Learning to Solve PDEs on Neural Shape Representations

Supplementary Material

1. Ablation and Design Analysis.

Each ablation isolates a single component of our method: we modify one factor at a time while keeping all other settings strictly identical (same dataset splits, schedules, and hyperparameters). This ensures that the observed variations can be attributed to the component being modified. Because ablations are run separately, with potentially different model variants, the numerical values reported across different tables are not meant to be compared to each other, only within each table. *Our main empirical findings are:*

- Normal-consistency term improves stability; $\alpha = 10^{-2}$ performs best, though MSE-only already yields good results.
- Positions and normals are the essential surface features.
- A local band size of $k \approx 400$ offers the best accuracy–runtime trade-off.
- Very shallow MLPs (2 layers) are sufficient.
- A learnable λ improves accuracy.
- Patch overlap has negligible effect; local behavior dominates.

1.1. Effect of the Normal-Consistency Loss

We first assess the impact of the normal-consistency term L_{NC}, which enforces invariance of the predicted function along surface normals. From Table 1, we observe that adding the normal-consistency term L_{NC} consistently improves the accuracy of the predicted solutions on both the analytical sphere and the Star Fruit surface (evaluated against SFEM as ground truth). In particular, the configuration $L = L_{\mathrm{MSE}} + 10^{-2}\, L_{\mathrm{NC}}$ yields the best average performance, achieving the lowest NMAE and NMaxE on both benchmarks. While the L_{MSE}-only setting also produces low errors, it does not explicitly enforce normal invariance, and the model must learn this behavior implicitly from the data. Adding a small L_{NC} term reliably improves consistency and reduces error. Therefore, we retain the mixed formulation with coefficient 10^{-2} as our default choice for all subsequent experiments.

Table 1. Ablation on loss function design. GT is analytical for the Sphere and SFEM for Star Fruit surfaces.

Sphere (GT analytic)		
Loss	NMAE ↓	NMaxE ↓
L_{MSE}	2.29×10^{-2}	$\underline{5.71 \times 10^{-2}}$
$+\,10^{-4}\, L_{\mathrm{NC}}$	$\underline{2.25 \times 10^{-2}}$	5.74×10^{-2}
$+\,10^{-3}\, L_{\mathrm{NC}}$	2.31×10^{-2}	5.88×10^{-2}
$+\,10^{-2}\, L_{\mathrm{NC}}$	$\mathbf{2.08 \times 10^{-2}}$	$\mathbf{5.43 \times 10^{-2}}$
$+\,10^{-1}\, L_{\mathrm{NC}}$	2.33×10^{-2}	5.86×10^{-2}
$+\, L_{\mathrm{NC}}$	2.40×10^{-2}	5.98×10^{-2}
Star Fruit (GT SFEM)		
Loss	NMAE ↓	NMaxE ↓
L_{MSE}	1.54×10^{-2}	4.03×10^{-2}
$+\,10^{-4}\, L_{\mathrm{NC}}$	1.51×10^{-2}	3.99×10^{-2}
$+\,10^{-3}\, L_{\mathrm{NC}}$	1.57×10^{-2}	4.10×10^{-2}
$+\,10^{-2}\, L_{\mathrm{NC}}$	$\mathbf{1.31 \times 10^{-2}}$	$\mathbf{3.56 \times 10^{-2}}$
$+\,10^{-1}\, L_{\mathrm{NC}}$	$\underline{1.34 \times 10^{-2}}$	$\underline{3.63 \times 10^{-2}}$
$+\, L_{\mathrm{NC}}$	1.42×10^{-2}	3.85×10^{-2}

1.2. Surface Features Ablation

Before selecting a final feature set, we evaluate how different geometric cues affect performance. Our architecture is feature-agnostic: in principle, any per-surface descriptor (positional, differential, learned, or otherwise) can be injected into the operator. Modulo training, the operator can adapt to whichever features it receives. To quantify the importance of explicit geometric cues, we remove subsets of surface features (e.g., normals, curvature) and report the resulting degradation in performance. As shown in Table 2, removing positional coordinates or surface normals degrades performance, confirming their importance for capturing local geometry and spatial context. In contrast, mean and Gaussian curvatures have little to no effect and may even introduce noise, slightly reducing accuracy. This suggests that curvature signals are either redundant with the local spatial and normal information, or too noisy to provide additional benefits. Interestingly, the performance drop remains moderate even when most surface features are removed: since the local band is constructed around the surface, it already provides rich geometric context and implicitly encodes high-level information about the shape, which helps maintain reasonable accuracy. In the final model, we therefore retain only positions and normals as surface features.

1.3. Effect of the Local Band Size k

We vary the number of neighbouring band points k per patch to analyse their impact on the performance. As shown in Table 3, increasing the local band size k leads to systematically lower errors for both the analytical sphere and the Star Fruit surface. However, the improvement is marginal. This behavior can be explained by the architecture of our network: in the final aggregation stage, the model performs a convex combination of the band contributions, assigning smaller attention weights to points that are farther from the query. Consequently, distant points have little influence on

Table 2. Ablation on surface features. GT is analytical for the Sphere and SFEM-based for the Star Fruit surface.

Sphere (GT analytic)		
Features	NMAE ↓	NMaxE ↓
All features	$\mathbf{1.94 \times 10^{-2}}$	$\underline{5.20 \times 10^{-2}}$
w/o normals	2.10×10^{-2}	5.42×10^{-2}
w/o curvatures	$\underline{1.98 \times 10^{-2}}$	$\mathbf{5.19 \times 10^{-2}}$
w/o points	2.22×10^{-2}	5.62×10^{-2}
Points only	2.15×10^{-2}	5.44×10^{-2}

Star Fruit (GT SFEM)		
Features	NMAE ↓	NMaxE ↓
All features	$\underline{1.15 \times 10^{-2}}$	$\underline{3.24 \times 10^{-2}}$
w/o normals	1.33×10^{-2}	3.64×10^{-2}
w/o curvatures	$\mathbf{1.08 \times 10^{-2}}$	$\mathbf{3.08 \times 10^{-2}}$
w/o points	1.25×10^{-2}	3.52×10^{-2}
Points only	1.36×10^{-2}	3.68×10^{-2}

the prediction, which limits the benefit of further enlarging the band. Choosing k therefore involves a trade-off between accuracy, computational cost (since larger k increases the number of parameters), and geometric flexibility: k directly constrains the admissible band thickness and the spacing between band points, which are all coupled by a geometric inequality. In practice, we fix $k = 400$, which provides a good balance between precision, stability, and efficiency.

1.4. Network Architecture Ablation

We evaluate the influence of our MLP design by ablating the number of layers and the hidden width. We also tested standard nonlinearities (ReLU, SiLU, GELU) and observed negligible differences, so we keep ReLU for all experiments for consistency. As shown in Table 4, increasing the depth or width of the Query and Local Band MLPs beyond the configuration 2 layers and 64 neurons does not lead to significant improvement. The best trade-off between accuracy and efficiency is achieved with the simplest setting (2 layers, 64 neurons), which already provides sufficient expressivity for both the analytical Sphere and the Star Fruit surface. Deeper or wider architectures slightly overfit and do not generalize better, suggesting that the geometric encoding remains well captured by compact networks.

As shown in Table 5, increasing the depth or width of the Surface Features MLP brings only marginal improvements. The overall trend is consistent across both the analytical Sphere and the Star Fruit surface: moderate configurations (2–3 layers with 64 neurons) already achieve near-optimal performance. This indicates that the local geometric features are relatively low-dimensional and can be effectively captured by shallow networks. Deeper or wider models yield no significant benefit, confirming that compact architectures provide the best balance between expressivity, generalization, and computational cost.

Table 3. Effect of local band size k on Poisson. Sphere uses analytical GT; Star Fruit uses SFEM GT.

Sphere (GT analytic)	NMAE ↓	NMaxE ↓
$k = 25$	3.51×10^{-2}	8.85×10^{-2}
$k = 50$	3.47×10^{-2}	8.68×10^{-2}
$k = 100$	3.46×10^{-2}	8.67×10^{-2}
$k = 150$	3.38×10^{-2}	8.58×10^{-2}
$k = 200$	3.13×10^{-2}	8.09×10^{-2}
$k = 250$	3.10×10^{-2}	7.92×10^{-2}
$k = 300$	3.10×10^{-2}	7.92×10^{-2}
$k = 350$	3.05×10^{-2}	7.84×10^{-2}
$k = 400$	2.96×10^{-2}	7.70×10^{-2}
$k = 450$	2.85×10^{-2}	7.49×10^{-2}
$k = 500$	2.83×10^{-2}	7.33×10^{-2}

Star Fruit (GT SFEM)	NMAE ↓	NMaxE ↓
$k = 25$	2.94×10^{-2}	7.38×10^{-2}
$k = 50$	2.91×10^{-2}	7.34×10^{-2}
$k = 100$	2.88×10^{-2}	7.31×10^{-2}
$k = 150$	2.77×10^{-2}	7.12×10^{-2}
$k = 200$	2.36×10^{-2}	6.21×10^{-2}
$k = 250$	2.24×10^{-2}	5.95×10^{-2}
$k = 300$	2.14×10^{-2}	5.73×10^{-2}
$k = 350$	2.13×10^{-2}	5.68×10^{-2}
$k = 400$	1.93×10^{-2}	5.26×10^{-2}
$k = 450$	1.75×10^{-2}	5.04×10^{-2}
$k = 500$	1.53×10^{-2}	4.58×10^{-2}

Table 4. Ablation on the Query and Local Band MLP architectures (layers × width). Both MLPs share the same structure since their output dimensions must match. Each cell reports NMAE / NMaxE, with all values scaled by 10^{-2}. GT is analytical for the Sphere and SFEM-based for the Star Fruit surface.

Sphere (GT analytic)			
Layers \ Width	32	64	128
2	2.23 / 5.63	**2.08 / 5.43**	2.19 / 5.61
3	2.27 / 5.69	2.24 / 5.68	2.24 / 5.70
4	2.32 / 5.83	<u>2.18 / 5.45</u>	2.24 / 5.77

Star Fruit (GT SFEM)			
Layers \ Width	32	64	128
2	<u>1.36 / 3.67</u>	**1.31 / 3.56**	1.43 / 3.82
3	1.48 / 3.92	1.43 / 3.84	1.41 / 3.81
4	1.55 / 4.07	<u>1.36 / 3.67</u>	1.45 / 3.92

Table 5. Ablation on the surface features MLP architecture (layers $\times$ width). Each cell reports NMAE / NMaxE, with all values scaled by 10^{-2}. GT is analytical for the Sphere and SFEM-based for the Star Fruit surface.

Sphere (GT analytic)			
Layers \ Width	32	64	128
2	2.23 / 5.63	**2.08 / 5.43**	2.19 / 5.61
3	2.27 / 5.69	2.24 / 5.68	2.24 / 5.70
4	2.32 / 5.83	2.18 / 5.45	2.24 / 5.77

Star Fruit (GT SFEM)			
Layers \ Width	32	64	128
2	1.48 / 3.91	1.31 / **3.56**	1.31 / 3.61
3	1.42 / 3.82	1.45 / 3.85	1.41 / 3.77
4	**1.30** / 3.64	1.35 / 3.67	1.36 / 3.65

1.5. Influence of the Learnable Parameter λ

We study the role of the learnable scalar λ that modulates the surface-aware penalty within each patch. As shown in Table 6, introducing the learnable parameter λ slightly improves accuracy on both analytical and SFEM-based benchmarks. This indicates that allowing the network to adapt the strength of the surface-aware penalty provides additional flexibility during training. Keeping λ learnable is therefore beneficial and theoretically consistent: if this term were unnecessary, the optimization would naturally drive λ toward ≈ 1.

Table 6. Ablation on the learnable λ parameter in the attention formulation. GT is analytical for the Sphere and SFEM-based for the Star Fruit surface.

Sphere (GT analytic)		
Configuration	NMAE $\downarrow$	NMaxE $\downarrow$
w/ learnable λ	$\mathbf{2.01 \times 10^{-2}}$	$\mathbf{5.28 \times 10^{-2}}$
w/o λ (fixed to 1)	2.16×10^{-2}	5.59×10^{-2}

Star Fruit (GT SFEM)		
Configuration	NMAE $\downarrow$	NMaxE $\downarrow$
w/ learnable λ	$\mathbf{1.10 \times 10^{-2}}$	$\mathbf{3.15 \times 10^{-2}}$
w/o λ (fixed to 1)	1.34×10^{-2}	3.63×10^{-2}

1.6. Patch Overlap and Aggregation

Finally, we explore the influence of patch overlap in the global reconstruction, varying the overlap ratio and the temperature parameter T used in the convex aggregation weights. The temperature has only a minor effect on accuracy: by comparing the aggregated field against a ground-truth closest point extension, we found that a moderate value ($T \approx 0.5$) provides stable and smooth reconstructions. Since changes in T had negligible numerical impact, we fix $T = 0.5$ in all experiments. The results in Table 7 show that the error remains virtually unchanged as the number of patches increases—from 600 to 1250, both NMAE and NMaxE vary by less than 1%. This stability arises because the operator performs a local convex aggregation: the softmax weights strongly emphasize nearby band samples (because the parameter T seen before is small), while assigning nearly zero weight to distant ones. As a consequence, enlarging the patch neighborhood adds points that contribute negligibly to the update. This saturation is consistent with our observations that the method relies primarily on local neighborhoods.

Table 7. Effect of patch overlap on Poisson (sphere, analytical GT). Increasing the number of patches has negligible impact on accuracy: because the operator performs a convex local aggregation with softmax weights centered around each query, distant samples contribute almost nothing.

Number of Patch	**NMAE** $\downarrow$	**NMaxE** $\downarrow$
600	1.33×10^{-2}	3.19×10^{-2}
650	1.32×10^{-2}	3.17×10^{-2}
1250	1.33×10^{-2}	3.18×10^{-2}

1.7. Choice of Training Functions

We explored several options for selecting a function family to supervise our operator:

- **Sobolev spaces**: Sampling functions from Sobolev spaces would be theoretically natural, since PDE solutions belong to these spaces. However, Sobolev spaces do not offer an explicit basis, making them impractical for local supervision.
- **Laplacian eigenfunctions**: Eigenfunctions form an orthonormal basis of L^2, but they are tied to the global geometry of a specific surface (e.g., spherical harmonics on $\mathbb{S}^2$). Using them would bake in shape-dependent biases, whereas our goal is to generalize beyond the training geometry.
- **Monomials**: We therefore use low-degree monomials $(x, y, z) \longmapsto x^i y^j z^k$. PDE solutions are smooth, and our operator is local, so any smooth function can be well approximated by the first terms of its Taylor expansion. Monomials span exactly this local polynomial space and provide a simple, geometry-agnostic basis that works across shapes and modalities.

2. Comparative Evaluation

2.1. Accuracy & Convergence

In this evaluation, we compare our method against classical solvers in terms of accuracy and runtime. Complementing the Poisson results in Table 1, Table 8 reports analogous results for the heat equation. SFEM benefits from decades of optimization in mesh-based numerical solvers, whereas our approach is still a research prototype. On this idealized benchmark, SFEM attains extremely low errors thanks to perfect mesh geometry and explicit discretization. When given the exact same geometric information (surface samples and normals extracted from the mesh), our method maintains stable accuracy across resolutions, with NRMSE consistently around 7×10^{-3} for the heat equation and NMAE around 1.3×10^{-2} for Poisson. Although not as precise as SFEM on this setting, our operator is resolution-independent, requires no meshing, and relies on a single learned update rule that generalizes across PDEs, shapes and modalities. These results show that even when geometry comes from the mesh—rather than a neural surface—our solver behaves robustly and does not benefit from mesh refinement, unlike classical discretization-based methods. A single outlier appears in the NMaxE metric (on the coarse sphere in Table 1), originating from a configuration that is ill-posed for our method. In this case, the solver receives almost no meaningful surface information: the local patch contains essentially a single isolated surface sample, which prevents the operator from inferring any reliable geometric structure. Unlike in the surface-features ablation (Sec 1.2 of the supplemental), where the narrow band is still well formed and the network can therefore produce good predictions even with limited features, here the band itself is poorly constructed, providing insufficient spatial context for the operator to recover a high-level representation of the underlying surface. Increasing the band thickness alleviates this issue: with a slightly larger band, the outlier NMaxE decreases from 9.14×10^{-1} to 5.26×10^{-1}, while maintaining a good NMAE (around 4.2×10^{-2}).

Table 8. **Heat equation on the sphere (analytic GT).** Error vs. resolution for SFEM, and our method. We report normalized root mean square (NRMSE) error (lower is better); all methods use identical initial condition and evaluation grids.

Solver	Resolution	NRMSE ↓
SFEM	Coarse	2.93×10^{-4}
	Medium	1.50×10^{-6}
	Fine	1.58×10^{-7}
Ours	Coarse	9.46×10^{-3}
	Medium	7.20×10^{-3}
	Fine	7.24×10^{-3}

Table 9 highlights a marked contrast in robustness to mesh resolution between FEM and our method. SFEM exhibits very large error variations when the mesh is coarsened—reaching increases of several orders of magnitude—while our operator remains remarkably stable, with variations typically within only a few percent. This confirms that accuracy is essentially resolution-independent for our approach. Naturally, FEM achieves higher absolute accuracy and lower runtime on this idealized sphere benchmark; its behavior here reflects decades of optimization and exact access to mesh-based geometry. Nonetheless, the comparison demonstrates that our method maintains consistent accuracy across discretizations, offering robustness where traditional solvers may degrade sharply under coarse or irregular meshing.

Table 9. Relative variation of accuracy and runtime for the Poisson equation on the sphere across mesh resolutions. For each solver, the *Very fine* (VF) resolution is used as the reference (0%) for accuracy, while the *Coarse* (C) resolution serves as the reference (0%) for runtime. Mesh resolutions follow the shorthand: C = Coarse, M = Medium, F = Fine, VF = Very Fine. Positive values denote degradation; negative values indicate improvement.

Solver	Res	ΔNMAE	ΔNMaxE	ΔT
SFEM	C	+ 9359%	+ 17884%	0%
	M	+ 483%	+ 1372%	**+ 33%**
	F	+ 155%	+ 388%	**+ 283%**
	VF	0 %	0%	+ 9000%
Ours	C	**+ 97%**	**+ 2686%**	0%
	M	**- 3%**	**- 1%**	+ 400%
	F	**- 1%**	**- 1%**	+ 645%
	VF	0%	0%	**+ 1323%**

2.2. Handling Different Shape Representations

We run our solver on multiple shape encodings (mesh, point cloud, SNS, SDF/occupancy INRs, Gaussian Splatting), using the same local features described above (positions and normals, with optional curvature), illustrated in Figure 1. Evaluation is performed on the steady-state solution, whose ground truth is analytic (given by the mean of the initial condition over the surface). Although a direct comparison is not strictly meaningful across different shapes and representations, similar trends are observed (see Table 10), with SNS yielding the best results—consistent with its superior geometric estimates. Generalization across modalities also indicates robustness to noisy geometric quantities, as different representations provide features of varying quality. Overall, these results highlight that our method performs consistently well across diverse modalities, rather than aiming for direct performance comparison between them.

Table 10. Comparison of different **surface representations** for the heat equation on the steady state. The metric is the Normalized Root Mean Squared Error (NRMSE), computed against the analytic solution.

Representation / Shape	NRMSE ↓
Neural SDF / Camera	2.17×10^{-2}
Overfitted SDF / Max Planck Face	3.03×10^{-2}
Spherical Neural Surface / Armadillo	$\mathbf{1.88 \times 10^{-2}}$
Gaussian Splatting / Snowman	6.15×10^{-2}
Point Cloud / Hat	2.94×10^{-2}
Mesh / Holey Human	1.92×10^{-2}

2.3. Generalization across shapes

Our neural operator, trained once on a single shape, transfers seamlessly to *unseen* shapes and topologies across input modalities. As illustrated in Figure 6 and quantified in Table 11, it closely matches SFEM reference solutions for Poisson across diverse geometries (e.g., organic, CAD parts with sharp transitions, and thin-structure cases) with consistently low NRMSE. This amortized, geometry-conditioned behavior underpins cross-shape generalization. Nonetheless, errors tend to appear near regions where closest points are not unique (e.g., sharp edges, thin parts, near the medial axis), a limitation inherited from the CPM formulation.

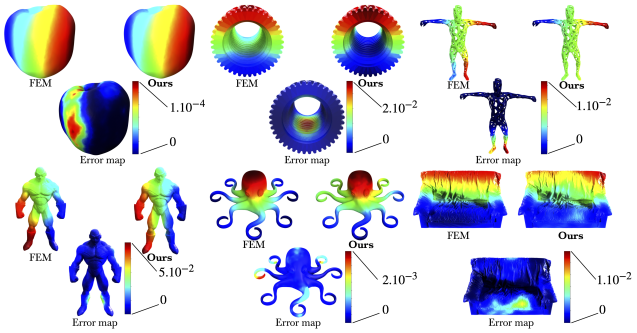

Figure 6. **Comparison to SFEM on diverse shapes.** For each object, left shows SFEM and right shows ours; the small inset below visualizes the pointwise error (ours vs. SFEM) with a hot–cold colormap. See color bar for error scale and the supplemental for per-shape statistics. (Error colormaps are normalized per instance.)

Table 11. Poisson equation results on different shapes (NRMSE ↓). Errors are computed against the FEM as ground truth solution.

Shape	NRMSE	Shape	NRMSE
Jared	5.12×10^{-2}	Sofa	2.65×10^{-2}
Octopus	1.13×10^{-2}	Holey Human	2.67×10^{-2}
Apple	3.20×10^{-3}	Fastener	3.95×10^{-2}

2.4. Runtime scaling

Table 12 shows that the runtime of our method is primarily controlled by the size of the local band rather than by the resolution of the underlying surface. Increasing the number of band samples leads to an increase in cost, whereas changing the mesh from 1k to 100k vertices has only a low impact. Together with the robustness results in Table 9, this confirms that our operator is effectively decoupled from the surface discretization and behaves consistently across meshes of different densities. We note that the reported runtimes include both the neural update and the PDE *solve* step inside the band. The latter is independent of our contribution and could be further optimized using other standard numerical or hardware-specific accelerations.

Table 12. Computation time (in seconds) for the Poisson equation as a function of surface resolution (number of vertices) and band size N. The runtime scales almost only with the number of band samples, while remaining relatively insensitive to changes in mesh resolution. This confirms that our method's complexity is primarily governed by the band size rather than the underlying surface discretization.

Band size	1k vertices	10k vertices	100k vertices
$N = 10k$	3.20s	4.13s	11.8s
$N = 20k$	5.89s	6.75s	15.3s
$N = 30k$	9.32s	10.7s	20.2s
$N = 40k$	16.4s	18.2s	27.6s
$N = 50k$	30.1s	35.6s	67.1s

2.5. Robustness to noise

FEM is more accurate on high-quality fixed meshes; however, it is highly dependent on the fitting of the element mesh to the shape. In the table below, we show that, on the Jared model, the performance of FEM has higher variance with respect to the noise-free version when the input has different scales (σ) of noise.

Table 13. Error of added Gaussian noise to the mesh with different σ, compared to the FEM on the noise-free mesh

Method	FEM	Ours
$\boldsymbol{\sigma = 0.0005}$	$\mathbf{2.39 \times 10^{-2}}$	3.16×10^{-2}
$\boldsymbol{\sigma = 0.001}$	$\mathbf{3.02 \times 10^{-2}}$	3.18×10^{-2}
$\boldsymbol{\sigma = 0.005}$	7.06×10^{-1}	$\mathbf{3.17 \times 10^{-2}}$
$\boldsymbol{\sigma = 0.01}$	8.64×10^{-1}	$\mathbf{3.18 \times 10^{-2}}$

2.6. Robustness to remeshing

To evaluate robustness to surface remeshing, we construct four triangulated discretizations of the unit sphere that share the same number of vertices (N=1500) but differing in their vertex sampling patterns:

- Regular: uniform triangles with Delaunay.
- Random: points are drawn independently from an isotropic 3D Gaussian, then normalized onto the unit sphere
- Jittered: add a small Gaussian-sampled noise to evenly distributed samples and map to the unit sphere.

- Blue-noise: sampling points with a blue noise pattern (from a list of random points, choose the farthest to yield a well-spaced, random distribution).

Table 14 reports the Poisson error for SFEM and for our method, together with the relative variation across mesh types. SFEM is strongly affected by the sampling strategy: its error varies by almost an order of magnitude across the four meshes, reflecting its sensitivity to triangle quality and vertex distribution. In contrast, our method shows negligible variation (within $\pm 3\%$ for NMAE and $\pm 6\%$ for NMaxE), confirming that the learned operator is largely invariant to surface discretization. This robustness stems from the fact that our solver operates in the narrow band with geometry extracted only locally, rather than relying on mesh-dependent differential operators.

Table 14. Robustness to mesh variation for the Poisson equation on the sphere. All meshes contain N=1500 vertices but differ in sampling strategy and triangulation. Errors are computed against the analytical solution. Top: absolute errors for both methods. Bottom: relative variation (min–max) w.r.t. the Regular mesh.

SFEM		
Metric	**Mesh type**	**Error ↓**
NMAE	Regular	6.48×10^{-4}
	Random	$7.09 \times 10^{-3} \pm 1.85 \times 10^{-5}$
	Blue-noise	$2.18 \times 10^{-3} \pm 3.30 \times 10^{-6}$
	Jittered	$1.12 \times 10^{-3} \pm 2.36 \times 10^{-9}$
NMaxE	Regular	1.90×10^{-3}
	Random	$1.03 \times 10^{-2} \pm 1.98 \times 10^{-5}$
	Blue-noise	$3.63 \times 10^{-3} \pm 3.88 \times 10^{-6}$
	Jittered	$3.09 \times 10^{-3} \pm 9.18 \times 10^{-8}$
Ours		
Metric	**Mesh type**	**Error ↓**
NMAE	Regular	1.34×10^{-2}
	Random	$1.30 \times 10^{-2} \pm 4.24 \times 10^{-4}$
	Blue-noise	$1.39 \times 10^{-2} \pm 4.12 \times 10^{-4}$
	Jittered	$1.37 \times 10^{-2} \pm 3.24 \times 10^{-4}$
NMaxE	Regular	3.22×10^{-2}
	Random	$3.21 \times 10^{-2} \pm 2.60 \times 10^{-4}$
	Blue-noise	$3.34 \times 10^{-2} \pm 9.69 \times 10^{-4}$
	Jittered	$3.42 \times 10^{-2} \pm 2.12 \times 10^{-3}$

Relative variation		
Method	ΔNMAE	ΔNMaxE
SFEM	+72% to +994%	+62% to +442%
Ours	**-2% to +3%**	**-0.3% to +6%**

2.7. Generalization on high frequency functions

We evaluate the Poisson equation ($\Delta u = f_\omega$) with sinusoidal input of the form $f_\omega(\mathbf{x}) = \sin(\omega x) \times \sin(\omega y) \times \sin(\omega z)$ of increasing frequency ω on a fixed shape. SFEM is used as reference, and we report NRMSE, see table 15. Our method remains stable as frequency increases and consistently outperforms the spike-only GINO baseline, while achieving better performance than CPM. As expected, performance degrades for higher frequencies due to increased smoothness requirements, but the method retains good accuracy relative to the training regime. We additionally show a qualitative heat example, with the sinusoidal pattern as the initial condition, with $\omega = 10$, shown in figure 4 (main article).

Table 15. High-frequency stress test (NRMSE). Performance degrades with frequency, but our method remains stable and competitive.

Method	$\omega = 1$	$\omega = 10$	$\omega = 20$
GINO (spike-only)	1.36×10^{-1}	2.63×10^{-1}	4.21×10^{-1}
CPM	1.92×10^{-2}	8.64×10^{-2}	5.34×10^{-2}
Ours (spike-only)	1.05×10^{-2}	3.05×10^{-2}	4.10×10^{-2}
GINO (overfitting)	3.33×10^{-3}	6.43×10^{-3}	1.03×10^{-2}

2.8. Dirichlet boundary condition

To generate a controlled open boundary, we cut the Max Planck surface with a single horizontal plane located at the neck region. This creates a well-defined boundary curve $\partial\mathcal{S}$ on which we impose Dirichlet conditions. We test two boundary conditions of increasing difficulty: a simple, constant one, and a more oscillatory *fancy* condition that stresses the solver's ability to handle nontrivial boundary signals. We then solve the surface heat equation

$$\frac{\partial u}{\partial t} = \Delta_{\mathcal{S}} u, \qquad u(t=0) = f,$$

on each truncated surface until reaching the steady state, which we compare against the FEM ground truth.

1. **Constant Dirichlet condition**

$$f = 0, \qquad u\big|_{\partial\mathcal{S}} = 1.$$

2. **Sinusoidal Dirichlet condition**

$$f = -\sin\big(k \arctan(x, z)\big),$$
$$u\big|_{\partial\mathcal{S}} = \sin\big(k \arctan(x, z)\big).$$

The results in Table 16 show that our method remains stable under both boundary regimes, with only a moderate increase in error for the more oscillatory sinusoidal condition. This robustness is expected, since our solver inherits the boundary–condition handling of the original Closest Point Method [39]: Dirichlet values are directly imposed on the band nodes whose closest-point projections lie on the boundary curve $\partial\mathcal{S}$. As a result, the behavior at the boundary is preserved accurately even on complex geometries.

Table 16. Heat equation results on the truncated Max Planck surface. Errors are computed against the FEM as the ground truth.

Boundary Conditions type	**NRMSE** $\downarrow$
standard	2.85×10^{-2}
fancy	3.40×10^{-2}

3. Notations

General maths:
- $\Delta_{\mathcal{S}}$: Laplace Beltrami operator.
- Δ: standard laplacian.
- $\nabla_{\mathcal{S}}$: surface gradient.
- ∇: standard gradient.
- $\mathbb{S}^2$: unit sphere of dim 2.
- ε: band thickness.
- Δx: grid points spacing.
- T: temperature parameter for global aggregation.
- dt: infénitésimal time step element.
- $\langle \cdot, \cdot \rangle$: dot product.
- $\mathrm{dist}(y, \mathcal{S})$: distance from a point y to a surface $\mathcal{S}$.

Method:
- $\mathcal{S}$: surface representation.
- $(\mathbf{n}(x), \mathbf{t}_1(x), \mathbf{t}_2(x))$: normal and principal curvature directions at $x \in \mathcal{S}$. $\mathbf{t}_1$ corresponds to the maximum curvature direction.
- G: cartesian grid with a grid points spacing Δx not specified in the notation.
- $\mathcal{B}_{\mathcal{S}}$: band coordinates around $\mathcal{S}$ with tickness ε not specified in notation.
- U: function define on the entire band coords. $\tilde{U}$ is same but also constant along the normals. Can also add underscore t to say function at time t (U_t).
- $u^i \in \mathbb{R}^k$: function define on the local part $\mathcal{B}_i$ of the band coords. $\tilde{u}^i$ is same but also constant along the normals. Can also add underscore t to say function at time t (u^i_t).
- k: number of points in the local band.

Patch:
- p_i^c: center of the patch i. $p_i^c \in \mathcal{S}$.
- $\mathcal{L}_i$: local frame (i.e, $\mathcal{L}_i = (p_i^c, \mathbf{n}(p_i^c), \mathbf{t}_1(p_i^c), \mathbf{t}_2(p_i^c))$.
- $\mathcal{F}_i$: surface features of patch i.
- $\hat{\mathcal{F}}_i$: surface features expressed in the local frame $\mathcal{L}_i$.
- $\mathcal{B}_i$: local band coordinates of patch i.
- $\hat{\mathcal{B}}_i$ local band coordinates expressed in the local frame $\mathcal{L}_i$.
- $\mathcal{P}$: patch or $\mathcal{P}_i$ if need to specify its index.

Network:
- $\mathcal{N}_\Theta : (q, \hat{\mathcal{B}}_i, \hat{\mathcal{F}}_i, u) \longmapsto \mathcal{N}_\Theta(q, \hat{\mathcal{B}}_i, \hat{\mathcal{F}}_i, u)$: global network.
- $\Phi_{\theta_1}, \Phi_{\theta_2}, \Phi_{\theta_3}$: MLPs.
- $\mathcal{N}_\Theta^{(\mathcal{P},u)} : q \longmapsto \mathcal{N}_\Theta(q, \hat{\mathcal{B}}, \hat{\mathcal{F}}, u)$: compressed network version where the patch and function are fixed. $(\hat{\mathcal{B}}, \hat{\mathcal{F}}) \in \mathcal{P}$.
- $\mathcal{N}_\Theta^{(\mathcal{P},u)}(Q) = \left\{\mathcal{N}_\Theta^{(\mathcal{P},u)}(q)\right\}_{q \in Q}$ where $Q \in R^{b \times 3}$. Vectorised version of $\mathcal{N}_\Theta$.

Training pipeline:
- cp: closest point operator.
- Π_i: image of $\mathcal{B}_i$ by cp.
- $\mathcal{M}$: set of monomials.
- g: to designate a given monomial.
- $\mathcal{D}$: dataset.
- $\mathcal{E}_i$: contains the functions (input/GT) for training.
- $L_{\mathrm{NC}}, L_{\mathrm{MSE}}$: losses.