
Improving GNN-Based Multiple Scattering Simulation with Error-Driven Adaptive Edge Sampling

Rémi Marsal
U2IS, ENSTA,
Institut Polytechnique de Paris
remi.marsal@ensta.fr

Stéphanie Chaillat
Laboratoire POEMS, CNRS, INRIA, ENSTA,
Institut Polytechnique de Paris
stephanie.chaillat@ensta.fr

Alexandre Chapoutot
U2IS, ENSTA,
Institut Polytechnique de Paris
alexandre.chapoutot@ensta.fr

Abstract

Recently, learning-based approaches allow very fast simulation of multiple scattering phenomena, i.e., the interactions between a field and several obstacles. They integrate a graph neural network (GNN) into the boundary element method framework (BEM), benefiting from the computational efficiency of the former and the reduction of the problem’s dimensionality from the latter. Thus, the GNN estimates the solution of the boundary integral equation (BIE) on the discretized obstacle surface before applying the boundary integral representation to recover the full solution of the multiple scattering problem. While the scattered field depends on the dense graph of interactions, only a subset can be modeled by the GNN to keep the method tractable. In this paper, we introduce a new GNN architecture for simulating multiple scattering phenomena that leverages an error-driven Adaptive Edge Sampling mechanism for selecting the interactions to model based on intermediate predictions of the expected error per node. Extensive experiments demonstrate that our approach improves the estimation of the BIE solution.

1 Introduction

Multiple scattering problems refer to physical phenomena in which a field (e.g., acoustic, electromagnetic, etc.) interacts with several obstacles. The boundary element method (BEM) is well-suited to address such problems. This numerical technique first solves the boundary integral equation (BIE), a reformulation of the governing PDE in which the unknowns belong to the obstacle surfaces. The BIE solution is then substituted into the boundary integral representation to recover the solution of the original PDE throughout the entire domain. The dimensionality reduction allowed by the BIE offers notable advantages of the BEM over the finite element method (FEM) in terms of both computational cost and accuracy. However, solving the BIE itself remains the main computational bottleneck of the BEM.

In order to accelerate multiple scattering simulation relative to traditional solvers such as the BEM, several learning-based approaches have been developed. First, these approaches rely on neural fields [Nair et al., 2025] or conditional neural fields [Hao et al., 2021] and are trained for a single 2D geometry. By design, the prediction domain is bounded even in the case of exterior problems in which the underlying domain is in fact infinite. The recently introduced PIBNet v1 and v2 approaches [Marsal and Chaillat, 2025b,a] address these limitations by integrating a GNN into the

BEM framework for solving the BIE. This reduces the computational cost of the BEM bottleneck while preserving its advantages: the dimension reduction and the ability to provide a solution in an infinite domain for exterior problems. Moreover, the GNN architecture, which takes as input the discretized surface of 3D obstacles, allows PIBNet v1 and v2 to handle a wide range of different geometries. The theory underlying the BEM states that the interactions between every pair of nodes must be considered for solving the BIE. Since dealing with the dense graph of interactions would be intractable for a GNN, PIBNet v1 [Marsal and Chaillat, 2025b] introduces an edge sampling strategy favoring the selection of short edges, which are likely to model stronger interactions between nodes. However, other factors beyond the distance influence the interaction between two nodes, such as the local context of each node, which may be impacted by the properties of the incident field or the presence of neighboring obstacles. Furthermore, edge length is a static criterion, so if several graphs of interactions are sampled and processed in a single forward pass, the importance of a given edge is always the same.

In this paper, we introduce an error-driven Adaptive Edge Sampling strategy for selecting more relevant interactions to model. To this end, we incorporate intermediate decoders into the PIBNet v2 architecture [Marsal and Chaillat, 2025a] that return predictions of the expected errors. Edges that model distant interactions are selected based on their length and on the predicted error at the source node. The interest of using error predictions is that it reflects nodes that are involved in strong and complex interactions due to their local context. In addition, new expected errors are predicted every time a graph of interaction is sampled during a forward pass, so the edge selection criterion is dynamically updated within a single forward pass. This differs from the work of Nguyen et al., which require two inference passes: the first one, performed on the raw graph, predicts errors that are used to construct an adapted graph, and the second one, performed on the adapted graph, yields the final predictions. Moreover, our error-driven Adaptive Edge Sampling creates edges between disconnected graphs. Finally, we have conducted extensive experiments to highlight the benefits of our approach on both in-distribution and out-of-distribution data for different types of problems.

2 Method

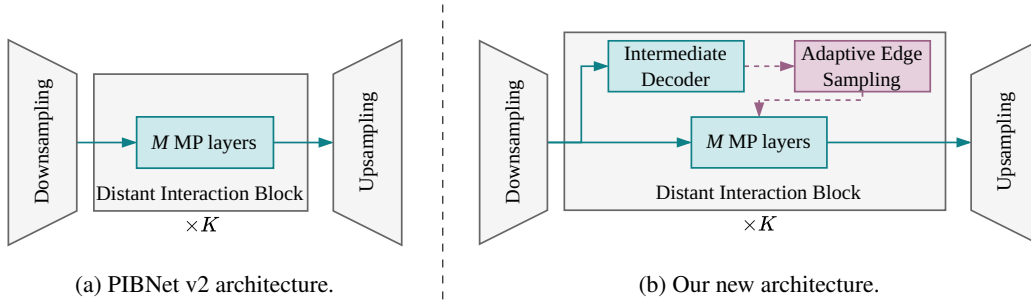


Figure 1: Illustration of PIBNet v2 architecture (a) and our new architecture (b).

2.1 Preliminaries on PIBNet v1 and v2 architecture

In this section, we describe the PIBNet architecture [Marsal and Chaillat, 2025b], including the improvements introduced by Marsal and Chaillat [2025a] illustrated in fig. 1 (left). It consists of a U-Net-like GNN that implements message-passing layers, following the Encoder-Processor-Decoder scheme of Pfaff et al. [2020]. Let the graph $\mathcal{G} = (V, E)$ represent the surface discretization of a set of disjoint obstacles with N nodes V characterized by 3D coordinates, and edges E that denote the local proximity of two nodes that belong to the same obstacle. First, PIBNet leverages dedicated encoders to initialize the features of the input nodes and edges, respectively. In the downsampling part of the processor, the resolution of the input meshes is reduced, providing a set of features associated with N_{low} low-resolution nodes V^{low} . Then, K Distant Interaction Block, each performing M message-passing layers, are applied to K graphs connecting the low-resolution nodes for modeling distant interactions. After upsampling the node features output by the last Distant Interaction Blocks, the final decoder returns the predictions $\hat{y} \in \mathbb{R}^{N \times d_f}$ of the BIE solution for each input node, where

d_f is 1 for Laplace problems and 2 for the real and imaginary parts of the solution of Helmholtz problems.

While the dense graph of interactions should be considered in theory, only a few connections can be modeled due to computational constraints. They are selected so that each low-resolution node serves as the destination node of a predefined number N_e of directed edges. For each interaction edge to be created, C candidate source nodes $n_c \in V^{\text{low}}$, $1 \leq c \leq C$ are randomly sampled; the one that is closest to the destination node $n_d \in V^{\text{low}}$ is retained. By favoring interactions between close nodes, this sampling mechanism is both consistent with physics and allows long-range interactions. However, the distance between the nodes is not the only criterion worth considering.

2.2 Error-Driven Adaptive edge sampling

Our new error-driven Adaptive Edge Sampling implements a learnable criterion for selecting interaction edges; see fig. 1, right. For this purpose, we introduce K intermediate decoders at the beginning of each Distant Interaction Block. Each decoder is composed of a two-layer MLP which takes as input the node features output by the previous message-passing layer. It predicts the BIE solution $\hat{\mathbf{y}}_k^{\text{low}} \in \mathbb{R}^{|V^{\text{low}}| \times d_f}$ and the associated expected error $\hat{\mathbf{e}}_k \in \mathbb{R}^{N_{\text{low}} \times d_f}$ for the low-resolution nodes V^{low} , $0 \leq k < K$. These intermediate predictions are supervised with the groundtruth solution restricted to low-resolution nodes $\mathbf{y}_{\text{low}}^* \in \mathbb{R}^{N_{\text{low}} \times d_f}$, and the ground truth error $\mathbf{e}_k^* = |\hat{\mathbf{y}}_k - \mathbf{y}_{\text{low}}^*| \in \mathbb{R}^{N_{\text{low}} \times d_f}$ of the k -th Distant Interaction Block, where $|\cdot|$ returns the element-wise absolute value. The final loss including the supervision of both intermediate low-resolution predictions and the final prediction on the original mesh is given by:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{huber}}(\hat{\mathbf{y}}, \mathbf{y}^*) + \sum_{k=0}^{K-1} \gamma^{K-k} \left(\mathcal{L}_{\text{huber}}(\hat{\mathbf{y}}_k^{\text{low}}, \mathbf{y}_{\text{low}}^*) + \mathcal{L}_{\text{huber}}(\hat{\mathbf{e}}_k, \mathbf{e}_k^*) \right) \quad (1)$$

where $\mathcal{L}_{\text{huber}}$ is the Huber loss $\mathcal{L}_{\text{huber}}$ [Huber, 1992], $\mathbf{y}^* \in \mathbb{R}^{N \times d_f}$ is the groundtruth solution, and $0 < \gamma < 1$ is a hyperparameter that reduces the weight of early intermediate predictions.

Thereafter, error predictions $\hat{\mathbf{e}}_k$ are used in the Adaptive Edge Sampling module of the k -th Distant Interaction Block. Similarly to PIBNet v1 and v2, for each of the N_e edges with destination node $n_d \in V^{\text{low}}$, C candidate directed edges corresponding to C candidate source nodes $n_c \in V^{\text{low}}$, $1 \leq c \leq C$, are randomly sampled. Instead of keeping the shortest candidate edge, each one is scored based on its length and on the expected error $\hat{e}_{k,n_c}$ of n_c according to the function f_{score}^α :

$$f_{\text{score}}^\alpha(p_{n_d}, p_{n_c}, \hat{e}_{k,n_c}) = \frac{\|p_{n_d} - p_{n_c}\|_2}{\left(\sum_{d=1}^{d_f} \hat{e}_{k,n_c,d}\right)^\alpha}, \quad (2)$$

where $\hat{e}_{k,n_c,d}$ is the d -th dimension of $\hat{e}_{k,n_c}$, p_{n_d} and p_{n_c} are the coordinates of the nodes n_d and n_c , respectively. $\alpha \in \mathbb{R}$ is a hyperparameter that weights the relative importance between the two criteria (expected error and edge length). Finally, only the candidate edge with the lowest score is kept and incorporated into the graph processed by the k -th Distant Interaction Block.

3 Experiments

We evaluate our approach with the benchmark introduced by Marsal and Chaillat [2025b]. It focuses on three multiple scattering problems that can be solved via the BEM: the Laplace Dirichlet problem and two Helmholtz problems, one governed by Dirichlet boundary conditions with a monopole source, the other by Neumann boundary conditions with an incident plane wave. Each problem comes with a training set containing three obstacles, along with test sets comprising three, six, and nine obstacles. We assess performance based on predictions of the solution of the BIE using the relative error Err_{rel} for the Laplace problem, and both the relative amplitude error Err_{ampl} and the absolute angle error $\text{Err}_{\text{angle}}$ for the Helmholtz problems (see Marsal and Chaillat [2025b] for further details). We provide comparison against standard GNN-based methods: MeshGraphNet [Pfaff et al., 2020], MuS-GNN [Lino et al., 2022], PIBNet v1 and v2 [Marsal and Chaillat, 2025b,a], transformer-based methods: PTv3[Wu et al., 2024b] and Transolver [Wu et al., 2024a], and Zhdanov et al. [2025]’s approach, Erwin, which includes both GNNs and transformers. For our method, we adopt the hyperparameters

of the PIBNet v2 architecture as well as its number of candidate nodes. The full implementation and hardware details of the different methods compared are provided in section A.

3.1 Results

Table 1: Benchmark results on three obstacle test datasets comparing the prediction errors.

Architecture Type		Method	Number of Parameters	Laplace	Helmholtz Dirichlet		Helmholtz Neumann	
GNN	Transformer			Err _{rel}	Err _{ampl}	Err _{angle}	Err _{ampl}	Err _{angle}
	✓	PTv3	19.2 M	0.081	0.148	0.141	0.070	0.073
	✓	Transolver	4.6 M	0.183	0.596	0.454	0.070	0.073
✓	✓	Erwin	7.9 M	0.119	0.218	0.175	0.074	0.076
✓		MeshGraphNet	5.3 M	0.173	0.286	0.181	0.068	0.068
✓		MuS-GNN	5.6 M	0.281	0.282	0.183	0.068	0.068
✓		PIBNet v1	5.2 M	0.044	0.107	0.105	0.049	0.050
✓		PIBNet v2	5.2 M	0.051	0.098	0.096	0.050	0.050
✓		Ours	5.6 M	<u>0.045</u>	0.087	0.086	0.044	0.043

Table 1 presents the in-distribution results, i.e., results on datasets with three obstacles per sample. We obtain that our method is on average 11% better than both PIBNet v1 and v2. The generalization capacities on problems with six and nine obstacles are studied in fig. 2 and section C. Our approach outperforms all the baselines on the three problems with an average improvement of 10% and 11% relative to PIBNet v1 and v2, respectively. In section B, we analyze the variability of our model at training and test time.

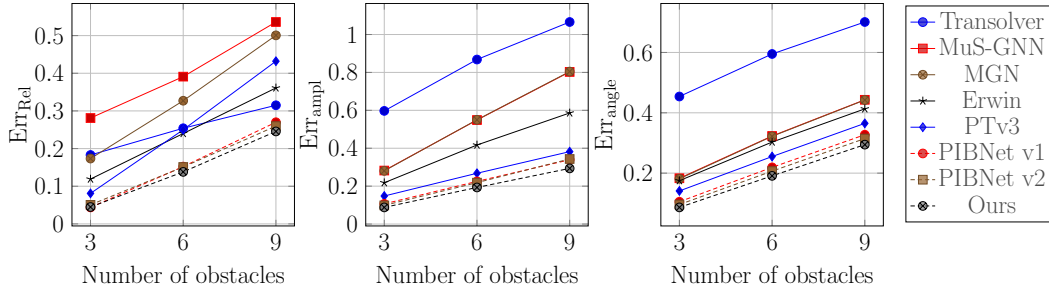


Figure 2: Estimation errors as a function of the number of obstacles for the Laplace (left) and the Helmholtz Dirichlet (middle and right) problems.

3.2 Ablation study

Table 2: Impact of the intermediate predictions and the Adaptive Edge Sampling on the performance.

Intermediate Predictions	Edge Sampling	Err _{ampl}	Err _{angle}
✗	PIBNet	0.098	0.096
✓	PIBNet	0.102	0.102
✓	Ours	0.087	0.086

In table 2, we study the impact of the two new components introduced in our approach, including the intermediate decoders and the Adaptive Edge Sampling. The first row is the baseline, which corresponds to PIBNet v2 [Marsal and Chaillat, 2025a]. In the second row, we show that implementing intermediate predictions alone slightly degrades performance by 7% on average. However, using the intermediate predictions for edge selection with our Adaptive Edge Sampling (last row) improves results by 11% relative to the baseline. These results demonstrate the benefit of our approach for learning-based simulation of multiple scattering. In sections D.1 and D.2, we provide a thorough study of the hyperparameters α in the edge score f_{score}^α , and γ in the loss function, respectively. The analysis of the number C of candidate nodes for training and testing in sections D.3 and D.4 confirms the optimal values found by PIBNet v1 and v2. In section E, we study the repartition of the error

predictions and show that the nodes that are the most likely to be involved in strong interactions are associated with higher expected errors. This justifies favoring the creation of edges from these nodes with our Adaptive Edge Sampling mechanism.

4 Conclusion

This paper presents a new GNN architecture for fast learning-based multiple scattering simulation. Based on the PIBNet architecture, it introduces an error-driven Adaptive Edge Sampling, in which edges that model distant interactions are selected based on their length and on predictions of the expected error per node performed by intermediate decoders during the forward pass. Extensive experiments in various scenarios highlight the superiority of our approach compared to existing methods. To confirm the benefits of our contribution in real-world cases, future work must address more challenging problems involving a greater number of obstacles and more complex shapes.

Acknowledgments

This work is supported by Agence de l’Innovation de Défense (AID) via Centre Interdisciplinaire d’Études pour la Défense et la Sécurité (CIEDS), through the APRO project.

References

- Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35: 16344–16359, 2022.
- Wenqu Hao, Yongpin P Chen, Pei-Yao Chen, Ming Jiang, Sheng Sun, and Jun Hu. Solving two-dimensional scattering from multiple dielectric cylinders by artificial neural network accelerated numerical green’s function. *IEEE Antennas and Wireless Propagation Letters*, 20(5):783–787, 2021.
- Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*, pages 492–518. Springer, 1992.
- Mario Lino, Stathi Fotiadis, Anil A Bharath, and Chris D Cantwell. Multi-scale rotation-equivariant graph neural networks for unsteady eulerian fluid dynamics. *Physics of Fluids*, 34(8), 2022.
- Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *International Conference on Learning Representations (ICLR)*, 2017.
- Rémi Marsal and Stéphanie Chaillat. Improving gnn-based multiple scattering simulation with rewiring. 2025a.
- Rémi Marsal and Stéphanie Chaillat. PIBNet: a physics-inspired boundary network for multiple scattering simulations, 2025b. URL <https://openreview.net/forum?id=MtYStkjrf0>.
- Siddharth Nair, Timothy F Walsh, Greg Pickrell, and Fabio Semperlotti. Multiple scattering simulation via physics-informed neural networks. *Engineering with Computers*, 41(1):31–50, 2025.
- Xuan Minh Vuong Nguyen, Nissrine Akkari, Fabien Casenave, Jonathan Viquerat, and Elie Hachem. Error-driven graph augmentation for mesh-based pde surrogates. In *Forty-third International Conference on Machine Learning*.
- Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter Battaglia. Learning mesh-based simulation with graph networks. In *International conference on learning representations*, 2020.
- Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for pdes on general geometries. *ICML*, 2024a.
- Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler faster stronger. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4840–4851, 2024b.
- Maksim Zhdanov, Max Welling, and Jan-Willem van de Meent. Erwin: A tree-based hierarchical transformer for large-scale physical systems. *ICML*, 2025.

A Implementation Details

A.1 Implementation details

Our approach uses PIBNet v2 [Marsal and Chaillat, 2025a] implementation details with $M = 2$, $K = 3$ and $C = 2$. However, in the Distant Interaction Blocks, instead of generating a number of edges per node that is proportional to the total number of nodes, we create a fixed number of $N_e = 20$ edges per node regardless of the size of the obstacle mesh. This allows the complexity of the method to grow linearly with the size of the input mesh rather than quadratically. For fair comparisons, both PIBNet v1 and v2 have been retrained with these settings. In the edge score function, α is set to 1, and γ in the loss function is set to 0.4, unless otherwise mentioned. The full implementation details of our method and the baselines used in the experiments are provided in table 3.

We train all models from scratch for 100 epochs, with AdamW optimizer [Loshchilov and Hutter, 2017], a batch size of 16, a learning rate starting at 10^{-4} and decreasing to 10^{-7} with a cosine scheduler, and gradient clipping by norm with a maximum value of 1.0. We apply the same random rotations to both the input mesh and the source as data augmentation.

We conduct all experiments on a single NVIDIA RTX 3090 GPU with 24GB memory. Training duration varied considerably across models: PTv3 completed in approximately 4 hours, aided by FlashAttention acceleration Dao et al. [2022], while Transolver required over 18 hours due to its need for gradient accumulation when batch sizes exceed 1. For the remaining models, training took approximately 8 hours for our approach, slightly less for PIBNet v1 and v2, 11 hours for MuS-GNN, 12 hours for Erwin, and 18 hours for MeshGraphNet.

B Stability Analysis

In tables 4 and 5, we provide the relative standard deviation of our approach at training and test time, respectively. The results at training time highlight low variability for the Laplace and the Helmholtz Neumann problems. The higher relative standard deviation for the Dirichlet Helmholtz problem is due to a single outlier seed; excluding it, the average relative standard deviation for the different metrics and number of obstacles drops to 1.19%. Regarding variability at inference, the results show a relative standard deviation of at most 0.31% regardless of the problem or the number of obstacles, which demonstrates the stability of the trained models.

C Additional Results

Figure 3 shows the generalization performance with six and nine obstacles for the Helmholtz Neumann problems.

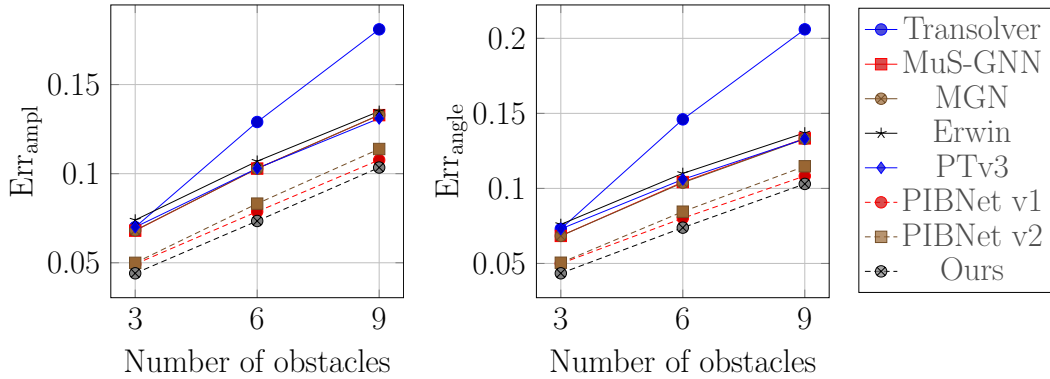


Figure 3: Estimation errors as a function of the number of obstacles for the Helmholtz Neumann problems.

Table 3: Implementation details of the different models compared in this paper.

Model	Parameter	Value
Ours	Number of levels	3
	Latent dimension at first level	64
	Expansion rate of latent dimension	2
	Number K of Distant Interaction Blocks	3
	Number M of message-passing layer per Distant Interaction Block	2
	Number N_e of interaction edges per low-resolution node	20
	Number C of candidate edges during training	2
	Number C of candidate edges during test	#obstacles/3 + 1
	α	1
PIBNet v1 [Marsal and Chaillat, 2025b]	γ	0.4
	Number of levels	3
	Expansion rate of latent dimension	2
	Latent dimension at first level	64
	Number of Distant Nodes Graph Processor Blocks	6
	Number of edges per low-resolution node	20
	Number of candidate edges during training	2
	Number C of candidate edges during test	#obstacles/3 + 1
PIBNet v2 [Marsal and Chaillat, 2025a]	Number of levels	3
	Latent dimension at first level	64
	Expansion rate of latent dimension	2
	Number of Distant Nodes Graph Processor Blocks	6
	Number of Distant Node Graphs	3
	Number of edges per low-resolution node	20
	Number of candidate edges during training	2
	Number C of candidate edges during test	#obstacles/3 + 1
MeshGraphNet [Pfaff et al., 2020]	Processor depth	15
	Latent dimension	128
MuS-GNN [Lino et al., 2022]	MPNN dim.	128
	Scale number	3
	Number of MPs at each level	(4, 2, 4)
Point Transformer V3 [Wu et al., 2024b]	Grid Size	0.2
	Encoder Latent dimensions	(64, 128, 256)
	Encoder depths	(2, 2, 6)
	Encoder heads	(4, 8, 16)
	Encoder patch size	2048
	Decoder Latent dimensions	(64, 128)
	Decoder depths	(2, 2)
	Decoder heads	(4, 8)
	Decoder patch size	2048
	Stride	2
Transolver [Wu et al., 2024a]	Latent dimension	256
	Number of layers	8
	MPL ratio	4
Erwin [Zhdanov et al., 2025]	MPNN dim.	64
	Latent dimensions	(64, 128, 256)
	Window sizes	(512, 512, 512)
	Encoder depths	(2, 2, 6)
	Encoder heads	(4, 8, 16)
	Decoder depths	(2, 2)
	Decoder heads	(4, 8)
	Stride	2
	Distance-based attention bias	Enabled

D Additional Ablations

D.1 Impact of the α Parameter in the Score Function f_{score}^α

In fig. 4, we examine the impact of the α parameter in the score function f_{score}^α , which controls the relative importance given to the error predictions compared to the edge length when sampling distant edges. For the Helmholtz Dirichlet problem, the results show that a negative α , i.e., favoring edges from nodes with low expected error, degrades performance. This behavior is consistent with the fact that such edges typically correspond to weak interactions (see section E), meaning they carry little useful information to the target node. At test time, when the number of obstacles matches that of the training set, the best results are obtained with $\alpha = \infty$, in which case f_{score}^α only accounts for the error predictions and the edge length is ignored. In contrast, when the number of obstacles is doubled or tripled relative to training conditions, smaller values of α yield better performance. We ultimately

Table 4: Relative standard deviation at training time. Each model was trained with five random seeds and evaluated with a fixed seed.

Number of obstacles	Laplace Err _{rel}	Helmholtz Err _{ampl}	Dirichlet Err _{angle}	Helmholtz Err _{ampl}	Neumann Err _{angle}
3	0.82%	7.37%	8.21%	0.36%	0.44%
6	1.15%	4.96%	5.35%	0.22%	0.42%
9	1.25%	3.63%	4.06%	0.18%	0.23%

Table 5: Relative standard deviation at training time. A single model is evaluated with five random seeds.

Number of obstacles	Laplace Err _{rel}	Helmholtz Err _{ampl}	Dirichlet Err _{angle}	Helmholtz Err _{ampl}	Neumann Err _{angle}
3	0.31%	0.18%	0.21%	0.10%	0.09%
6	0.24%	0.15%	0.15%	0.13%	0.09%
9	0.03%	0.16%	0.07%	0.09%	0.08%

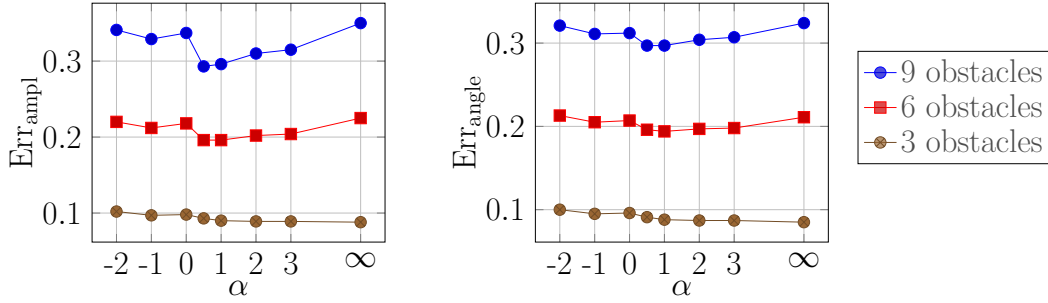


Figure 4: Estimation errors as a function of the α parameter in the score function f_{score}^α . When $\alpha = \infty$, the score function only depends on error predictions and the edge length is ignored.

set $\alpha = 1.0$ for our main experiments, as this value offers a reasonable compromise between these two regimes.

D.2 Impact of the weight γ in the loss $\mathcal{L}_{\text{total}}$

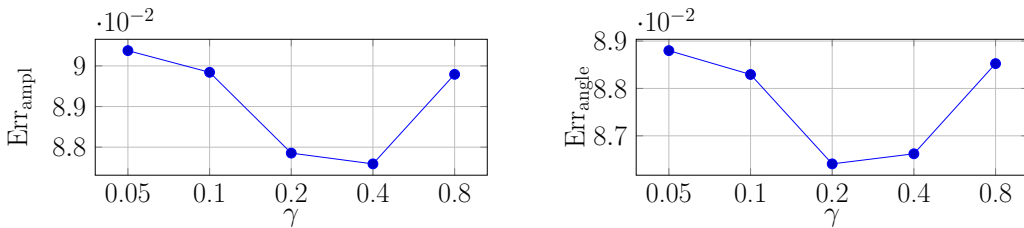


Figure 5: Errors for the Helmholtz Dirichlet problem with three obstacles as a function of the γ parameter in the loss function $\mathcal{L}_{\text{total}}$.

Figure 5 presents the effect of the γ parameter in the loss function $\mathcal{L}_{\text{total}}$, which controls the weighting of intermediate predictions. For the Helmholtz Dirichlet problem with three obstacles, results indicate that $\gamma = 0.2$ and $\gamma = 0.4$ achieve the best performance when evaluating the model on in-distribution obstacle shapes. We adopt $\gamma = 0.4$ throughout the main experiments of this work.

D.3 Impact of the Number C of Candidate Edges during Training

In fig. 6, we examine the effect of varying the number C of candidate edges from which each edge of a Distant Interaction Graph $\mathcal{G}_k^{L-1}$, $1 \leq k \leq K$, is selected during training. We obtain optimal results with $C = 2$.

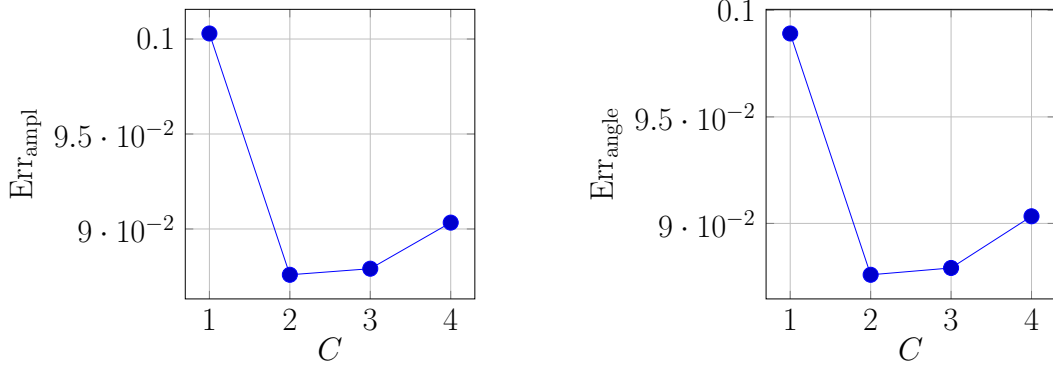


Figure 6: Estimation errors with three obstacles as a function of the number C of candidate edges from which each edge in a Distant Interaction Graph $\mathcal{G}_k^{L-1}$, $1 \leq k \leq K$, is selected. More specifically, C varies during both training and testing.

D.4 Scaling the Number C of Candidate Nodes with the Number of Obstacles at Test Time

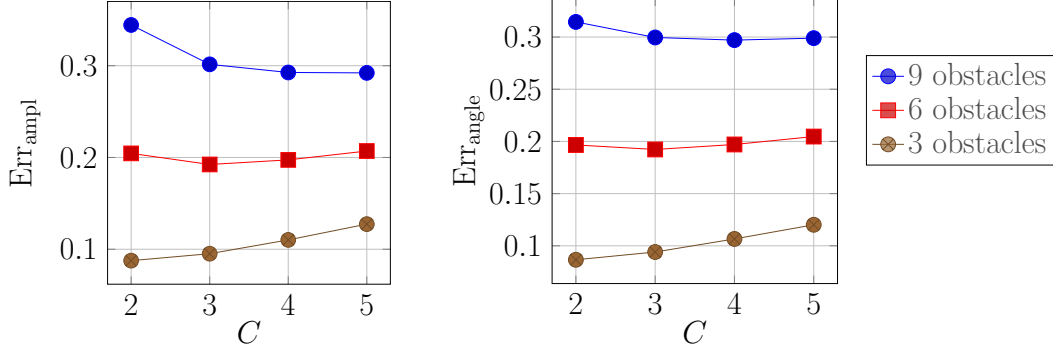


Figure 7: Estimation errors as a function of the number C of candidate edges for selecting each edge in the Distant Interaction Graph $\mathcal{G}_k^{L-1}$, $1 \leq k \leq K$. Here, C varies at test time while the evaluated model has been trained with $C = 2$.

In the BEM theory, the number of interactions grows quadratically with the number of nodes, but the number of edges per node in the Distant Interaction Graph $\mathcal{G}_k^{L-1}$, $1 \leq k \leq K$, is fixed, so edge selection in our adaptive edge sampling must become more stringent. To ensure that the most relevant edges are retained, we increase the number of candidate edges C as the number of obstacles increases. Figure 7 illustrates that when our method is evaluated with more obstacles than seen in the training set, performance can be optimized by increasing the number C of candidate edges at test time while the model has been trained with $C = 2$. This observation motivates a simple heuristic for scaling C with the number of obstacles: when the obstacle count is multiplied by a factor x , C should be increased by $x - 1$.

E Analysis of the Repartition of the Prediction Errors

Table 6: Correlations between the expected error predicted by each Distant Interaction Block and the inverse of the two distances, including the source distance or the distance to the closest obstacle.

Index of the Distant Interaction Block	1	2	3
Inverse distance to the source	0.65	0.46	0.50
Inverse distance to the closest obstacle	0.21	0.51	0.47

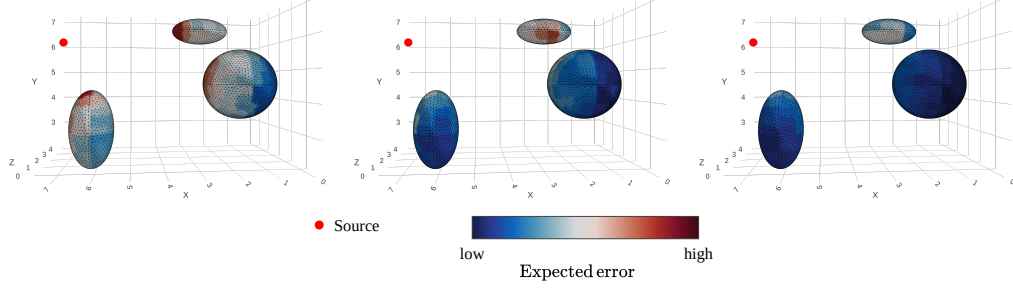


Figure 8: Expected error predicted by the intermediate decoders for a sample from the Helmholtz Dirichlet test set. The intermediate decoder predictions on the low-resolution graph have been interpolated to the original mesh.

To better understand why error-driven edge selection improves performance, we verify that error predictions highlight the nodes that are involved in strong interactions. To this end, we conducted a statistical analysis of the error predictions of our models on the Helmholtz Dirichlet problems with three obstacles. In particular, we study in table 6 the correlations between the predicted error in each Distant Interaction Block and two quantities indicating whether a node is likely to interact strongly with others: the inverse distance to the monopole source and the inverse distance to the closest node from another obstacle. The results reveal strong correlations between the predicted errors and both quantities. The only exception is the inverse distance to the closest obstacle in the first Distant Interaction Block because, at this early stage of the GNN, no interaction between distinct obstacles has yet been explored. The illustration of this phenomenon in fig. 8 confirms that the nodes with high expected error are likely to be involved in strong interactions. While it suggests that a custom edge selection criterion could be introduced based on the inverse distance to the monopole source and the inverse distance to the closest node from another obstacle instead of the error prediction, such a criterion is only valid in the case of a monopole source. Instead, error prediction provides a generic edge selection criterion that automatically adapts to any problem.