}) + \boldsymbol{\beta}, \tag{9}$$

where $\mathrm{LN}(\cdot)$ denotes Layer Normalization and $\odot$ denotes element-wise multiplication.

FiLM is applied before the coarse pathway and after the fine pathway, before the block-level feed-forward network. Its projections are zero-initialized ($\boldsymbol{\gamma} = \boldsymbol{\beta} = \mathbf{0}$), so the modulation initially reduces to standard Layer Normalization.

3.1.4 Training Objective

DRAFT is trained with an end-to-end prediction objective, optionally augmented with ranking and regularization terms for the node-scoring mechanism:

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{e2e}} + \lambda_{\mathrm{rank}}\mathcal{L}_{\mathrm{rank}} + \lambda_{\mathrm{reg}}\mathcal{L}_{\mathrm{reg}}, \tag{10}$$

where $\mathcal{L}_{\mathrm{e2e}}$ is the task-dependent prediction loss and λ_{rank} and λ_{reg} control the auxiliary contributions.

To supervise node selection, we define the detached prediction error of node i, computed in normalized output space (Appendix B), as

$$e_i = \mathrm{sg}\left[\frac{1}{d_{\mathrm{out}}}\sum_{r=1}^{d_{\mathrm{out}}}\left(\widehat{\widetilde{Y}}_{ir} - \widetilde{Y}_{ir}\right)^2\right]. \tag{11}$$

We sample P node-pairs (a_p, b_p), where a_p is a node drawn from the upper τ-quantile of the node-error distribution and b_p is drawn from all nodes. The pairwise hard-margin ranking loss is

$$\mathcal{L}_{\mathrm{rank}} = \frac{1}{P}\sum_{p=1}^{P}\max\left(0,\, \mu - \mathrm{sign}\left(e_{a_p} - e_{b_p}\right)\frac{s_{a_p} - s_{b_p}}{\mathrm{Std}(\mathbf{s}) + \epsilon}\right), \tag{12}$$

where $\mathbf{s} \in \mathbb{R}^N$ contains the raw importance scores from equation 5, $\mathrm{Std}(\mathbf{s})$ denotes their standard deviation over the N nodes, μ is the ranking margin. This objective encourages nodes with larger prediction errors to receive higher importance scores.

We additionally use $\mathcal{L}_{\mathrm{reg}} = \frac{1}{N}\sum_{i=1}^{N} s_i^2$ to discourage excessively large score magnitudes. The ranking loss guides high-error nodes into the fixed-budget mask; complete loss and sampling details are provided in Appendix A.4.

4 EXPERIMENTS

4.1 DATASET

AirFRANS AirFRANS (Bonnet et al., 2023) is a 2D external-aerodynamics benchmark based on steady-state Reynolds-Averaged Navier–Stokes simulations around airfoils. Each mesh contains approximately 180k nodes; following prior work (Bonnet et al., 2023; Qi et al., 2025), we randomly sample 32k nodes for training and evaluation. DRAFT predicts velocity field (v_x, v_y), pressure p, and turbulent viscosity ν_t, conditioned on the inlet velocity components. Following Qi et al. (2025), we report volume and surface normalized-field MSEs (Vol and Surf), relative lift-coefficient error C_L, and its Spearman correlation ρ_L.

ShapeNet Car ShapeNet Car (Chang et al., 2015) is a 3D external-aerodynamics benchmark over vehicle geometries, simulated at a fixed inlet speed of $20\,\mathrm{m/s}$. Each mesh contains approximately 32k nodes, and the targets are velocity field (v_x, v_y, v_z) and pressure p. Following Qi et al. (2025), we report relative ℓ_2 errors on exterior velocity (Vol) and surface pressure (Surf), relative drag-coefficient error C_D, and its Spearman correlation ρ_D.

CMOS Semiconductor Simulations The CMOS dataset (Kovani et al., 2026) contains 3D semiconductor-device simulations governed by the drift–diffusion equations on irregular meshes with spatially varying donor and acceptor concentrations. DRAFT predicts electrostatic potential φ, electron density n, and hole density p, conditioned on the drain and gate voltages $(V_\mathrm{d}, V_\mathrm{g})$. For φ, we report mean absolute error (MAE), maximum absolute error (MaxAE), mean relative error (MRE), mean squared error (MSE), and root mean squared error (RMSE). For n and p, we report log-space mean absolute error (LogMAE), MRE, mean global relative error (MGRE), and maximum relative error (MaxRE).

SPACIER-M SPACIER-M (Wu & Luo, 2025) contains 1,222 biomolecular systems governed by the Poisson–Boltzmann equation, ranging from 268 to more than 140,000 atoms (Baker et al., 2001; Fogolari et al., 2002). Each atom is described by its coordinates, charge, and elemental identity, and DRAFT predicts its electrostatic solvation potential.

Following Wu & Luo (2025), we evaluate the resulting electrostatic solvation free energy using the coefficient of determination (R^2), mean absolute error (MAE), and mean absolute percentage error (MAPE). Higher R^2 is better, whereas lower MAE and MAPE are better.

5 RESULTS

We evaluate DRAFT on four benchmarks covering different geometries, discretizations, dimensionalities, and governing physical processes: AirFRANS (Bonnet et al., 2023), ShapeNet Car (Chang et al., 2015), CMOS semiconductor simulations, and SPACIER-M (Wu & Luo, 2025).

For each benchmark, we report DRAFT with and without ranking supervision. Figure 4 shows where the two variants allocate fine-grained computation, while Figure 5 quantifies the alignment between the selected nodes and the corresponding ranking targets. Complete definitions, sample-level statistics, and additional discussion are provided in Appendix B.

AirFRANS. Table 1 reports results on the AirFRANS benchmark. Both DRAFT variants outperform Transolver and SpiderSolver across all reported metrics, with DRAFT without ranking supervision achieving the strongest overall performance. Compared with SpiderSolver (Qi et al., 2025), it reduces the Vol, Surf, and C_L errors by approximately 88.2%, 72.1%, and 77.2%, respectively, while increasing the Spearman correlation ρ_L from 0.9988 to 0.9998.

Ranking supervision substantially increases alignment with the normalized local-error target (Figure 5), but shifts part of the fixed budget from the airfoil surface toward wake regions with turbulent-viscosity errors (Figure 4). Because the target averages errors across normalized output channels, these residuals can compete with surface-pressure errors despite their different physical and metric relevance. The more surface-focused no-ranking allocation is therefore better aligned with Surf and C_L, explaining its stronger performance.

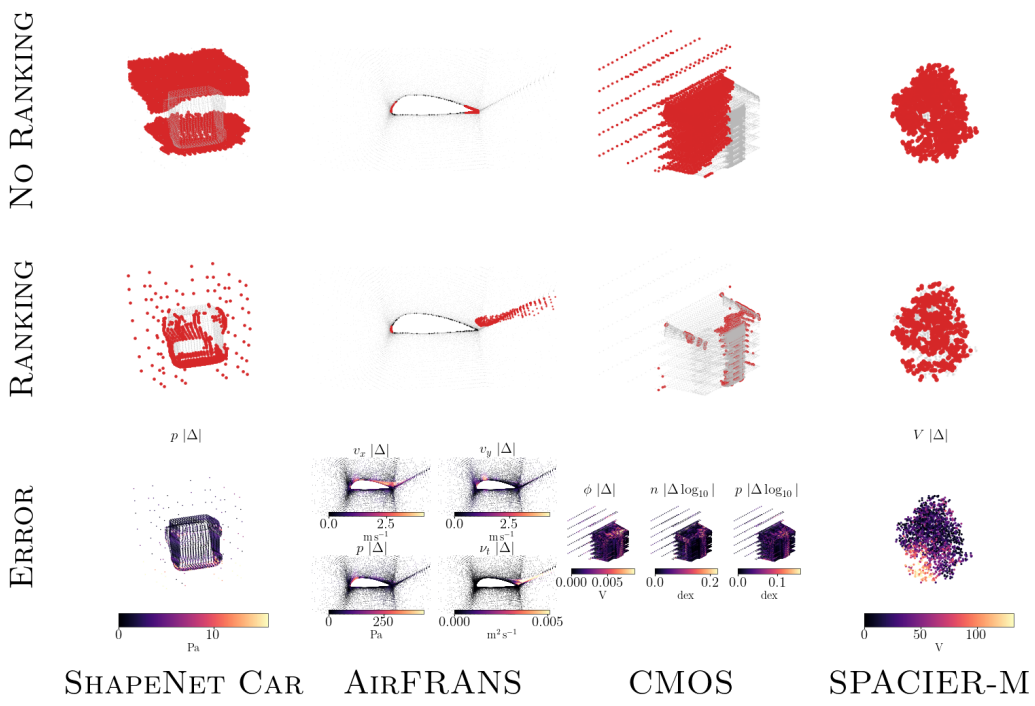

Figure 4: Fine-grained allocation across the four benchmarks. Rows show DRAFT without ranking supervision, with ranking supervision, and post-hoc error maps. Low-opacity and highlighted points denote unselected and selected nodes, respectively. Masks are inferred from the input and reused across blocks; errors are unavailable at inference time.

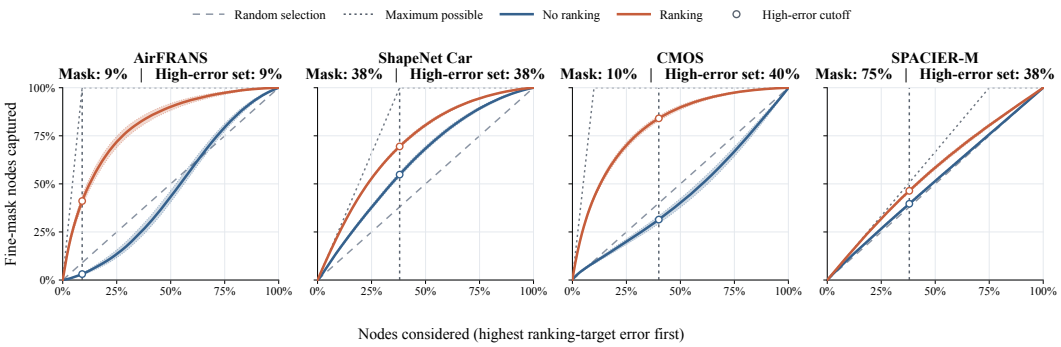

Figure 5: Alignment between the fine mask and ranking target. Curves show the fraction of selected nodes recovered when inspecting nodes by decreasing target error. The dashed diagonal denotes random selection, the dotted curve the budget-constrained maximum, and markers the high-error cutoff. Shading shows 95% bootstrap confidence intervals; definitions are provided in Appendix B.

ShapeNet Car. Results on ShapeNet Car are reported in Table 2. DRAFT with ranking supervision achieves the best performance on both field-level metrics, Vol and Surf. Relative to the no-ranking variant, ranking reduces Vol from 0.0218 to 0.0172 and Surf from 0.0889 to 0.0737, corresponding to relative improvements of approximately 21.1% and 17.1%, respectively. SpiderSolver remains stronger on the drag-related metrics C_D and ρ_D.

Ranking supervision concentrates selected nodes within high-error, pressure-critical surface regions (Figures 5 and 4), aligning the allocation with the Surf metric. Unlike SpiderSolver (Qi et al., 2025),

Table 1: Comparison on AirFRANS. Lower is better for Vol, Surf, and C_L; higher is better for ρ_L.

Method	Vol ↓	Surf ↓	$\mathbf{C_L}$ ↓	$\boldsymbol{\rho_L}$ ↑
Transolver (Wu et al., 2024)	0.0025	0.0080	0.1026	0.9977
SpiderSolver (Qi et al., 2025)	0.0017	0.0043	0.0741	0.9988
DRAFT (no ranking loss) **(ours)**	**0.0002**	**0.0012**	**0.0169**	**0.9998**
DRAFT (ranking loss) **(ours)**	0.0004	0.0028	0.0291	0.9997

Table 2: Comparison on ShapeNet Car. Lower is better for Vol, Surf, and C_D; higher is better for ρ_D.

Method	Vol ↓	Surf ↓	$\mathbf{C_D}$ ↓	$\boldsymbol{\rho_D}$ ↑
Transolver (Wu et al., 2024)	0.0228	0.0793	0.0129	0.9916
SpiderSolver (Qi et al., 2025)	0.0210	0.0738	**0.0100**	**0.9928**
DRAFT (no ranking loss) **(ours)**	0.0218	0.0889	0.0151	0.9883
DRAFT (ranking loss) **(ours)**	**0.0172**	**0.0737**	0.0123	0.9917

this surface-focused allocation is learned without prescribing surface or near-surface regions and without an additional surface-pressure loss.

CMOS Semiconductor Simulations. Table 3 shows that both DRAFT variants outperform the attention-based baselines across electrostatic potential, electron density, and hole density prediction. DRAFT without ranking supervision achieves the best value for every reported metric, demonstrating that the dual-scale architecture effectively captures the coupled three-dimensional interactions governing semiconductor-device behaviour.

The no-ranking variant distributes fine-grained computation across the device volume and interface-adjacent regions, whereas ranking concentrates the same budget on fewer locally difficult nodes (Figures 4 and 5). Since its target aggregates normalized errors across differently transformed outputs, this concentration does not necessarily preserve their relative physical and metric importance, consistent with the higher errors of the ranking variant.

SPACIER-M. Table 4 reports results on SPACIER-M. Following Wu & Luo (2025), DRAFT uses an absolute-charge-weighted prediction objective. Although it does not perform message passing, we retain local-environment information by summarizing $15\,\text{Å}$ neighbourhoods as fixed node features during preprocessing (using their minimum, maximum, mean, and standard deviation), using the best-performing cutoff from Wu & Luo (2025, Figure 3b). DRAFT therefore obtains local environment information without storing or traversing the corresponding edges during training or inference.

DRAFT without ranking supervision achieves the best R^2 and MAE among the considered methods while remaining competitive in MAPE. Figure 5 shows only a limited difference between the ranking and no-ranking allocations on SPACIER-M. One contributing factor is the large fine-attention budget, which already covers approximately 75% of the atoms and therefore limits how strongly the selected set can be concentrated within the 38% high-error subset (see Appendix B). More importantly, the ranking target is a local absolute-charge-weighted residual, whereas solvation energy is a signed global sum (Wu & Luo, 2025). Local magnitudes do not represent the signs and cancellation of atomic contributions, explaining why ranking does not improve the benchmark metrics.

When is ranking supervision beneficial? The main limitation of the current ranking objective is that node-wise errors are computed in normalized space and averaged across outputs. Equal normalized residuals therefore receive equal priority, even when their physical magnitudes or downstream relevance differ. On AirFRANS, for example, turbulent-viscosity errors in the wake can become comparable to surface-pressure errors after normalization, despite having a smaller physical influence on Surf and C_L; a related ambiguity arises across the differently transformed CMOS outputs. This limitation is not inherent to multi-output prediction: ranking remains applicable when the output channels have comparable physical relevance.

Table 3: CMOS results using mean aggregation. Lower is better. MGRE and MaxRE denote mean global and maximum relative error, respectively.

Output	Method	MAE ↓	MaxAE ↓	MRE ↓	MSE ↓	RMSE ↓
φ	MHAT-GNN (Jang et al., 2023)	0.0058	0.0998	0.0575	8.24×10^{-5}	0.0087
	OHAT Attention v1 (Kovani et al., 2026)	0.0044	0.0688	0.0330	4.38×10^{-5}	0.0063
	DRAFT (no ranking loss) **(ours)**	**0.0017**	**0.0324**	**0.0125**	$\mathbf{6.50\times10^{-6}}$	**0.0024**
	DRAFT (ranking loss) **(ours)**	0.0022	0.0387	0.0169	1.02×10^{-5}	0.0031
Output	**Method**	**LogMAE ↓**	**MRE ↓**	**MGRE ↓**	**MaxRE ↓**	
n	MHAT-GNN (Jang et al., 2023)	0.0843	0.2235	0.0080	65.7600	
	OHAT Attention v1 (Kovani et al., 2026)	0.0738	0.1936	0.0068	95.4300	
	DRAFT (no ranking loss) **(ours)**	**0.0278**	**0.0653**	**0.0021**	**6.5534**	
	DRAFT (ranking loss) **(ours)**	0.0356	0.0840	0.0026	7.0890	
p	MHAT-GNN (Jang et al., 2023)	0.0840	0.2099	0.0004	45.0900	
	OHAT Attention v1 (Kovani et al., 2026)	0.0708	0.1723	0.0003	10.6600	
	DRAFT (no ranking loss) **(ours)**	**0.0281**	**0.0660**	**0.0001**	**2.7362**	
	DRAFT (ranking loss) **(ours)**	0.0353	0.0832	0.0002	3.2610	

Table 4: Comparison on SPACIER-M. Higher is better for R^2; lower is better for MAE and MAPE.

Method	$\mathbf{R^2}$ ↑	MAE ↓	MAPE ↓
PB-U-Net (Wu & Luo, 2025)	0.931	481.947	10.268
PB-FNO (Wu & Luo, 2025)	0.939	513.808	11.649
PB-GNN (Wu & Luo, 2025)	0.995	151.540	**3.822**
DRAFT (no ranking loss) **(ours)**	**0.998**	**137.768**	5.437
DRAFT (ranking loss) **(ours)**	0.995	203.710	7.080

Once the relative importance of the outputs is preserved, ranking is most appropriate for local or region-specific objectives, where difficult nodes identify where refinement is useful, as on ShapeNet Car. It is less naturally aligned with global quantities determined by distributed, signed, or cancelling contributions, as for the SPACIER-M solvation energy. Thus, ranking is beneficial when its local target preserves both the cross-output physical hierarchy and the spatial structure of the downstream metric; otherwise, direct end-to-end allocation or a metric-aware ranking target is preferable. All the details related to the reproducibility of the experiments are in Table 5 of the Appendix B.

6 Conclusion

We introduced DRAFT, a dual-scale architecture for surrogate modeling on large unstructured meshes that combines global exchange through latent tokens with sparse attention over a learned, fixed-budget node subset. Both routing mechanisms are trained end-to-end without predefined regions or masks. Across four benchmarks in aerodynamics, semiconductor-device simulation, and biomolecular electrostatics, DRAFT performs strongly against graph-, neural-operator-, and transformer-based baselines in both boundary- and bulk-driven regimes. Its learned masks also reveal where fine-grained computation is allocated. Ranking supervision further improves allocation when its local error target aligns with the downstream objective, as on ShapeNet Car. Overall, DRAFT provides a flexible framework for learning computational allocation in neural PDE surrogates.

AI use statement

We used LLMs to assist with drafting and revising portions of the manuscript, including improving clarity, readability, and presentation. The authors reviewed and edited the AI-assisted text and checked its consistency with the methods and experimental results reported in the paper. The authors take full responsibility for the final manuscript, including all scientific claims and conclusions.

Ethics statement

This work investigates surrogate modeling for physical simulations in aerodynamics, semiconductor devices, and biomolecular electrostatics. The reported experiments do not involve human participants or personal data. As with other learned physical surrogates, predictions may be unreliable outside the conditions covered by the evaluation. Applications in safety-critical scientific or engineering workflows therefore require independent validation against appropriate numerical solvers or experimental measurements.

Reproducibility statement

The architecture and learning objectives are described in Section 3.1. Appendix A provides additional implementation details and pseudocode for the complete forward pass and the coarse- and fine-grained attention pathways. Dataset-specific token counts, fine-attention budgets, and ranking-supervision settings are reported in Table 5. The experimental section describes the datasets, prediction targets, and evaluation metrics, while Appendix B details the analysis of learned computational allocation.

References

Benedikt Alkin, Andreas Fürst, Simon Schmid, Lukas Gruber, Markus Holzleitner, and Johannes Brandstetter. Universal physics transformers: A framework for efficiently scaling neural operators, 2025. URL https://arxiv.org/abs/2402.12365.

Nathan Baker, David Sept, Simpson Joseph, Michael Holst, and J. McCammon. Electrostatics of nanosystems: Application to microtubules and the ribosome. *Proceedings of the National Academy of Sciences*, 98:10037–10041, 08 2001. doi: 10.1073/pnas.181342398.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation, 2013. URL https://arxiv.org/abs/1308.3432.

Florent Bonnet, Ahmed Jocelyn Mazari, Paola Cinnella, and Patrick Gallinari. AirfRANS: High fidelity computational fluid dynamics dataset for approximating Reynolds-Averaged Navier-Stokes solutions, 2023. URL https://arxiv.org/abs/2212.07564.

Johannes Brandstetter, Daniel E. Worrall, and Max Welling. Message passing neural pde solvers. In *International Conference on Learning Representations*, 2022.

Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An information-rich 3D model repository, 2015. URL https://arxiv.org/abs/1512.03012.

Jingyang Deng, Xingjian Li, Haoyi Xiong, Xiaoguang Hu, and Jinwen Ma. Geometry-guided conditional adaption for surrogate models of large-scale 3d PDEs on arbitrary geometries, 2024. URL https://openreview.net/forum?id=EyQO9RPhwN.

Federico Fogolari, Alessandro Brigo, and Henriette Molinari. The Poisson-Boltzmann equation for biomolecular electrostatics: a tool for structural biology. *Journal of Molecular Recognition*, 15: 377–392, 12 2002. doi: 10.1002/jmr.577.

Wonik Jang, Sanghoon Myung, Jae Myung Choe, Young-Gu Kim, and Dae Sin Kim. TCAD device simulation with graph neural network. *IEEE Electron Device Letters*, 44(8):1368–1371, 2023. doi: 10.1109/LED.2023.3290930.

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural Operator: Learning Maps Between Function Spaces. *arXiv e-prints*, art. arXiv:2108.08481, August 2021. doi: 10.48550/arXiv.2108.08481.

Konstantina Kovani, Greta Chiaravalli, Francesco Di Stefano, Michal Dyrek, Semin Kwak, Vamsi Spandan Arza, Ryota Hachiya, Yonggwi Cho, Hideki Minari, Jun Komachi, Shinya Yamakawa, and Lorenzo Servadei. A deep learning framework for fast, unified process and device TCAD simulation. In *2026 International Conference on Simulation of Semiconductor Processes and Devices (SISPAD)*, Kumamoto, Japan, September 2026. URL https://pub.confit.atlas.jp/en/event/sispad2026/presentation/Session_11-04. To appear.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations, 2020a. URL https://arxiv.org/abs/2003.03485.

Zongyi Li, Nikola B. Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew M. Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *CoRR*, abs/2010.08895, 2020b. URL https://arxiv.org/abs/2010.08895.

Zongyi Li, Daniel Zhengyu Huang, Burigede Liu, and Anima Anandkumar. Fourier Neural Operator with Learned Deformations for PDEs on General Geometries. *arXiv e-prints*, art. arXiv:2207.05209, July 2022. doi: 10.48550/arXiv.2207.05209.

Zongyi Li, Nikola Borislavov Kovachki, Chris Choy, Boyi Li, Jean Kossaifi, Shourya Prakash Otta, Mohammad Amin Nabian, Maximilian Stadler, Christian Hundt, Kamyar Azizzadenesheli, and Anima Anandkumar. Geometry-informed neural operator for large-scale 3d pdes, 2023. URL https://arxiv.org/abs/2309.00583.

William Peebles and Saining Xie. Scalable diffusion models with transformers, 2023. URL https://arxiv.org/abs/2212.09748.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer, 2017. URL https://arxiv.org/abs/1709.07871.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W. Battaglia. Learning mesh-based simulation with graph networks. In *International Conference on Learning Representations*, 2021.

Kai Qi, Fan Wang, Zhewen Dong, and Jian Sun. SpiderSolver: A geometry-aware transformer for solving PDEs on complex geometries. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=hWtvsL51hO.

Louis Serrano, Lise Le Boudec, Armand Kassaï Koupaï, Thomas X Wang, Yuan Yin, Jean-Noël Vittaut, and Patrick Gallinari. Operator learning with neural fields: Tackling PDEs on general geometries. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=4jEjq5nhg1.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A Fast Transformer Solver for PDEs on General Geometries. *arXiv e-prints*, art. arXiv:2402.02366, February 2024. doi: 10.48550/arXiv.2402.02366.

Yongxian Wu and Ray Luo. SPACIER: A dataset for modeling electrostatic Poisson-Boltzmann atomic solvation potentials. In *Machine Learning and the Physical Sciences Workshop at NeurIPS*, 2025.

Zherui Yang, Haiyang Xin, Tao Du, and Ligang Liu. Simple yet effective: Low-rank spatial attention for neural operators. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=PhSw9Xhw6e.

Yanshun Zhao, Xiaoyu Peng, Jiamin Jiang, Congcong Zhu, and Jingrun Chen. MeshTok: Efficient multi-scale tokenization for scalable PDE transformers. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=I8QGwNBO4T.

Weiheng Zhong, Jing Bi, Victor Oancea, and Hadi Meidani. PGD-NO: A neural operator with precomputed geometry decomposition for 3d million-scale physics simulations. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=h5vFCxA37Y.

Hang Zhou, Haixu Wu, Haonan Shangguan, Yuezhou Ma, Huikun Weng, Jianmin Wang, and Mingsheng Long. Transolver-3: Scaling up transformer solvers to industrial-scale geometries, 2026. URL https://arxiv.org/abs/2602.04940.

O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu. *The Finite Element Method: Its Basis and Fundamentals, Sixth Edition*. Butterworth-Heinemann, 6 edition, May 2005. ISBN 0750663200.

A Detailed Architecture and Implementation

This appendix provides additional architectural and implementation details for DRAFT. We first describe the complete forward pass and then detail the coarse-grained token pathway and the fine-grained sparse pathway. Additional details of the ranking objective are provided in Section A.4.

A.1 Overall DRAFT Forward Pass

The token-assignment matrix $\mathbf{A}$, the node-importance scores $\mathbf{s}$, and the fine-grained node set $\mathcal{I}$ are computed from the initial node representations and reused across all L DRAFT blocks. Each block is conditioned by a FiLM transformation before the dual-scale update and by a second FiLM transformation after it. The complete computation is summarized in Algorithm 1.

Algorithm 1 DRAFT forward pass with optional ranking supervision

Require: Node features $\mathbf{X} \in \mathbb{R}^{N \times d_{\text{in}}}$, global parameters $\mathbf{c}$, number of tokens K, fine-attention budget b, number of blocks L, and targets $\mathbf{Y}$ when training

Ensure: Node-wise predictions $\widehat{\mathbf{Y}} \in \mathbb{R}^{N \times d_{\text{out}}}$ and optional ranking loss $\mathcal{L}_{\text{rank}}$

1: $\mathbf{H}^{(0)} \leftarrow \text{Encode}(\mathbf{X})$
2: $\mathbf{S} \leftarrow \text{MLP}_{\text{assign}}(\mathbf{H}^{(0)})$ ▷ $\mathbf{S} \in \mathbb{R}^{N \times K}$
3: $\mathbf{A} \leftarrow \text{softmax}_K(\mathbf{S})$ ▷ Row-wise token probabilities
4: $\mathbf{s} \leftarrow \text{MLP}_{\text{score}}(\mathbf{H}^{(0)})$ ▷ $\mathbf{s} \in \mathbb{R}^N$
5: $M \leftarrow \max(1, \lfloor bN \rfloor)$
6: $\mathcal{I} \leftarrow \text{TopMIndices}(\mathbf{s}, M)$ ▷ $|\mathcal{I}| = M$
7: $m_i^{\text{hard}} \leftarrow \mathbb{1}[i \in \mathcal{I}], \quad i = 1, \ldots, N$
8: $\mathbf{m}^{\text{soft}} \leftarrow \sigma(\mathbf{s})$
9: $\mathbf{m} \leftarrow \mathbf{m}^{\text{hard}} + \mathbf{m}^{\text{soft}} - \text{sg}(\mathbf{m}^{\text{soft}})$ ▷ Hard forward, soft backward
10: **for** $\ell = 0, \ldots, L-1$ **do**
11: $\quad \mathbf{H}_{\text{in}}^{(\ell)} \leftarrow \text{FiLM}_{\text{pre}}^{(\ell)}\left(\mathbf{H}^{(\ell)} \mid \mathbf{c}\right)$
12: $\quad \mathbf{H}_{\text{coarse}}^{(\ell)} \leftarrow \text{CoarseBlock}^{(\ell)}\left(\mathbf{H}_{\text{in}}^{(\ell)}, \mathbf{A}\right)$ ▷ Algorithm 2
13: $\quad \mathbf{H}_{\text{block}}^{(\ell)} \leftarrow \text{FineBlock}^{(\ell)}\left(\mathbf{H}_{\text{coarse}}^{(\ell)}, \mathcal{I}, \mathbf{m}\right)$ ▷ Algorithm 3
14: $\quad \mathbf{H}'^{(\ell)} \leftarrow \mathbf{H}^{(\ell)} + \mathbf{H}_{\text{block}}^{(\ell)}$
15: $\quad \mathbf{H}^{(\ell+1)} \leftarrow \mathbf{H}'^{(\ell)} + \text{MLP}^{(\ell)}\left(\text{FiLM}_{\text{post}}^{(\ell)}\left(\mathbf{H}'^{(\ell)} \mid \mathbf{c}\right)\right)$
16: **end for**
17: $\widehat{\mathbf{Y}} \leftarrow \text{Decode}(\mathbf{H}^{(L)})$
18: **if** ranking supervision is enabled **then**
19: $\quad \mathbf{e} \leftarrow \text{sg}\left[\text{NodeError}(\widehat{\mathbf{Y}}, \mathbf{Y})\right]$
20: $\quad \mathcal{L}_{\text{rank}} \leftarrow \text{PairwiseRankingLoss}(\mathbf{s}, \mathbf{e})$
21: $\quad$ **return** $\widehat{\mathbf{Y}}, \mathcal{L}_{\text{rank}}$
22: **else**
23: $\quad$ **return** $\widehat{\mathbf{Y}}$
24: **end if**

Lines 10–11 show that the outer residual connection $\mathbf{H}'^{(\ell)} \leftarrow \mathbf{H}^{(\ell)} + \mathbf{H}_{\text{block}}^{(\ell)}$ is placed around a quantity, $\mathbf{H}_{\text{block}}^{(\ell)}$, that itself already carries the FiLM-normalized input $\mathbf{H}_{\text{in}}^{(\ell)}$ forward through the coarse pathway's own internal residual connection (Section A.2, Algorithm 2). Consequently, both the pre-normalization state $\mathbf{H}^{(\ell)}$ and its FiLM-normalized counterpart $\mathbf{H}_{\text{in}}^{(\ell)}$ contribute additively to $\mathbf{H}'^{(\ell)}$: the block contains two nested identity paths across the dual-scale attention stage, one bypassing it entirely and one carried forward from the coarse pathway's own residual connection. This is an intentional design choice rather than a notational inconsistency with Eq. equation 2.

A.2 Coarse-Grained Token Pathway

The coarse pathway performs global information exchange through a fixed set of K latent tokens. Node representations are first aggregated into tokens using the soft assignment matrix $\mathbf{A}$. After global self-attention in token space, the same assignment probabilities are reused to route the updated token representations back to the nodes.

The assignment softmax is applied independently to every node:

$$A_{ik} = \frac{\exp(S_{ik})}{\sum_{j=1}^{K} \exp(S_{ij})}, \qquad \sum_{k=1}^{K} A_{ik} = 1. \tag{13}$$

Consequently, each row of $\mathbf{A}$ defines a probability distribution over the K tokens.

The token-mass vector

$$\mathbf{z} = \mathbf{A}^\top \mathbf{1}_N + \epsilon \in \mathbb{R}^K \tag{14}$$

contains the total soft assignment received by each token. In Algorithm 2, $\oslash$ denotes division of every token representation by its corresponding scalar mass, broadcast over the hidden dimension. The small constant ϵ prevents numerical instability when a token receives negligible assignment mass.

The complete coarse-grained operation is reported in Algorithm 2.

The k-th token is

$$\mathbf{T}^{(\ell)}_{k:} = \frac{\sum_{i=1}^{N} A_{ik} \mathbf{H}^{(\ell)}_{\text{in},i:}}{\sum_{i=1}^{N} A_{ik} + \epsilon}. \tag{15}$$

The matrix $\widetilde{\mathbf{T}}^{(\ell)} \in \mathbb{R}^{K \times h}$ contains the globally updated tokens, while $\mathbf{A}\widetilde{\mathbf{T}}^{(\ell)} \in \mathbb{R}^{N \times h}$ routes their information back to the nodes.

Algorithm 2 Coarse-grained token-attention block

Require: Node representations $\mathbf{H} \in \mathbb{R}^{N \times h}$, assignment probabilities $\mathbf{A} \in \mathbb{R}^{N \times K}$, block index ℓ
Ensure: Coarse-updated representations $\mathbf{H}_{\text{coarse}} \in \mathbb{R}^{N \times h}$

$\mathbf{z} \leftarrow \mathbf{A}^\top \mathbf{1}_N + \epsilon$ ▷ Soft token masses
$\mathbf{T} \leftarrow \left(\mathbf{A}^\top \mathbf{H}\right) \oslash \mathbf{z}$ ▷ $\mathbf{T} \in \mathbb{R}^{K \times h}$
$\widetilde{\mathbf{T}} \leftarrow \mathrm{TokenAttention}^{(\ell)}(\mathbf{T})$ ▷ Global interaction among tokens
$\mathbf{R} \leftarrow \mathbf{A}\widetilde{\mathbf{T}}$ ▷ $\mathbf{R} \in \mathbb{R}^{N \times h}$
$\Delta\mathbf{H}_{\text{coarse}} \leftarrow \mathrm{MLP}^{(\ell)}_{\text{coarse}}(\mathbf{R})$
$\mathbf{H}_{\text{coarse}} \leftarrow \mathbf{H} + \Delta\mathbf{H}_{\text{coarse}}$ ▷ Residual coarse update
return $\mathbf{H}_{\text{coarse}}$

The node-to-token aggregation and token-to-node routing each require $\mathcal{O}(NK)$ interactions, while self-attention among the K tokens requires $\mathcal{O}(K^2)$. The resulting global interaction cost is therefore $\mathcal{O}(NK + K^2)$, excluding linear projections.

A.3 Fine-Grained Sparse Pathway

The fine pathway applies node-level self-attention only to the $M = \max(1, \lfloor bN \rfloor)$ nodes selected by the scoring network. The hard index set $\mathcal{I}$ determines which node representations are gathered in the forward pass. The straight-through mask $\mathbf{m}$ preserves hard routing while providing a differentiable surrogate for the node-scoring network.

At block ℓ, the selected coarse representations are gathered into an $M \times h$ tensor and processed using fine-grained self-attention. The resulting residual updates are weighted by the restricted straight-through mask $\mathbf{m}_{\mathcal{I}}$ and scattered back to their original node positions. All positions outside $\mathcal{I}$ are assigned zero fine-grained updates.

The complete operation is summarized in Algorithm 3.

Algorithm 3 Fine-grained sparse-attention block

Require: Coarse-updated node representations $\mathbf{H}_{\text{coarse}} \in \mathbb{R}^{N \times h}$, selected-node indices $\mathcal{I} \subseteq \{1, \ldots, N\}$, straight-through mask $\mathbf{m} \in \mathbb{R}^N$, block index ℓ
Ensure: Dual-scale representations $\mathbf{H}_{\text{block}} \in \mathbb{R}^{N \times h}$

$\mathbf{H}_{\mathcal{I}} \leftarrow \mathrm{Gather}\left(\mathbf{H}_{\text{coarse}}, \mathcal{I}\right)$ ▷ $\mathbf{H}_{\mathcal{I}} \in \mathbb{R}^{M \times h}$
$\Delta\mathbf{H}_{\mathcal{I}} \leftarrow \mathrm{FineAttention}^{(\ell)}\left(\mathbf{H}_{\mathcal{I}}\right)$ ▷ Residual updates for selected nodes
$\mathbf{m}_{\mathcal{I}} \leftarrow \mathrm{Gather}\left(\mathbf{m}, \mathcal{I}\right)$
$\Delta\mathbf{H}_{\text{fine}} \leftarrow \mathrm{Scatter}\left(\mathbf{m}_{\mathcal{I}} \odot \Delta\mathbf{H}_{\mathcal{I}}, \mathcal{I}, N\right)$ ▷ Zero outside $\mathcal{I}$
$\mathbf{H}_{\text{block}} \leftarrow \mathbf{H}_{\text{coarse}} + \Delta\mathbf{H}_{\text{fine}}$
return $\mathbf{H}_{\text{block}}$

Since $\mathbf{m}_{\mathcal{I}} = \mathbf{1}$ in the forward pass, the selected nodes receive the complete fine-grained residual updates. During backpropagation, gradients with respect to the importance scores are propagated

through the sigmoid component of the straight-through mask. Nodes outside $\mathcal{I}$ retain only the coarse-pathway output.

Fine-grained self-attention is evaluated over M nodes rather than all N nodes, reducing the corresponding node-level interaction cost from $\mathcal{O}(N^2)$ to $\mathcal{O}(M^2)$.

The ranking objective changes only how the node-scoring network is supervised during training; it does not modify the inference-time topology. At inference time, both variants compute the importance scores $\mathbf{s}$ from the input representations, select the corresponding top-M nodes, and execute the same coarse- and fine-grained pathways.

A.4 Ranking-Supervision Details

The primary end-to-end objective is the mean squared error or a task-dependent weighted variant. For the standard unweighted case,

$$\mathcal{L}_{\text{e2e}} = \frac{1}{N d_{\text{out}}} \sum_{i=1}^{N} \sum_{r=1}^{d_{\text{out}}} \left(\widehat{Y}_{ir} - Y_{ir} \right)^2, \tag{16}$$

where $\widehat{\mathbf{Y}}$ and $\mathbf{Y}$ denote the predicted and ground-truth physical fields, respectively.

For ranking supervision, the detached prediction error associated with node i is computed in normalized output space. Letting $\widetilde{Y}_{ir}$ and $\widehat{\widetilde{Y}}_{ir}$ denote the normalized target and prediction for output channel r of node i (Section B), it is defined as

$$e_i = \text{sg}\left[\frac{1}{d_{\text{out}}} \sum_{r=1}^{d_{\text{out}}} \left(\widehat{\widetilde{Y}}_{ir} - \widetilde{Y}_{ir} \right)^2 \right], \tag{17}$$

where $\text{sg}(\cdot)$ prevents gradients from propagating through the ranking targets.

Hard nodes are identified from the upper tail of the detached node-error distribution:

$$\mathcal{H} = \{ i \in \{1, \dots, N\} \mid e_i \geq Q_\tau(\mathbf{e}) \}, \tag{18}$$

where $Q_\tau(\mathbf{e})$ denotes the τ-quantile of the node errors.

We sample P pairs (a_p, b_p) according to

$$a_p \sim \text{Uniform}(\mathcal{H}), \qquad b_p \sim \text{Uniform}(\{1, \dots, N\}), \qquad p = 1, \dots, P. \tag{19}$$

For every sampled pair, we compute the error difference and normalized score difference:

$$\Delta e_p = e_{a_p} - e_{b_p}, \qquad \Delta \bar{s}_p = \frac{s_{a_p} - s_{b_p}}{\text{Std}(\mathbf{s}) + \epsilon}. \tag{20}$$

These quantities are substituted into the pairwise hard-margin objective in Eq. equation 12. The resulting loss encourages the scoring network to assign larger importance scores to nodes with larger detached prediction errors.

Finally, the optional score regularization term is

$$\mathcal{L}_{\text{reg}} = \frac{1}{N} \sum_{i=1}^{N} s_i^2, \tag{21}$$

which limits the magnitude of the unnormalized importance scores.

B Additional Analysis of Ranking-Based Allocation

This section provides additional details on how ranking supervision changes the learned fine-grained allocation. The ranking objective modifies only the supervision of the node-scoring network: within each dataset, the ranking and no-ranking variants use the same architecture and fine-attention budget. At inference time, both variants predict the importance scores from the input representations, select the corresponding top-M nodes, and execute the same coarse- and fine-grained pathways. Prediction errors are therefore unavailable when the masks are constructed.

Table 5: Dataset-specific architectural and ranking-allocation settings. K is the number of coarse tokens, b is the fine-attention budget, and the high-error set contains approximately the upper $1-\tau$ fraction of nodes. All ranking variants use $P = 2048$ pairs, margin $\mu = 1$, $\lambda_{\text{rank}} = 1$, and $\lambda_{\text{reg}} = 10^{-6}$.

Dataset	K	b	τ	Ranking target
AirFRANS	32	0.09	0.91	Normalized multi-output MSE
ShapeNet Car	72	0.38	0.62	Normalized multi-output MSE
CMOS	72	0.10	0.60	Normalized multi-output MSE
SPACIER-M	128	0.75	0.62	Charge-weighted normalized squared error

B.1 Ranking Targets and Fine-Attention Budgets

Let $\widetilde{\mathbf{Y}}$ and $\widehat{\widetilde{\mathbf{Y}}}$ denote the normalized target and prediction, respectively. For AirFRANS, ShapeNet Car, and CMOS, the detached node-wise ranking target is

$$e_i = \text{sg}\left[\frac{1}{d_{\text{out}}}\sum_{r=1}^{d_{\text{out}}}\left(\widehat{\widetilde{Y}}_{ir} - \widetilde{Y}_{ir}\right)^2\right], \tag{22}$$

where $\text{sg}(\cdot)$ denotes stop-gradient. For SPACIER-M, which has a single potential output, the same residual is weighted by the absolute atomic charge:

$$e_i^{\text{SPACIER}} = |q_i|\ \text{sg}\left[\left(\widehat{\widetilde{Y}}_i - \widetilde{Y}_i\right)^2\right]. \tag{23}$$

Here, $\widetilde{Y}_i$ and $\widehat{\widetilde{Y}}_i$ denote the normalized target and predicted atomic reaction-field potential $\phi_{\text{rxn}}(\mathbf{r}_i)$, respectively, from which the electrostatic solvation free energy is computed (Eq. equation 30).

The hard-node set used by the pairwise ranking objective is

$$\mathcal{H} = \{i \mid e_i \geq Q_\tau(\mathbf{e})\}, \tag{24}$$

where τ is the dataset-specific error quantile. The corresponding hard-error fraction is approximately $1-\tau$. Table 5 summarizes the ranking targets, fine-attention budgets, and hard-error fractions used in the experiments.

For the no-ranking models, the corresponding error definition is used only during post-hoc analysis. Applying the same dataset-specific target to both variants allows their learned masks to be compared against a common type of signal, while each mask is evaluated using the residuals produced by its own model.

B.2 Cumulative-Capture Analysis

To quantify the relation between fine-mask selection and prediction error, we use the cumulative-capture curves shown in Figure 5. Let $\mathcal{I}_v$ denote the node set selected by model variant v, and let

$$b_v = \frac{|\mathcal{I}_v|}{N} \tag{25}$$

be its fine-attention budget. For a fraction $x \in [0,1]$, let $\mathcal{E}_v(x)$ contain the $\lfloor xN \rfloor$ nodes with the largest ranking-target errors produced by the same model. We define the cumulative capture as

$$C_v(x) = \frac{|\mathcal{I}_v \cap \mathcal{E}_v(x)|}{|\mathcal{I}_v|}. \tag{26}$$

Thus, $C_v(x)$ measures the fraction of fine-mask nodes encountered after inspecting the highest-error fraction x of the domain. For a randomly selected mask,

$$\mathbb{E}\left[C_{\text{random}}(x)\right] = x, \tag{27}$$

which gives the diagonal baseline in Figure 5. The maximum capture permitted by a budget b_v is

$$C_{\max}(x) = \min\left(\frac{x}{b_v}, 1\right). \tag{28}$$

This maximum is dataset dependent because the mask budgets differ. When $x < b_v$, even a perfectly error-aligned mask cannot be completely recovered because fewer than $|\mathcal{I}_v|$ nodes have been inspected. When $x \geq b_v$, complete capture is theoretically possible.

The vertical line in each panel marks the training-defined hard-error fraction $1 - \tau$. The corresponding marker therefore reports the fraction of selected fine-mask nodes lying within the hard-error set. This distinction is particularly important for SPACIER-M, where the fine mask covers approximately 75% of the nodes but the hard-error set contains only 38%.

Curves are computed independently for each validation sample and then averaged across paired sample indices shared by the two variants. The shaded regions denote 95% bootstrap confidence intervals obtained by resampling validation samples rather than individual nodes. Each mask is evaluated against the final residuals produced by its corresponding model.

These curves measure post-hoc association between selection and prediction error. They do not estimate the causal reduction in error that would result from assigning additional computation to an individual node. In particular, a node may have a large residual because it is intrinsically difficult even after receiving fine-grained attention.

B.3 Dataset-Specific Interpretation

AirFRANS. Ranking supervision produces a substantially stronger concentration of selected nodes within the high-error portion of the domain (Figure 5). This confirms that the auxiliary objective successfully aligns the scoring network with its normalized local-error target. Nevertheless, this stronger alignment does not improve the AirFRANS benchmark metrics.

The qualitative allocation in Figure 4 explains this difference. The ranking-based model devotes part of its fixed budget to the downstream wake, where the post-hoc maps contain localized turbulent-viscosity errors. The no-ranking model instead concentrates more computation around the airfoil surface, leading edge, and forebody. Because Surf and C_L depend more directly on surface and near-surface prediction, the allocation learned through the end-to-end objective is better aligned with the reported aerodynamic quantities. The AirFRANS result therefore distinguishes successful optimization of the ranking target from successful optimization of the final evaluation metrics.

ShapeNet Car. Ranking supervision also increases the concentration of selected nodes within high-error regions on ShapeNet Car. In this case, however, Figure 4 shows that the selected nodes are predominantly located on the vehicle surface and overlap with pressure-critical regions. The ranking target is therefore more closely aligned with the field-level Surf metric than on AirFRANS.

This alignment is reflected in the improved Vol and Surf results of the ranking-based model. Importantly, the surface-focused allocation is learned directly from the training signal: DRAFT does not prescribe a surface mask or explicitly restrict fine-grained computation to surface and near-surface nodes.

CMOS Semiconductor Simulations. The CMOS panel in Figure 5 evaluates a single shared mask against the ranking target obtained by averaging normalized errors across electrostatic potential, electron density, and hole density. Consequently, the curve measures alignment with this aggregate target rather than with the error of any individual output.

The qualitative maps in Figure 4 show that ranking supervision concentrates the fine budget within a smaller collection of locally difficult regions, whereas the no-ranking model maintains broader coverage of the device volume and several interface-adjacent regions. This distinction is relevant because φ, n, and p have different physical roles, transformations, and evaluation metrics. A node can therefore receive high priority because of one normalized output without producing a corresponding improvement in the remaining fields. The uniformly better metrics of the no-ranking variant indicate that alignment with the aggregate local target is not sufficient to obtain the best joint allocation for the three CMOS outputs.

SPACIER-M. The ranking and no-ranking curves are much closer on SPACIER-M than on AirFRANS or ShapeNet Car. This comparison must be interpreted together with the large fine-attention budget. The mask covers approximately 75% of the atoms, whereas the high-error set contains approximately 38%. At the high-error cutoff, the maximum possible fraction of selected nodes contained in the hard set is therefore

$$\frac{0.38}{0.75} \approx 0.51. \tag{29}$$

The corresponding random expectation is 0.38, leaving a substantially smaller range for enrichment than in datasets with lower fine budgets. The limited separation between the two curves is therefore not, by itself, evidence that the mask is uninformative.

A second distinction concerns the downstream quantity. The ranking target is the magnitude of a local absolute-charge-weighted squared potential residual, while the benchmark evaluates the electrostatic solvation free energy

$$\Delta G_{\mathrm{elec}} = \frac{1}{2} \sum_{i=1}^{N} q_i \widehat{\phi}_{\mathrm{rxn}}(\mathbf{r}_i). \tag{30}$$

The final energy is a signed global sum, so atomic contributions may reinforce or cancel one another. Local squared residual magnitudes do not capture these signs or cancellations. Consequently, greater local error alignment is not necessarily equivalent to reducing the error of the global energy functional.

B.4 Limitations and Future Directions

The comparison across datasets reveals two limitations of the current ranking formulation. First, the target is computed in normalized output space. Normalization prevents variables with larger numerical scales from dominating the objective, but it removes information about physical units, downstream sensitivity, and direct contribution to the reported metrics. The ranking loss therefore identifies nodes with large normalized residuals rather than nodes with the largest expected utility from additional computation.

Second, the current scoring network predicts one scalar importance score per node and produces a single fine mask shared across all output fields. This assumes that the same spatial allocation is optimal for every predicted quantity. Such an assumption may be restrictive for multi-output systems such as AirFRANS and CMOS, where different fields can exhibit distinct spatial error structures.

Future work could investigate ranking targets computed in physical space, metric-aware targets weighted by their influence on downstream quantities, and output-specific scoring heads $\{\mathbf{s}^{(r)}\}_{r=1}^{d_{\mathrm{out}}}$. The latter would require a budget-constrained mechanism for combining multiple field-dependent selections: taking their union may exceed the available budget, while directly averaging scores may suppress regions that are important only for a particular output. More generally, future selection objectives should estimate the expected improvement obtained by assigning computation to a node, rather than relying exclusively on its current prediction residual.

C Fine-Mask Consistency Across Validation Samples

Figure 4 in the main paper visualizes the learned allocation for one representative sample from each benchmark. To assess whether the observed spatial patterns persist across different geometries and physical configurations, we additionally visualize the fine masks for the first twelve validation samples of each dataset and model variant.

Figures 6–9 show the complete selected sets using the same visualization conventions as the main figure. Selected fine-attention nodes are highlighted in red, while the remaining domain is shown in black with reduced opacity. Within each dataset, the ranking and no-ranking variants use the same fine-attention budget. The masks are inferred independently for every sample from its input representations and are subsequently reused across all DRAFT blocks.

Across samples, the grids reveal consistent dataset-dependent allocation strategies. On AirFRANS, the no-ranking model repeatedly concentrates computation near the airfoil and leading-edge region, whereas ranking supervision allocates a larger fraction of the budget to downstream wake structures.

On ShapeNet Car, ranking consistently increases the concentration of selected nodes on the vehicle surface. For CMOS, the no-ranking model exhibits broader volumetric and interface-adjacent coverage, while ranking produces more localized allocations. Finally, both SPACIER-M variants select a broadly distributed collection of atoms, consistently with the substantially larger fine-attention budget used on this dataset. These observations indicate that the allocation patterns discussed in the main paper are not restricted to the individual examples shown in Figure 4.

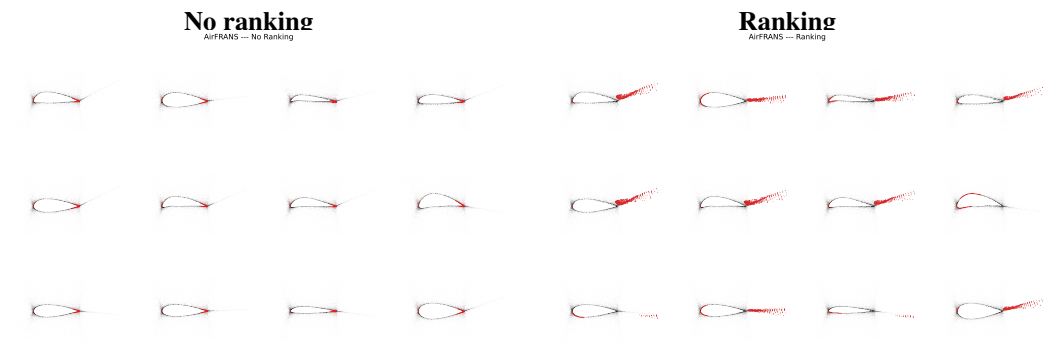

Figure 6: Fine-mask allocations for twelve AirFRANS validation samples. Without ranking supervision, selected nodes remain concentrated predominantly around the airfoil and leading-edge region. Ranking supervision systematically redirects part of the fixed budget toward downstream wake structures.

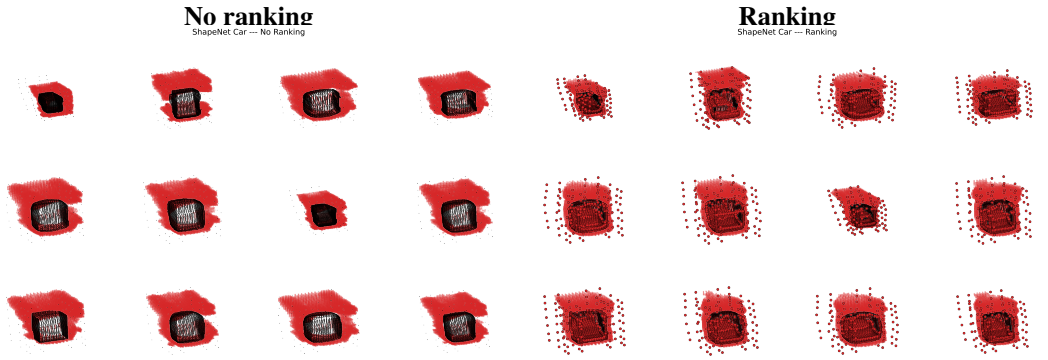

Figure 7: Fine-mask allocations for twelve ShapeNet Car validation samples. Ranking supervision consistently increases the concentration of selected nodes on the vehicle surface, whereas the no-ranking variant allocates a larger fraction of its budget to surrounding volume nodes. Surface-selected nodes are emphasized by larger markers with black edges.

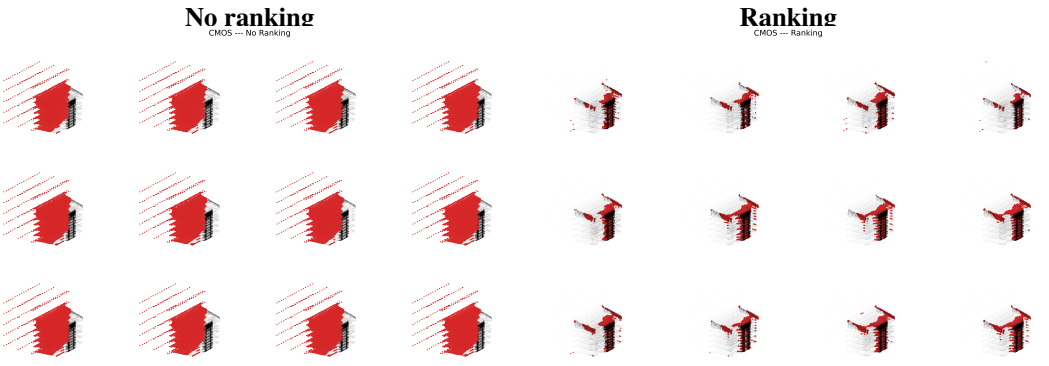

Figure 8: Fine-mask allocations for twelve CMOS validation samples. The no-ranking model distributes fine-grained computation across a broader portion of the device volume and multiple interface-adjacent regions, whereas ranking supervision produces more spatially concentrated selections.

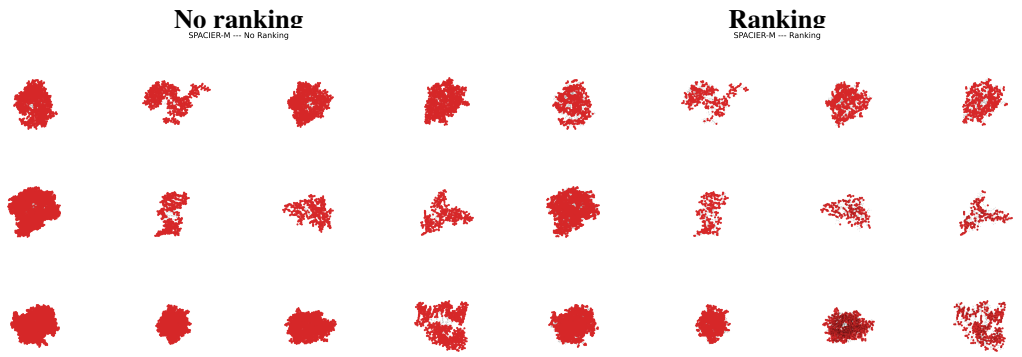

Figure 9: Fine-mask allocations for twelve SPACIER-M validation samples. Both variants produce broadly distributed selections across the molecular systems, consistently with the fine-attention budget of approximately 75%. Ranking supervision produces comparatively limited changes in the overall spatial coverage.