EIGENRR: LEARNING TETRAHEDRAL SEARCH SUBSPACES FOR IMPACT SOUND SYNTHESIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Synthesizing a realistic sound for a struck object first requires computing how that object vibrates. Physically, the object is meshed into many small tetrahedra (a finite-element discretization), and a very large eigenvalue problem is then solved. Classical solvers such as ARPACK and LOBPCG are accurate but too slow. Learning-based approaches either regress modal quantities directly or learn a search subspace on a voxelized mesh that requires interpolation to tetrahedral FEM meshes, risking transfer error. EigenRR learns an overcomplete search subspace on the tetrahedral FEM mesh, and Rayleigh–Ritz extracts approximate eigenpairs using that mesh's assembled stiffness and mass matrices. The resulting modes are consistent with the physical operator, so they also warm-start an iterative solver. Twenty LOBPCG steps from this prediction reach a mel-frequency error of 2.4×10^{-4}, $1.55\times$ and $63\times$ more accurate than NeuralSound and random initializations under the same budget. One forward pass already beats 20 iterations from a random start. We build VibraVerse++, a geometry–acoustics dataset of tetrahedral meshes. We train and evaluate on it, and release the meshes together with the assembled FEM matrices and reference eigenspectra. The method returns accurate frequencies and mode shapes at interactive speed.

1 INTRODUCTION

Tapping a cup produces a ring; tapping wood produces a knock. An object's shape and material determine how it vibrates, and how it vibrates determines the sound it makes. Physically based sound synthesis computes this causal chain Van Den Doel et al. (2001); Chadwick et al. (2009); James et al. (2006): the object is discretized into a tetrahedral mesh, yielding a stiffness matrix K that describes the elastic coupling between vertices and a mass matrix M that describes the mass distribution. Solving the generalized eigenvalue problem $KU = MU\Lambda$ then gives the object's natural frequencies and mode shapes Reddy (2019).

The bottleneck is this eigenvalue problem. Its size is $3N \times 3N$, where N is the number of vertices and is often thousands to tens of thousands. Classical solvers such as ARPACK Lehoucq et al. (1998) and LOBPCG Knyazev (2001) are accurate, but each object takes seconds to minutes. That cost is unacceptable for interactive sound design, or for computing modes over a dataset of tens of thousands of objects.

Amortizing the cost with a neural network is a natural idea, but existing attempts fall short. One approach regresses frequencies directly from shape Pang et al. (2026). It is fast, but the map from geometry to eigenvalues is highly distorted, so regression accuracy is poor. NeuralSound Jin et al. (2022) takes another route: it predicts mode shapes on a coarse voxel grid and refines them with an iterative solver. Voxelization itself introduces a systematic scale bias, and mapping the modes from voxels back onto the true mesh costs additional error and time.

These methods ask the network to guess the eigenpairs. For a symmetric generalized eigenproblem, a good subspace is enough: Rayleigh–Ritz extracts the best approximation inside it, and LOBPCG converges much faster from a good start.

We therefore propose EigenRR, which **learns a subspace rather than the eigenpairs**. On the tetrahedral mesh, a graph network predicts $k+m = 80$ candidate vibration directions. A differentiable Rayleigh–Ritz layer mass-orthogonalizes them, projects K into at most 80 dimensions, and

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.

returns Ritz approximations of the first 40 elastic modes. When the basis is full rank and the orthogonalization is accurate, the minimax principle makes those Ritz values upper bounds on the true eigenvalues. Near-degenerate truncation means we treat this as a variational interpretation under those conditions, not as an unconditional certificate. At test time the Ritz vectors can warm-start LOBPCG. Training still uses the reference spectrum; the residual, which needs no labels, is left for later fine-tuning.

Following the data generation pipeline of VibraVerse Pang et al. (2026), we construct VibraVerse++, a collection of about 16,000 high-quality tetrahedral meshes. On these meshes, one forward pass takes about 39 ms and reaches a mel-frequency error of 7.2×10^{-3}, about $2.1\times$ and $12.3\times$ below NeuralSound and direct frequency regression. At most 20 warm-started LOBPCG iterations, 0.19 s in total, reduce the error to 2.4×10^{-4}, $1.55\times$ and $63\times$ below NeuralSound and random initializations. The same budget from a random start remains worse than our direct prediction (Section 5).

Our contributions are:

- **A neural Rayleigh–Ritz paradigm for modal analysis.** The network predicts an overcomplete search subspace on the tetrahedral mesh, and a differentiable Rayleigh–Ritz projection recovers the eigenpairs. The projected spectrum drops the lowest 6 Ritz pairs, which correspond to rigid motion.
- **A learned warm start for iterative eigensolvers.** Predicted Ritz vectors plus 6 analytic rigid-body directions initialize LOBPCG on the exact FEM operator.
- **Data and an evaluation protocol for learning-based modal analysis.** We release tetrahedral meshes, assembled stiffness and mass matrices, and ARPACK reference spectra derived from VibraVerse++, and report subspace, eigenvalue, and frequency error.

2 Related Work

2.1 Modal Sound Synthesis and Acoustic Transfer

Impact-sound synthesis separates structural vibration, acoustic radiation, and runtime excitation. Modal analysis turns elastodynamics into damped oscillators, so collisions and friction can be rendered efficiently Van Den Doel et al. (2001); Barbic (2012), but every new object still requires an eigendecomposition of matrices assembled from geometry and material. Given the modes, radiation can be precomputed with BEM Kirkup (2019), PAT James et al. (2006), FFAT maps, or Klein-PAT Chadwick et al. (2009); Wang & James (2019). Those methods accelerate radiation from a known vibration. This paper computes the frequencies and mode shapes they take as input.

2.2 Learning-Based Modal Analysis

Earlier learning-based approaches to modal sound synthesis predicted modal quantities directly. Deep-Modal Jin et al. (2020) predicts frequencies and amplitudes from object shape and impact conditions, while Lin et al. (2026) predict natural frequencies and nodal mode shapes from finite-element mesh graphs. NeuralSound Jin et al. (2022) instead learns a voxel-based search subspace for eigensolving. We share its subspace-based perspective, but learn the search space on the tetrahedral discretization of the target eigenproblem and train it against reference modes and perceptually weighted frequencies.

DiffSound Jin et al. (2024) makes modal analysis and rendering differentiable in order to invert geometry and material from sound. VibraVerse Pang et al. (2026) pairs geometry, material, modes, and sound, and includes direct frequency regression. Regression does not return mode shapes of the finite-element operator. Rayleigh–Ritz on that operator produces both.

2.3 Iterative Eigensolvers and Warm Starts

Only a few eigenpairs of a large sparse system are typically needed. ARPACK uses an implicitly restarted Krylov method Lehoucq et al. (1998), while LOBPCG uses block updates and Rayleigh–Ritz extraction Knyazev (2001); high accuracy can still require many sparse matrix operations.

Neural approaches represent eigenfunctions and optimize for eigenpairs, either without eigenpair labels Ben-Shaul et al. (2023) or through Rayleigh-quotient objectives and orthogonalization Rowan et al. (2025). Our network instead learns a search subspace across objects, leaving eigenpair extraction to each object's FEM operator. As NeuralSound demonstrated Jin et al. (2022), a learned subspace can also warm-start LOBPCG. Learned directly on the tetrahedral degrees of freedom, the same subspace provides both immediate Ritz estimates and a starting point for further refinement under the target FEM operator.

2.4 Geometry–Acoustics Datasets

ShapeNet Chang et al. (2015), Objaverse Deitke et al. (2023), and ObjectFolder Gao et al. (2021) provide geometry or multisensory observations, not operators and reference spectra that verify the finite-element eigen-equation. VibraVerse Pang et al. (2026) links geometry, material, modes, and sound.

3 Background

3.1 FEM Modal Analysis

FEM partitions the object into tetrahedra whose vertices move in three directions. Vibration is then described by a stiffness matrix K, set by the geometry and by Young's modulus E and Poisson's ratio ν, and a mass matrix M, set by the density ρ. The equation of motion is

$$M\ddot{x} + C\dot{x} + Kx = f(t), \tag{1}$$

where $x \in \mathbb{R}^{3N}$ stacks all vertex displacements and $C = \alpha M + \beta K$ is Rayleigh damping Reddy (2019); Chowdhury & Dasgupta (2003).

Simple harmonic motion, $x = ue^{i\omega t}$, turns Equation (1) into the generalized eigenvalue problem

$$KU = MU\Lambda. \tag{2}$$

Each pair (λ_i, u_i) is one mode: $\lambda_i = \omega_i^2$ sets the pitch $f_i = \sqrt{\lambda_i}/(2\pi)$, and u_i is the shape. The first 6 eigenvalues are zero and span rigid translation and rotation. Audible elastic modes start at order 7. We target the first $K{=}40$ of them.

3.2 From Modes to Sound

A strike is a sum of decaying sinusoids Barbic (2012),

$$S(t) = \sum_i A_i \, e^{-d_i t} \sin(2\pi f_i t), \tag{3}$$

with decay $d_i = \frac{1}{2}(\alpha + \beta\omega_i^2)$ and amplitude A_i set by the strike location and the mode shape. Eigenvalues place the partials; eigenvectors set their relative loudness. Frequency error is the primary metric (Section 5.1). Eigenvalues alone cannot tell strike points apart (Section 5.7).

3.3 Subspace Eigensolvers: Rayleigh–Ritz and LOBPCG

For Equation (2), classical methods **guess a subspace, then solve exactly inside it**. Given a trial basis $V \in \mathbb{R}^{3N\times s}$ with $s \ll 3N$ (for example $s{=}80$ against $3N{\approx}10^4$), Rayleigh–Ritz mass-orthonormalizes V to a basis U with $U^\top MU = I$, compresses the operator,

$$\hat{A} = U^\top KU, \qquad \hat{B} = U^\top MU = I, \tag{4}$$

and solves the $s \times s$ problem $\hat{A}y = \hat{\lambda}y$. The Ritz pair is $(\hat{\lambda}_i, Uy_i)$. It is optimal inside the subspace. By the Courant–Fischer principle it is also an upper bound, $\hat{\lambda}_i \geq \lambda_i$. The open question is how to choose the subspace.

LOBPCG Knyazev (2001) repeats Rayleigh–Ritz and a preconditioned correction from an initial block. A closer start needs fewer iterations. That is the reason to warm-start it from a predicted subspace (Section 4.5).

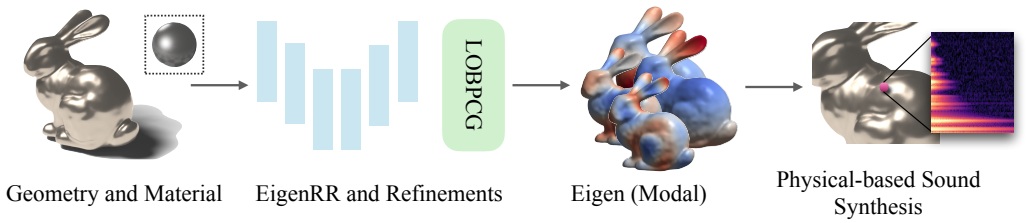

Figure 1: Schematic overview of our method. A graph neural network predicts an 80-dimensional search basis V from the tetrahedral mesh. A differentiable Rayleigh–Ritz layer extracts the first 40 elastic modes, which can further warm-start LOBPCG for higher accuracy.

4 METHOD

4.1 OVERVIEW

Given a tetrahedral mesh and $K, M \in \mathbb{R}^{3N\times 3N}$, we compute the first k=40 elastic modes of Equation (2). The network does not regress those modes. It predicts $k+m = 80$ candidate directions (m=40 of them are surplus) that span a search subspace. A differentiable Rayleigh–Ritz layer (Section 4.3) solves the projected eigenproblem in that span. For higher accuracy, the Ritz vectors warm-start LOBPCG on the exact operator (Section 4.5, Figure 1).

4.2 NEURAL SUBSPACE PREDICTION

Mesh vertices are graph nodes and mesh edges are graph edges. Each node receives its normalized coordinate $p \in [-1, 1]^3$, lifted by a sinusoidal positional encoding $\gamma(p)$ Mildenhall et al. (2021) so that high-frequency vibration detail is not left implicit in a smooth coordinate. A graph U-Net Ronneberger et al. (2015) pools and unpools across scales, emits a 256-dimensional feature per node, and applies two rounds of neighbor averaging. A residual MLP then maps each feature to 80×3 displacements. Stacking every vertex produces the search basis $V \in \mathbb{R}^{3N\times 80}$. The backbone can be swapped; octree variants Wang et al. (2017; 2022); Wu et al. (2024) are implemented but not compared here.

4.3 DIFFERENTIABLE RAYLEIGH–RITZ LAYER

This layer turns the 80 predicted directions into eigenpair estimates, in four steps. Matrix multiplications, the symmetric eigendecomposition, and the normalization on the training path are all differentiated by PyTorch, so gradients flow back to the network.

(1) Mass orthogonalization. The 80 directions may be linearly dependent, and projecting them directly is numerically unstable. The implementation follows an SVQB-style procedure Jin et al. (2022). It first forms the Gram matrix $G = V^\top MV$, drops original columns whose diagonal mass energy does not exceed 10^{-8}, and then eigendecomposes the diagonally scaled and symmetrized Gram matrix. Gram eigenvalues below 10^{-8} are clamped to that floor before the inverse square root is computed; the corresponding directions are not dropped. When the search basis is well conditioned, the resulting U therefore satisfies $U^\top MU \approx I$. Near degeneracy, the relation holds only approximately.

(2) Projection. The stiffness matrix K is compressed into the orthogonalized search subspace:

$$\hat{A} = \tfrac{1}{2}\left(U^\top KU + (U^\top KU)^\top\right). \tag{5}$$

If no original column was dropped for zero mass energy, $\hat{A}$ is 80×80; otherwise its dimension shrinks accordingly.

(3) Small-matrix solve. A symmetric dense eigendecomposition solves $\hat{A}Y = Y\hat{\Lambda}$, and the solution is mapped back by $X = UY$. Each column is then re-normalized so that $x_i^\top Mx_i = 1$. The pairs $(\hat{\lambda}_i, x_i)$ are the Ritz approximations in the network's search subspace. This step takes about 3.8 ms on average in our evaluation.

(4) Skipping the lowest 6 Ritz pairs. For a connected, unconstrained 3D elastic body, the lowest 6 orders of the exact spectrum are 3 translations and 3 rotations. The current Rayleigh–Ritz implementation does not construct or match rigid-body modes analytically at this stage. It sorts the projected eigenvalues in ascending order, discards the lowest 6 Ritz pairs, and takes the next 40 as the elastic-mode estimates. Analytic rigid-body directions are constructed explicitly only in the later LOBPCG warm start.

4.4 Training Objective

Subspace alignment loss. Let $U_{\mathrm{gt}} \in \mathbb{R}^{3N\times 40}$ be the ground-truth elastic modes. Coverage of that subspace is

$$\mathcal{L}_{\mathrm{sub}} = 1 - \frac{1}{k}\left\|X^{\top} M\, U_{\mathrm{gt}}\right\|_F^2 . \tag{6}$$

Each column of $X^{\top}MU_{\mathrm{gt}}$ is the mass-inner-product projection of one ground-truth mode onto the prediction. The loss is invariant to the basis chosen inside the subspace.

Mel-frequency loss. Frequencies are $f = \sqrt{\lambda}/(2\pi)$. Perception is roughly logarithmic, so we map to mels, $m(f) = 2595\log_{10}(1 + f/700)$, then rescale using the mel values at 20 Hz and 20 kHz and an extra factor of $1/2$:

$$\widetilde{m}(f) = \frac{1}{2}\left(2\,\frac{m(f) - m(20)}{m(20000) - m(20)} - 1\right). \tag{7}$$

The loss is

$$\mathcal{L}_{\mathrm{mel}} = \mathrm{MSE}\big(\widetilde{m}(\hat{f}),\, \widetilde{m}(f_{\mathrm{gt}})\big). \tag{8}$$

This is the mel-frequency MSE used in training and in the tables. The map does not clip frequencies outside 20 Hz–20 kHz, and eigenvalues inside the square root are floored at 1 during training.

Basis regularization. Without a scale constraint the directions collapse or drift, and orthogonalization becomes unstable. A light penalty pulls each direction's mass energy toward 1:

$$\mathcal{L}_{\mathrm{reg}} = \mathrm{mean}\Big(\big(\log \mathrm{diag}(V^{\top}MV)\big)^2\Big). \tag{9}$$

The total loss is

$$\mathcal{L} = 1.0\,\mathcal{L}_{\mathrm{sub}} + 7.0\,\mathcal{L}_{\mathrm{mel}} + 0.1\,\mathcal{L}_{\mathrm{reg}}. \tag{10}$$

Eigenvalue MSE and the Rayleigh residual are logged during training as monitoring metrics, but they do not enter the gradient update. The three weights are compared in Section 5.4.

4.5 Warm-Started LOBPCG Refinement

At test time the Rayleigh–Ritz output can be used directly: about 39 ms, of which the network is about 35 ms and Rayleigh–Ritz about 3.8 ms. For higher accuracy we build 3 translations and 3 rotations about the vertex centroid, mass-orthogonalize them, and concatenate them with the 40 Ritz vectors. LOBPCG then runs on the exact K and M in float64, with orthogonalization and no extra preconditioner. "LOBPCG20" means at most 20 iterations, with early stopping on the residual. Reported times include the network, Rayleigh–Ritz, and the solver iterations. They exclude data loading, matrix assembly, and part of the CPU–GPU transfer.

5 Experiments

5.1 Experimental Setup

The benchmark is built from our VibraVerse++ geometry–acoustics dataset. Following the generation method described in VibraVerse Pang et al. (2026), we extract fine-grained textual descriptions from rendered images of uncurated meshes and pass these prompts through a cascaded text-to-image-to-3D pipeline, synthesizing watertight 3D models with diverse semantics. After generating these surface meshes, each object is tetrahedralized with fTetWild Hu et al. (2020), assigned material

Table 1: Modal-analysis accuracy and per-sample runtime on the test set (n=1595, mean over samples). LOBPCG refinement uses 20 iterations. The best result in each column is in bold.

Method	Subspace error ↓	Mel-frequency MSE ↓	Time (s) ↓
VibraVerse (direct regression)	—	0.088113	0.148
NeuralSound Jin et al. (2022)	0.1314	0.014753	0.054
EigenRR (forward only)	0.0995	0.007170	**0.039**
Random init. + LOBPCG20	0.3607	0.014981	**0.150**
NeuralSound + LOBPCG20	0.0788	0.000366	0.251
EigenRR + LOBPCG20	**0.0630**	**0.000236**	0.192

parameters (ρ, E, ν) from a library of 10 materials (steel, copper, aluminum, plastic, polycarbonate, wood, glass, ceramic, concrete, and stone), and assembled into K and M with linear FEM elements. Ground truth is computed with ARPACK Lehoucq et al. (1998): we request 70 eigenpairs, discard the 6 rigid-body modes, store the remaining 64 elastic modes, and use the first 40 for training and evaluation. VibraVerse++ contains 15,941 tetrahedral meshes, split into training and test sets in the ratio 9:1 (14,346 / 1,595). No validation set is used; the test set is held out from training.

Metrics. We report four metrics. (i) **Subspace error** $\mathcal{L}_{\text{sub}}$ (Equation (6)) measures how well the predicted subspace covers the ground-truth subspace. (ii) **Eigenvalue error** is the maximum normalized MSE of the Ritz values. (iii) **Mel-frequency MSE** is the primary metric, aligned with human hearing (Equation (8)). (iv) **Per-sample runtime** is split into the network forward pass, the Rayleigh–Ritz solve, and LOBPCG refinement.

Baselines. There are three baselines. (i) *VibraVerse* Pang et al. (2026) regresses frequencies directly from shape and material features, and represents the pure-regression route. (ii) *NeuralSound* Jin et al. (2022) predicts mode shapes on a voxel grid, and represents the voxel-plus-refinement route. Its output must be interpolated back onto our tetrahedral mesh before a fair comparison. (iii) *Randomly initialized LOBPCG* starts from a random block on the exact operator, and quantifies the value of the warm start itself. All learned methods are trained and evaluated on the same data split.

Training details. We train for 400 epochs with AdamW (learning rate 5×10^{-5}, linearly decayed to 0 after a 20-epoch warmup), batch size 16, and gradient clipping at 0.5, on 2× RTX 4090 GPUs for 45.5 hours.

5.2 Main Results

A forward pass with no iteration already reaches mel error 0.0072 in 39 ms, half of NeuralSound and $12.3\times$ below direct regression (Table 1). The same 20 LOBPCG steps reach 2.4×10^{-4} from our prediction, 3.7×10^{-4} from NeuralSound, and 1.5×10^{-2} from a random block, so 20 random steps remain worse than one direct prediction. EigenRR+LOBPCG20 also finishes in 0.19 s, against 0.25 s for NeuralSound+LOBPCG20. The gap persists from 0 to 100 iterations (Section 5.3).

5.3 Accuracy–Latency Trade-off

Table 1 shows only the 0-step and 20-step points. We do another experiment that sweeps the LOBPCG caps over $0, 20, 50, 100$ on the same test set. Results are shown in Figure 2. Due to nondeterminism in GPU floating-point arithmetic and in LOBPCG algorithm, and to differences in machine load, the accuracy and timings of this experiment differ from those in Table 1.

EigenRR with no iteration (about 0.068 s, mel error 6.2×10^{-3}) already beats 20 random steps (0.215 s, 1.5×10^{-2}). At 20, 50, and 100 steps its errors are 2.5×10^{-4}, 8.1×10^{-5}, and 2.8×10^{-5}, below NeuralSound at the same caps (3.7×10^{-4}, 1.3×10^{-4}, 4.4×10^{-5}). Random initialization at 100 steps (about 1.05 s, 7.5×10^{-4}) is still worse than EigenRR at 20 steps.

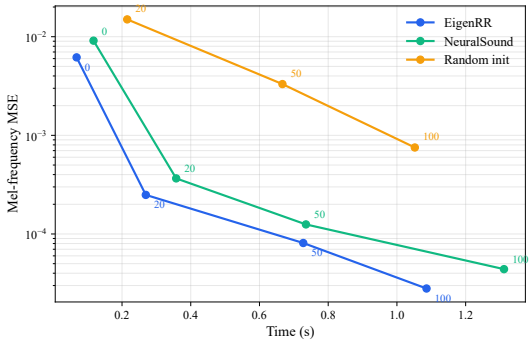

Figure 2: Accuracy–latency trade-off. EigenRR achieves lower mel-frequency MSE in less time than NeuralSound and random initialization. Labels denote LOBPCG step caps; random initialization has no 0-step result.

Table 2: Skipping Rayleigh–Ritz and regressing 40 mode shapes directly. Blank cells in the forward pass are there because this path does not output Ritz values. The best is highlighted in bold.

	Forward only			LOBPCG20		
Training path	Sub. ↓	Eig. ↓	Mel ↓	Sub. ↓	Eig. ↓	Mel ↓
Direct regression of 40 modes	0.1721	—	—	0.0892	0.0236	0.000483
EigenRR	**0.0995**	0.4683	0.007170	**0.0630**	**0.0099**	**0.000236**

5.4 Ablations

Both ablations use the test set of Table 1 (n=1595).

Skipping Rayleigh–Ritz. The control changes the head to 40×3, skips Rayleigh–Ritz in training, and supervises the 40 raw outputs with the subspace loss. That path has no Ritz values before iteration, so the forward columns of Table 2 report only subspace error.

Direct regression leaves subspace error 0.1721, against 0.0995 after Rayleigh–Ritz. After 20 LOBPCG steps its mel and eigenvalue errors are 4.8×10^{-4} and 0.0236, against 2.4×10^{-4} and 0.0099 for the full model. The projected subspace is a tighter cover and a better warm start.

Loss combinations. Weights are $(\mathcal{L}_{\text{sub}}, \mathcal{L}_{\text{mel}}, \mathcal{L}_{\text{reg}})$. The full model is $(1, 7, 0.1)$, the same run as EigenRR in Table 1. Blank rows in Table 3 drop either the basis regularizer, $(1, 7, 0)$, or the subspace term, $(0, 7, 0.1)$. Those four evaluations were too inaccurate to report.

Subspace loss alone reaches subspace error 0.1189 and mel error 0.011915; adding the mel term brings mel error to 0.007170, and 20 LOBPCG steps then reach 3.3×10^{-4} versus 2.4×10^{-4}. Setting the basis weight to 0, or supervising mel frequency alone at $(0, 7, 0.1)$, leaves no usable test result. The subspace term supplies the directions, the mel term tightens frequency, and the regularizer keeps the basis at unit mass energy so that orthogonalization stays stable.

5.5 Forward Sound Synthesis

We synthesize strikes with Equation (3) from ground-truth modes and from EigenRR, NeuralSound, VibraVerse, and random initialization. Figure 3 shows glass struck at vertex 3485 and wood struck at vertex 2089, both after 20 LOBPCG steps, under one spectrogram setting.

Glass energy sits below about 4 kHz and decays; wood decays within about 0.2 s. On log-magnitude spectra within 8 kHz and a 512-point window, mean squared error against ground truth is about 88, 97, 111, and 2304 (EigenRR, NeuralSound, VibraVerse, random) for glass, and about 26, 43, 133, and 1328 for wood. EigenRR is closer than NeuralSound on both samples. NeuralSound partials sit high, as expected while Ritz values remain upper bounds. VibraVerse misses the ground-truth

Table 3: Loss-weight ablation. Entries in parentheses are subspace / mel / basis regularization. "—" means the error was too large to report. The best result in each column is in bold.

	Forward only			LOBPCG20		
Setting	Sub. ↓	Eig. ↓	Mel ↓	Sub. ↓	Eig. ↓	Mel ↓
Subspace only $(1/0/0.1)$	0.1189	1.0203	0.011915	0.0722	0.0142	0.000333
Subspace + mel $(1/7/0)$	—	—	—	—	—	—
Full $(1/7/0.1)$	**0.0995**	**0.4683**	**0.007170**	**0.0630**	**0.0099**	**0.000236**
Mel only $(0/7/0.1)$	—	—	—	—	—	—

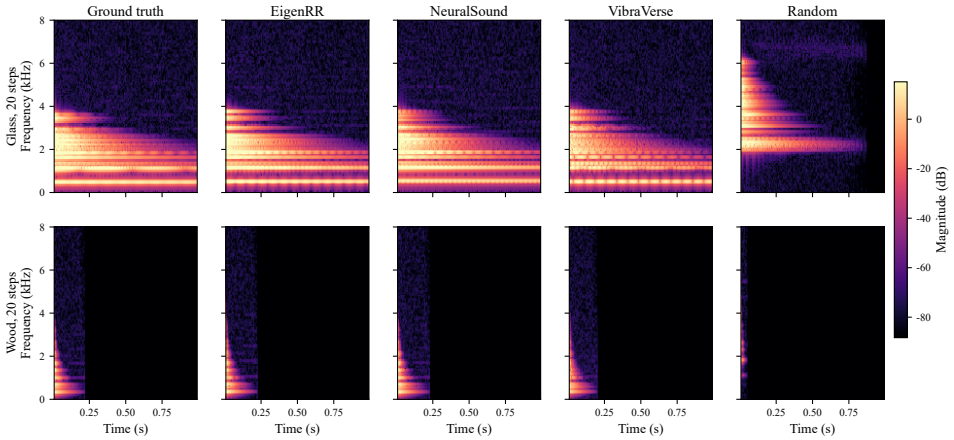

Figure 3: Forward-synthesis spectrograms after 20 LOBPCG steps. Top: glass, strike vertex 3485. Bottom: wood, strike vertex 2089. Our EigenRR shows the best performance close to GT.

partials, and random initialization either goes silent early or adds a high band the ground truth lacks. These two samples are not a test-set metric.

5.6 Impact-Point Estimation

Given a sound and a mesh, we score each vertex by the cosine similarity between its unit-strike modal amplitudes and the amplitudes implied by the sound, and take the top vertex as the predicted strike. Rank 1 means the ground-truth vertex wins.

Figure 4 is qualitative: high scores sit near the strike, and we do not yet report a test-set hit rate. The task needs eigenvectors.

5.7 Comparison with Eigenvalue-Only VibraVerse

VibraVerse regresses frequencies and has no mode shapes. Since A_i in Equation (3) is the strike's projection onto those shapes, a change of strike point does not change its audio.

Figure 5 shows three strikes on one object. EigenRR's spectrum changes with location; VibraVerse's does not. Frequency error in Table 1 is still comparable, but VibraVerse cannot synthesize location-dependent impacts or solve Section 5.6.

5.8 Mode-Shape Visualization

Figure 6 in the appendix visualizes mode 10 on a glass cup. Rows x, y, and z are displacement components, blue–white–red for sign, with prediction and ground truth side by side. The last row is per-vertex cosine similarity: green is aligned (near $+1$), yellow is reversed (near -1).

The 0-step prediction is not yet an aligned mode. The direction aligns by 20 steps, and the components themselves by 50.

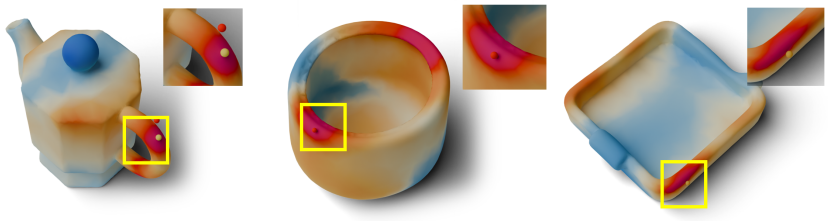

Figure 4: Strike-point similarity. Cosine similarity between per-vertex modal amplitudes and the target sound for a teapot, cup, and tray. Regions are zoomed in with dots marking strike locations.

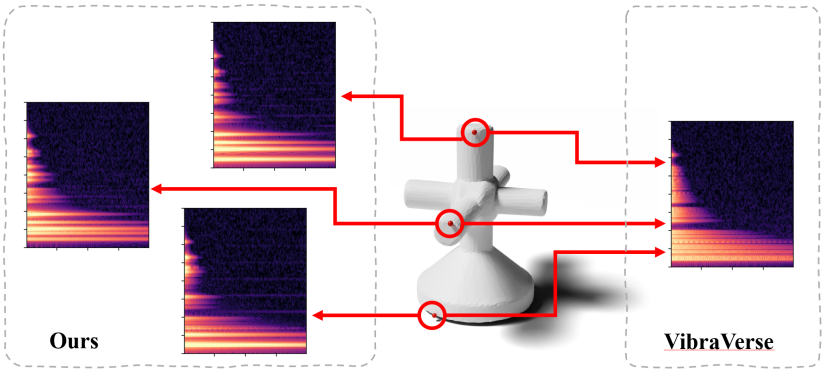

Figure 5: Strike-location-dependent spectra. Our method produces distinct spectra for three strike points, whereas VibraVerse uses a single eigenvalue set and therefore yields the same spectrum regardless of strike location.

6 Conclusion

EigenRR learns an overcomplete search subspace on tetrahedral meshes and extracts modal eigenpairs through differentiable Rayleigh–Ritz projection on each object's FEM operator. Its central idea is to learn useful vibration directions in the discretization where the eigenproblem is solved, amortizing the search across objects while retaining operator-based extraction and refinement. The resulting Ritz modes provide fast estimates and can warm-start LOBPCG when higher accuracy is needed. Experiments show an improved accuracy–latency trade-off over NeuralSound and random initialization; the recovered mode shapes also support strike-location-dependent synthesis and a qualitative demonstration of strike-point estimation. More broadly, this work suggests a route for learning-assisted eigensolving: networks supply compact search spaces that established numerical methods can use and progressively refine. We release VibraVerse++, including tetrahedral meshes, assembled FEM matrices, and reference spectra.

Limitations and future work. The method assumes linear elasticity and one homogeneous material per object. The Rayleigh residual needs no labels, so self-supervised fine-tuning on unlabeled meshes is a direct next step, as are anisotropic and multi-material objects and coupling the modes to an acoustic-transfer solver.

AI Usage Statement

Generative models enter this work only through the construction of VibraVerse++, described in Section 5.1. Renders of Objaverse assets are captioned by a vision-language model and passed through the same text-to-image and image-to-3D cascade as VibraVerse.

Large language models were also used to polish the English text. The method, the experiments, and the reported numbers were designed, implemented, and run by the authors. The authors reviewed

every AI-assisted passage against the code and the experimental logs. We take responsibility for the final content, including text written with language-model assistance.

REFERENCES

Jernej Barbic. Siggraph 2012 course notes fem simulation of 3d deformable solids: A practitioner's guide to theory, discretization and model reduction. part 2: Model reduction (version: August 4, 2012). *ACM Trans. Graph. (SIGGRAPH)*, 2012.

Ido Ben-Shaul, Leah Bar, Dalia Fishelov, and Nir Sochen. Deep learning solution of the eigenvalue problem for differential operators. *Neural Computation*, 35(6):1100–1134, 2023. doi: 10.1162/neco_a_01583.

Jeffrey N Chadwick, Steven S An, and Doug L James. Harmonic shells: a practical nonlinear sound model for near-rigid thin shells. *ACM Trans. Graph.*, 28(5):119, 2009.

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. *arXiv preprint arXiv:1512.03012*, 2015.

Indrajit Chowdhury and Shambhu P Dasgupta. Computation of rayleigh damping coefficients for large systems. *The Electronic Journal of Geotechnical Engineering*, 8(0):1–11, 2003.

Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13142–13153, 2023.

Ruohan Gao, Yen-Yu Chang, Shivani Mall, Li Fei-Fei, and Jiajun Wu. Objectfolder: A dataset of objects with implicit visual, auditory, and tactile representations. In *CoRL*, 2021.

Yixin Hu, Teseo Schneider, Bolun Wang, Denis Zorin, and Daniele Panozzo. Fast tetrahedral meshing in the wild. *ACM Transactions on Graphics (ToG)*, 39(4):117:1 – 117:18, 2020.

Doug L James, Jernej Barbič, and Dinesh K Pai. Precomputed acoustic transfer: output-sensitive, accurate sound generation for geometrically complex vibration sources. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2006)*, 25(3):987–995, 2006.

Xutong Jin, Sheng Li, Tianshu Qu, Dinesh Manocha, and Guoping Wang. Deep-modal: Real-time impact sound synthesis for arbitrary shapes. In *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 1171–1179, 2020. doi: 10.1145/3394171.3413572.

Xutong Jin, Sheng Li, Guoping Wang, and Dinesh Manocha. Neuralsound: learning-based modal sound synthesis with acoustic transfer. *ACM Transactions on Graphics (TOG)*, 41(4):1–15, 2022.

Xutong Jin, Chenxi Xu, Ruohan Gao, Jiajun Wu, Guoping Wang, and Sheng Li. Diffsound: Differentiable modal sound rendering and inverse rendering for diverse inference tasks. In *SIGGRAPH '24: ACM SIGGRAPH Conference Papers*, New York, NY, USA, 2024. Association for Computing Machinery. doi: 10.1145/3641519.3657493.

Stephen Kirkup. The boundary element method in acoustics: A survey. *Applied Sciences*, 9(8): 1642, 2019.

Andrew V Knyazev. Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. *SIAM Journal on Scientific Computing*, 23(2):517–541, 2001.

Richard B Lehoucq, Danny C Sorensen, and Chao Yang. *ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*. SIAM, 1998.

Chengqi Lin, Junzhe Ding, Jianlin Chen, Yuan Wang, Jianhao Jiao, Pui-In Mak, and Nan Wang. Graph-based surrogate modeling for structural eigenvalue problems with modal matching training. *Engineering with Computers*, 42:72, 2026. doi: 10.1007/s00366-026-02310-8.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

Bo Pang, Chenxi Xu, Jierui Ren, Guoping Wang, and Sheng Li. Vibraverse: A large-scale geometry-acoustics alignment dataset for physically-consistent multimodal learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

J. N. Reddy. *Introduction to the Finite Element Method*. McGraw-Hill Education, 4 edition, 2019.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 2015.

Conor Rowan, Alireza Doostan, Kurt Maute, and John Evans. Solving engineering eigenvalue problems with neural networks using the rayleigh quotient. *International Journal for Numerical Methods in Engineering*, 126(24):e70209, 2025. doi: 10.1002/nme.70209.

Kees Van Den Doel, Paul G Kry, and Dinesh K Pai. Foleyautomatic: physically-based sound effects for interactive simulation and animation. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 537–544, 2001.

Jui-Hsien Wang and Doug L. James. Kleinpat: Optimal mode conflation for time-domain precomputation of acoustic transfer. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2019)*, 38(4):122:1–122:12, July 2019. ISSN 0730-0301.

Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. O-CNN: Octree-based convolutional neural networks for 3D shape analysis. *ACM Trans. Graph. (SIGGRAPH)*, 36(4), 2017.

Peng-Shuai Wang, Yang Liu, and Xin Tong. Dual octree graph networks for learning adaptive volumetric shape representations. *ACM Trans. Graph. (SIGGRAPH)*, 41(4), 2022.

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*, 2024.