Final Project For ECE228 Track Number # 1

Group Number #15: Harini Gurusankar **Girish Krishnan**

Ryan Irwandy **Yash Puneet**

Abstract

Full-waveform inversion (FWI) retrieves subsurface velocity from multi-source seismic gathers but is notoriously expensive and noise-sensitive. Within the Kaggle FWI challenge[1] we benchmark several neural operators on a 2 000-sample OpenFWI subset (5 sources, 1000×70 grid). The candidates are Fourier DeepONet, InversionNet, VelocityGAN, residual UNet, 2-D Fourier Neural Operator, three (Augmented) Neural ODEs, and a PCA + MLP baseline and are trained end-to-end to predict 70×70 velocity maps. Fourier DeepONet delivers the lowest mean-absolute error, with VelocityGAN and InversionNet close behind, while Neural-ODE and PCA models lag by roughly an order of magnitude.

1 Introduction

Geological waveform inversion is used to image subsurface structures by analyzing how seismic waves propagate through the Earth. Sources like drills generate waves recorded by receivers (Figure 1), from which we estimate the underground velocity map $c(x, z)$. High velocities suggest dense materials, while low velocities may indicate voids or softer regions. These maps are crucial for oil exploration, infrastructure planning, and earthquake fault detection.

The forward process is modeled by the acoustic wave equation:

$$\nabla^2 p(x, z, t) - \frac{1}{c(x, z)^2} \frac{\partial^2 p(x, z, t)}{\partial t^2} = s(x, z, t)$$

where p is the seismic wavefield and s is the source. Inversion recovers $c(x, z)$ from observed p.

While underground receivers offer accurate data, they are costly and disruptive. Surface receivers are cheaper but introduce significant noise. Our work explores learning-based approaches to infer velocity maps from surface seismic data using various neural architectures.

We evaluate several models, including Fourier DeepONet, Residual UNet, InversionNet, VelocityGAN, 2D Fourier Neural Operator, Augmented Neural ODEs, and a PCA-based MLP. Fourier DeepONet achieved the best mean absolute error on our testing set, placing us **591st of 6446 entrants** in the Kaggle competition.

Contributions. Brief summary of our contributions (see Appendix for full breakdown):

- **Fourier DeepONet:** Implemented spectral DeepONet with U-Net decoder; best performance on both datasets; evaluated noise robustness. [Girish]
- **Neural ODEs:** Built MLP and CNN-based Neural ODEs, including augmented variants. [Harini]
- **VelocityGAN & InversionNet:** Reproduced and trained both baselines. VelocityGAN performed well; InversionNet overfit. [Girish]

[1] Competition link: `https://www.kaggle.com/competitions/waveform-inversion`

Preprint. Under review.

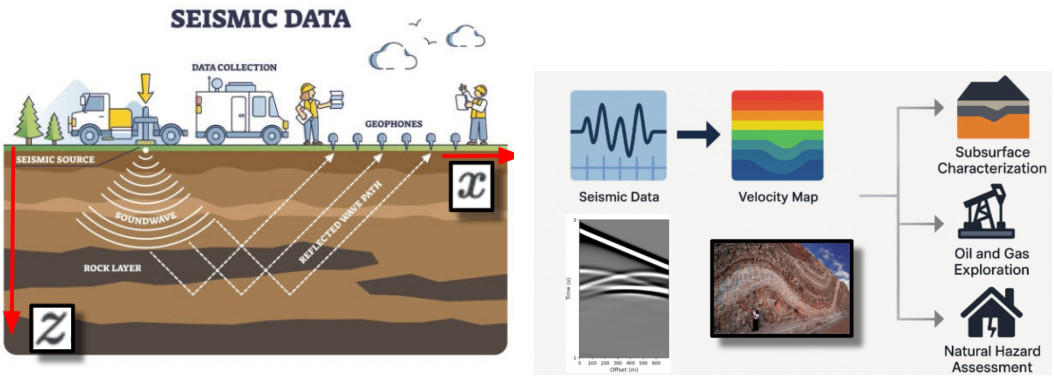


Figure 1: (Left) Surface-seismic acquisition. (Right) High-resolution velocity model inferred from recorded data.

- **Residual UNet:** Trained deeper UNet variant with skip connections. [Girish]
- **Fourier Neural Operator:** Adapted 2D FNO; stable but not state-of-the-art. [Yash]
- **PCA + MLP:** Designed a compact baseline combining PCA with MLP. Motivates future PCA-CNN variants. [Ryan]

2 Related work

Learning-based FWI. CNN surrogates *InversionNet* and cGAN-based *VelocityGAN* replace iterative adjoints, giving 10-100$\times$ speed-ups but suffer from (i) narrow receptive fields, (ii) poor long-range physics, and (iii) fragility to noise and domain shift. *Fourier DeepONet* (F-DONet) [12] mitigates (i)-(ii) with spectral convolutions and branch-trunk conditioning, setting the current OpenFWI SOTA, yet is compared only to the two baselines above.

Operator networks. Fourier Neural Operators (FNO) [5] provide mesh-independent spectral layers; Augmented Neural ODEs (ANODE) [1] offer continuous-depth models. Neither has been evaluated on seismic gathers and both face open practical issues: GPU memory (FNO) and solver stability (ODEs).

Dimensionality reduction. Receiver redundancy has motivated statistical preprocessing (e.g. PCA for gas-hydrate analysis [3]), but its merit for high-fidelity FWI regression is untested.

We bridge these gaps by (i) benchmarking seven architectures—UNet, InversionNet, VelocityGAN, F-DONet, 2-D FNO, Neural ODE, ANODE, and a PCA + MLP baseline—on a common OpenFWI split; (ii) probing their robustness across depth, velocity, and domain shift (Kaggle test); and (iii) scaling the best model via distributed training to the full 622 GB dataset, advancing practical SOTA in learned FWI.

3 Methodology

This section describes the different models we implemented along with their training strategies.

3.1 Fourier DeepONet

Fourier DeepONet (F-DONet) maps input seismic data $u \in \mathbb{R}^{5\times1000\times70}$ and dummy parameter $p \in \mathbb{R}$ to velocity fields $v \in \mathbb{R}^{70\times70}$. The *branch* net $\mathcal{B}(u) \in \mathbb{R}^{w\times70\times70}$ and *trunk* net $\mathcal{T}(p) \in \mathbb{R}^{w\times1\times1}$ are combined via element-wise product with bias to yield $z_0 = \mathcal{B}(u) \odot \mathcal{T}(p)$.

The decoder applies four stacked U-Fourier blocks followed by projection:

$$z_i = \text{U-Fourier}(z_{i-1}), \quad \hat{v} = Q(z_4),$$

where each U-Fourier block performs:

$$\text{U-Fourier}(z) = \sigma\big(\text{FourierConv}(z) + \text{Ublock}(z)\big).$$

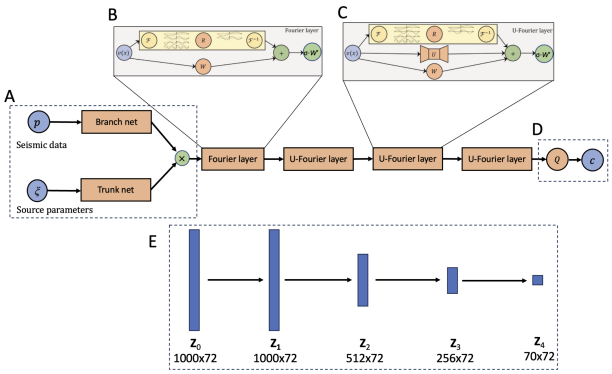
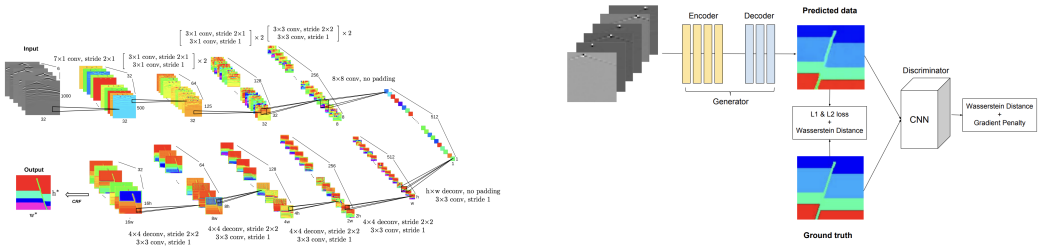


Figure 2: **Fourier-DeepONet architecture.** **(A)** Branch $\mathcal{B}$ and trunk $\mathcal{T}$ networks lift inputs to latent space; merger denotes pointwise multiplication. **(B)** Fourier layer [4]. **(C)** U-Fourier block [9]. **(D)** Projection layer Q. **(E)** Feature maps $\mathbf{z}_i$, $i = 0, \ldots, 4$. *Source*: [12].



(a) InversionNet encoder-decoder.

(b) VelocityGAN architecture.

Figure 3: **(a)** InversionNet: 3D seismic input $(B, 5, 1000, 70)$ is compressed via conv layers and decoded into a Tanh-activated 70×70 velocity map. **(b)** VelocityGAN: InversionNet serves as generator G, paired with patch-wise CNN discriminator D.

Here, $\text{FourierConv}(z) = \mathcal{F}^{-1}(R_\phi \cdot \mathcal{F}(z))$, with $\mathcal{F}$ denoting the 2D FFT.

F-DONet thus combines latent conditioning with multiscale spectral processing for efficient operator learning. See Figure 2 for an overview.

3.2 InversionNet and VelocityGAN

InversionNet [10] is a CNN-based encoder-decoder that maps seismic inputs to velocity maps (Fig. 3a). The encoder compresses the $5 \times 1000 \times 70$ input into a latent code, which the decoder upsamples to a 70×70 output using Tanh to match the normalized velocity range.

VelocityGAN [11] extends InversionNet by using it as a generator G in a conditional GAN framework (Fig. 3b). A discriminator D assigns patch-wise real/fake scores. The generator minimizes:

$$\mathcal{L}_G = \text{BCE}(D(G(s)),\, \mathbf{1}) + \lambda \|G(s) - v\|_1,$$

where $\lambda = 100$ weights the L_1 loss. The discriminator loss is:

$$\mathcal{L}_D = \tfrac{1}{2}\big[\text{BCE}(D(v),\, \mathbf{1}) + \text{BCE}(D(G(s)),\, \mathbf{0})\big].$$

This setup balances adversarial training with pixel-wise fidelity, encouraging realistic velocity map generation.

3.3 Residual UNet

Residual UNet adapts the widely used U-Net [8] by integrating residual connections from ResNet [2] into each convolutional block. As shown in Figure 4, the network comprises a down-sampling path that applies multiple *ResidualDoubleConv* modules to encode the 5×1000×70 seismic input into

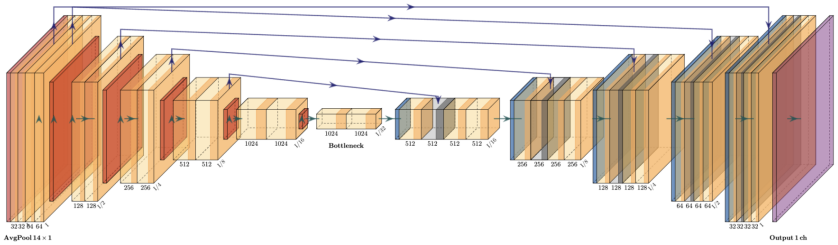
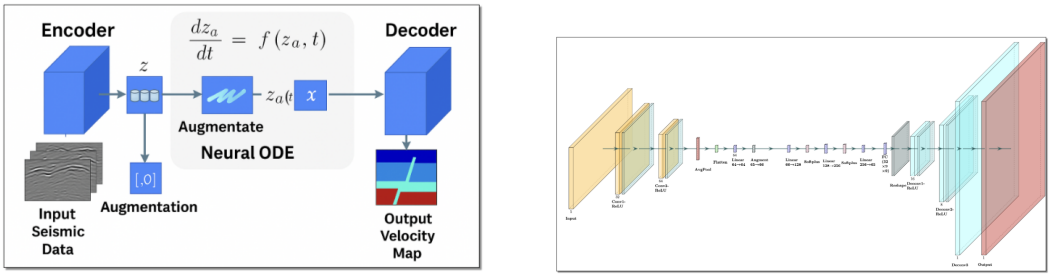


Figure 4: Residual UNet architecture for seismic-to-velocity inversion. The network uses an encoder to compress the input seismic cube into latent features and a decoder with symmetric up-sampling paths that leverage skip connections. Residual blocks ensure stable gradient flow and enable training of deeper architectures.



(a) General structure

(b) Augmented MLP base

Figure 5: Augmented Neural ODE architectures. (a) shows the general structure, where input seismic data is encoded, augmented, solved via ODE integration, and decoded to produce the velocity map. (b) shows the specific Augmented Neural ODE model using an MLP base function.

a lower-dimensional latent representation. In the up-sampling path, each decoder block receives concatenated features from its encoder counterpart via skip connections, followed by residual up-convolution to reconstruct a 70×70 velocity map.

Formally, each residual block computes: $\mathbf{y} = \sigma\big(\mathrm{BN}(W_2 * \sigma(\mathrm{BN}(W_1 * \mathbf{x}))) + \mathcal{S}(\mathbf{x})\big)$, where $*$ denotes convolution, BN denotes batch normalization, σ is ReLU, and $\mathcal{S}$ is either identity or a learned linear shortcut. These shortcuts enable smoother gradient propagation through the network.

3.4 Neural ODE

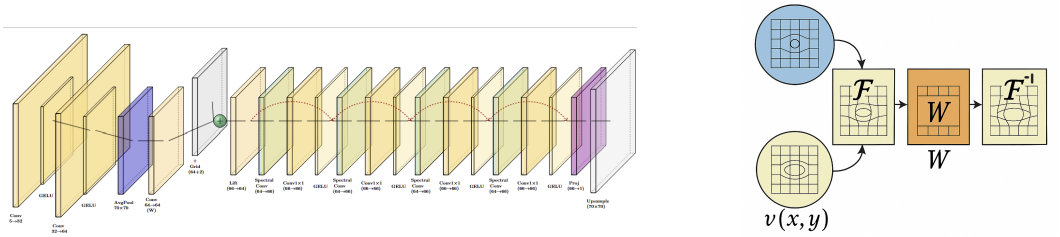
A Neural ODE treats depth as continuous. A conv-encoder E maps the seismic cube $u \in \mathbb{R}^{5\times1000\times70}$ to an initial latent $z_0 \in \mathbb{R}^d$. Latent evolution is defined by a time-invariant vector field f_θ (3-layer MLP) via the ODE: $\dot{z}(t) = f_\theta\big(z(t)\big), \qquad z(0) = z_0$, integrated with `odeint` (`torchdiffeq`). The terminal state $z(1)$ is decoded by a transposed-conv head $D : \mathbb{R}^d \to \mathbb{R}^{1\times70\times70}$, yielding $\hat{v} = D\big(z(1)\big)$ (Fig. 5a).

Augmented variants. Following Dupont et al. [1], we append one inert dimension a so that $\tilde{z} = (z, a)$ and integrate $\dot{\tilde{z}} = (f_\theta(\tilde{z}), 0)$. We consider two choices for f_θ: (i) the same MLP (Fig. 5b) and (ii) a shallow CNN, giving three ODE models in total.

3.5 2-Dimensional Fourier Neural Operator

To directly learn the space-to-space map $u \mapsto v$ we adapt the Fourier Neural Operator [5] to 2-D seismic gathers (time × receiver). A conv-encoder first compresses u to a $70\times70\times c$ feature grid, where c is the number of channels (a hyperparameter). After concatenating a positional grid $g \subset [0,1]^2$ (step 0.2), the tensor passes through $L = 4$ spectral blocks (Fig.6):

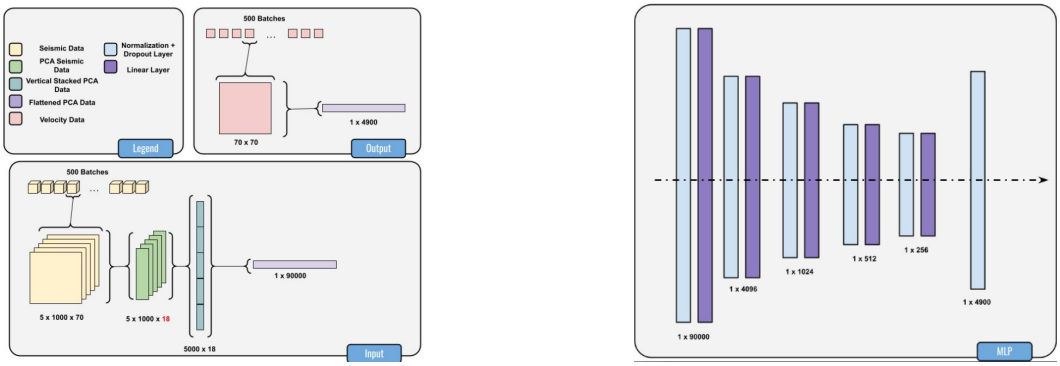
$$\mathrm{Spectral}(x) = \mathcal{F}^{-1}\big(W_{\text{spec}} \cdot \mathcal{F}(x)\big) + W_{\text{local}} * x,$$



(a) 2D Fourier Neural Operator

(b) Spectral Convolution Block

Figure 6: (a) Full 2D Fourier Neural Operator architecture used in our experiments. (b) Internal schematic of the spectral convolution block.



(a) PCA Data Preparation Pipeline.

(b) PCA-MLP Architecture.

Figure 7: (a) Seismic waveform data reshaped and reduced via PCA. (b) MLP model applied to PCA-transformed data.

where $\mathcal{F}$ is the 2-D FFT, W_{spec} keeps the lowest 32 modes, and the second term is a 1×1 convolution (skip connection). A linear projection converts the final feature map to $\hat{v} \in \mathbb{R}^{1\times 70\times 70}$.

3.6 Principal Component Analysis (PCA) along with MLP

We form a lightweight baseline by reducing the receiver-axis with Principal Component Analysis (PCA). For each shot $u \in \mathbb{R}^{5\times 1000\times 70}$ we stack sources into a 5000×70 matrix, apply z-score normalisation, and project onto the top $k = 18$ components ($\approx 95\%$ variance). Flattening yields a vector $x \in \mathbb{R}^{5000\times k}$ that feeds a four-layer MLP with LayerNorm $\rightarrow$ ReLU $\rightarrow$ Dropout. The network outputs $\hat{v} \in \mathbb{R}^{1\times 70\times 70}$ after a final reshape (Fig.7). This model tests how far linear dimensionality reduction can go before spatial structure must be learned end-to-end.

4 Experiments

4.1 Dataset and Data Processing

OpenFWI is a public, 622 GB benchmark that gathers multifold synthetic seismic-to-velocity pairs generated with finite-difference forward modelling over a variety of 2-D geological scenarios (e.g. *FlatVel*, *CurveVel*, *FlatFault*, *CurveFault*, *Style*) [6, 7]. Each sample consists of a tensor of raw pressure gathers $\mathbf{s} \in \mathbb{R}^{5\times 1000\times 70}$ (five sources, 1000 time steps, 70 receivers) and its corresponding velocity map $v \in \mathbb{R}^{1\times 70\times 70}$. For rapid prototyping we curate a *small subset* composed of one 500-sample batch from the categories `CurveFault_A`, `CurveVel_A`, `FlatFault_A`, and `Style_A` (2 000 samples in total). After min-max normalization of velocity (1500-4500 m/s) and log-amplitude scaling of gathers, the subset is randomly divided 70% / 20% / 10% into training, validation, and test partitions. The best architecture found on this subset is subsequently trained on the *full* OpenFWI corpus and submitted to the Kaggle "Geophysical Waveform Inversion" leaderboard.

Model	Opt.	LR	B	Ep.	Loss	Scheduler / key architectural settings
Fourier DeepONet	Adam	1×10^{-3}	4	100	ℓ_1	width 64, modes 20×20
InversionNet	Adam	1×10^{-3}	8	100	ℓ_1	—
VelocityGAN (G)	Adam	2×10^{-4}	8	50	BCE $+\,\lambda\ell_1$	$\lambda=100,\ \beta=(0.5,0.999)$
VelocityGAN (D)	Adam	2×10^{-4}	8	50	BCE	$\beta=(0.5,0.999)$
Residual UNet	Adam	1×10^{-3}	8	100	ℓ_1	depth 5, init-feat 32
2-D FNO	Adam	5×10^{-3}	4	250	ℓ_1	StepLR ($\times0.5$/75ep), modes 32×32, width 64, layers 4
Neural ODE (MLP)	Adam	1×10^{-3}	8	50	ℓ_1	StepLR ($\times0.5$/10ep), tol 10^{-3}, max 100 steps
Aug. ODE (MLP)	Adam	1×10^{-3}	8	50	ℓ_1	as above, aug. dim 1
Aug. ODE (CNN)	Adam	1×10^{-3}	8	50	ℓ_1	CNN base, aug. dim 1
PCA + MLP	Adam	1.5×10^{-1}	16	100	MSE	StepLR ($\times0.5$/15ep), hidden [2048,512,128], drop 0.05

Table 1: Final hyper-parameters. LR = learning rate, B = batch size, Ep. = epochs.

Model	Val. MAE ↓	Test MAE ↓	RMSE ↓	SSIM ↑	Rel. L_2 ↓
Fourier DeepONet	59.5	**71.4**	**76.3**	**0.916**	**0.029**
VelocityGAN	**52.3**	72.4	114.7	0.904	0.044
InversionNet	68.1	82.5	135.6	0.634	0.052
Residual UNet	68.4	72.4	146.4	0.903	0.049
Fourier Neural Operator (2-D)	124.2	106.9	166.7	0.727	0.056
Neural ODE (MLP)	468.2	471.5	595.0	0.879	0.199
Aug. Neural ODE (MLP)	468.0	471.4	595.2	0.876	0.199
Aug. Neural ODE (CNN)	468.0	471.4	595.2	0.876	0.199
PCA + MLP	311.8	308.0	430.0	0.707	0.151

Table 2: Performance on the held-out test split ($N=200$). Best value in each column is **bold**.

4.2 Hyperparameters and Training

All models were trained on a single GPU (provided on DSMLP) with mixed-precision (`torch.cuda.amp`) and the same 70 / 20 / 10 train-val-test split described earlier. Core hyper-parameters were selected with a 40–trial Optuna Bayesian search per model, using validation MAE as the objective. Table 1 summarizes the final settings.

Training protocol. Gradients were left unclipped; weight decay was 10^{-4} for operator models and omitted elsewhere. Mini-batches were shuffled each epoch. During hyperparameter tuning, early stopping (15 epoch patience) selected the best checkpoint, which was then retrained from scratch using the final configuration. All test scores in Table 2 reflect these final retrained models.

4.3 Quantitative Results and Ablation Studies

We evaluate predicted velocity maps $\hat{v}$ against ground truth v using four metrics: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Structural Similarity Index (SSIM), and relative ℓ_2 error. MAE is the primary metric used in the Kaggle leaderboard. All scores are computed on the test split, with MAE on the full dataset reported for the best model. See Appendix 5 for definitions. Table 2 reports validation MAE (used for early stopping) and four test-set metrics for all nine models. Lower is better except for SSIM, where higher indicates greater structural fidelity.

Since the 2D FNO and Neural ODE models initially did not perform as well as the state-of-the-art models (e.g. Fourier DeepONet, InversionNet), we performed hyper-parameter sweeps to identify the best configurations for these models. The results of these ablations are summarized in Figure 8.

Table 3: 2-D FNO ablations (bold = final choice).

Parameter	Explored values and observations
# spectral blocks	2: coarse details 5: heavy over-fit **4**: best MAE, mild over-fit
Width	16, 32: under-fit **64**: better MAE, tolerable run-time
Fourier modes	4–16: jagged maps **32**: captures fine structure 64: GPU OOM / worse generalization
Epochs	10: under-train 200: modest gain **250**: sweet-spot before over-fit
Batch size	4: slow **10**: stable / memory-safe 20: occasional OOM, noisier loss

Table 4: Neural-ODE ablations (MLP unless noted).

Parameter	Final choice and rationale
Hidden dimension	**64**; 128 slowed training, no MAE gain
Learning rate	1×10^{-3}; 1×10^{-4} too slow, 3×10^{-4} used for CNN base
Batch size	**8**; 32 caused GPU OOM
Solver tol./steps	10^{-3} **/ 100**; faster convergence
Augmented dim.	**1**; adds capacity with negligible cost
Weight decay	5×10^{-5} (CNN base only)

Figure 8: Summary of hyper-parameter sweeps used to select the final configurations for two models: 2-D Fourier Neural Operator (FNO) and Neural ODE. The tables list the parameters explored, their values, and the observations that led to the final choices.

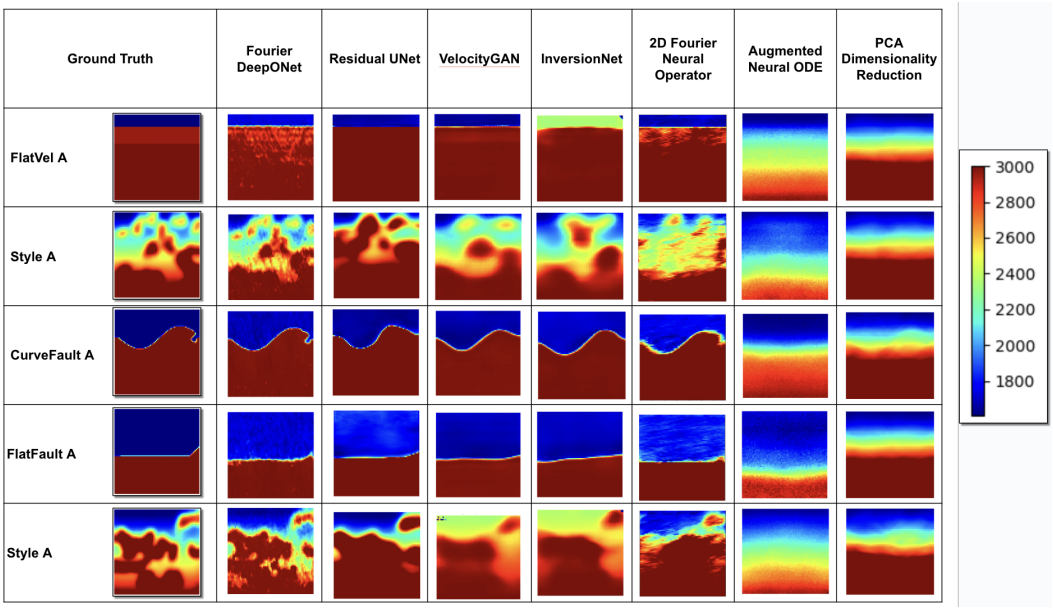


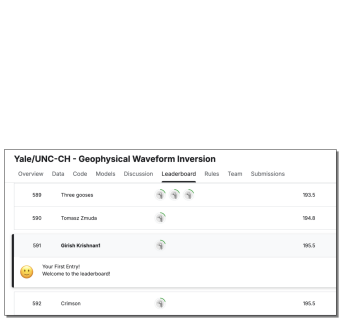
Figure 9: Qualitative comparison of predicted velocity maps from different models against the ground truth. Each row corresponds to a different sample in the testing set, with the first column showing the ground truth velocity map. The subsequent columns show the predictions from each model. The colorbar indicates the velocity scale in m/s. The Fourier DeepONet predictions closely match the ground truth, while other models exhibit varying degrees of deviation.

4.4 Qualitative Results

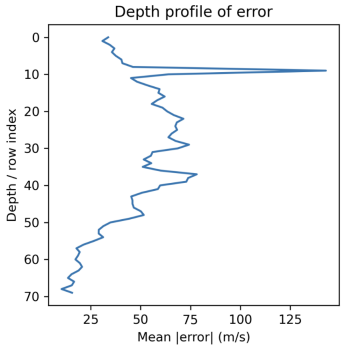
Figure 9 shows qualitative results for the best-performing model (Fourier DeepONet) and the baseline InversionNet. The predicted velocity maps exhibit high structural fidelity, with the Fourier DeepONet capturing fine details and long-range correlations better than the other models.

4.5 Discussion and Model Comparison

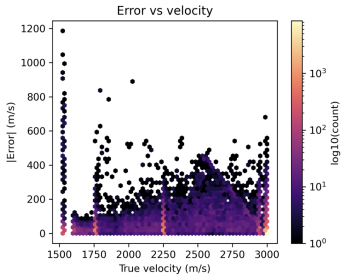
Fourier DeepONet leads across all metrics, likely due to its spectral operator structure and multiscale skip connections. VelocityGAN achieves the lowest validation MAE but overfits slightly, while



(a) Public-leaderboard rank (591/6 446).



(b) MAE vs. depth.



(c) MAE vs. velocity.

Figure 10: Fourier-DeepONet submission analysis on the hidden Kaggle test set. Figure (a) shows our current public leaderboard rank of 591 out of 6,446 participants. Figures (b) and (c) plot the mean absolute error (MAE) of our predictions against depth and velocity, respectively. This gives us insight into how well the model generalizes across different geological scenarios.

Residual UNet performs well on MAE yet has larger RMSE, suggesting it captures coarse structure but misses fine features. FNO lags in absolute error due to bandwidth constraints. Neural ODE variants and the PCA baseline perform worst, showing the need for spatial priors and expressive decoders.

4.6 Kaggle Competition Submission

To scale to the full 622 GB dataset, we trained Fourier DeepONet using `torchrun` on 2×2080 GPUs with per-GPU batch size 4, FP16, DDP, and AdamW (lr=3×10^{-4}, weight decay 10^{-4}, EMA with α=0.999). Early stopping triggered after 15 epochs. Training converged in 80 epochs (18 GPU-hrs). The resulting model achieved an MAE of 195.5 m/s on the hidden Kaggle test set. The gap with our in-house validation (71.4 m/s) suggests domain shift, especially at greater depths and velocities (Figure 10b–c), motivating future work in adaptation.

5 Conclusion

Summary and Key Insights This project investigated a variety of data-driven models for seismic-to-velocity inversion. We implemented baseline architectures (InversionNet, VelocityGAN, Residual UNet), advanced neural operators (Fourier DeepONet, FNO), and a PCA-MLP dimensionality reduction pipeline. Among these, the Fourier DeepONet achieved the lowest test error on the Kaggle competition set when trained on the full OpenFWI dataset. Architectures that preserved spatial structure and captured long-range correlations consistently outperformed simpler MLP-based models.

What Worked and What Did Not Convolutional and operator-based models like UNet, DeepONet, and FNO were the most effective, as they leveraged spatial continuity and hierarchical features. Lightweight models like PCA-MLP were fast but less accurate due to loss of spatial information. Training on the reduced Kaggle subset led to frequent overfitting, indicating the need for larger-scale data or better regularization.

Future Directions Future work can focus on three areas:

- **Noise Robustness:** Evaluate model stability under input perturbations to simulate real-world sensor noise, following findings in [12].
- **Full-Scale Training:** Extend training of all models to the full OpenFWI dataset. Currently, only the Fourier DeepONet has been fully scaled.
- **Dimensionality Reduction with Stronger Backbones:** Replace the MLP in PCA-MLP with spatially-aware architectures like CNNs or Residual UNet to improve performance while retaining the benefits of dimensionality reduction.

Appendix

GitHub Link

The link to our GitHub repository is as follows: `https://github.com/Girish-Krishnan/ECE-228-Final-Project/tree/main`.

This repo contains all the code used to run our models, including the data preprocessing, model training, and evaluation scripts. The `README.md` file provides detailed instructions on how to set up the environment, install dependencies, and run the code to reproduce our results.

Team Member Contribution

Table 5 summarizes the contributions of each team member to the project. The goal was to have each team member implement at least 1-2 different model architectures so that we have several different models to compare against each other. The team members were also responsible for writing the report and poster sections that corresponded to their model implementations.

Member	Key Contributions
Harini Gurusankar	• Designed the baseline *Neural ODE* and *Augmented Neural ODE* models. • Implemented MLP and CNN-based ODE variants in PyTorch. • Wrote the methodology subsection and results discussion for ODE models. • Contributed methodology content to the poster and presented it.
Girish Krishnan	• Coordinated the project and designed the experimental pipeline. • Implemented *Fourier DeepONet*, residual *UNet*, *InversionNet*, and *VelocityGAN*. • Trained and submitted full-data DeepONet (best MAE: 59.5 m/s). • Created metric visualizations (bar charts, heatmaps, radar plots). • Wrote the abstract and results sections. • Designed and presented the results section of the poster.
Ryan Irwandy	• Developed the PCA-based baseline and accompanying MLP model. • Wrote PCA utilities and normalization scripts. • Ran experiments on component counts and dropout strategies. • Wrote parts of the related work and methodology sections. • Created and presented the poster's introduction and PCA baseline sections.
Yash Puneet	• Implemented the 2-D *Fourier Neural Operator* with spectral blocks. • Tuned mode counts and channel widths via hyperparameter sweeps. • Created architecture figures for the FNO section. • Wrote FNO-related work, methodology, and future work sections. • Designed and presented the related work and future work sections of the poster.

Table 5: Summary of individual contributions to the project.

Confirmation of Teaching Evaluation Submission (Bonus Points)

Our team confirms that we have submitted our teaching evaluation for the ECE 228 course and the teaching assistant evaluations. Here are screenshots that confirm our submission (Figure 11).

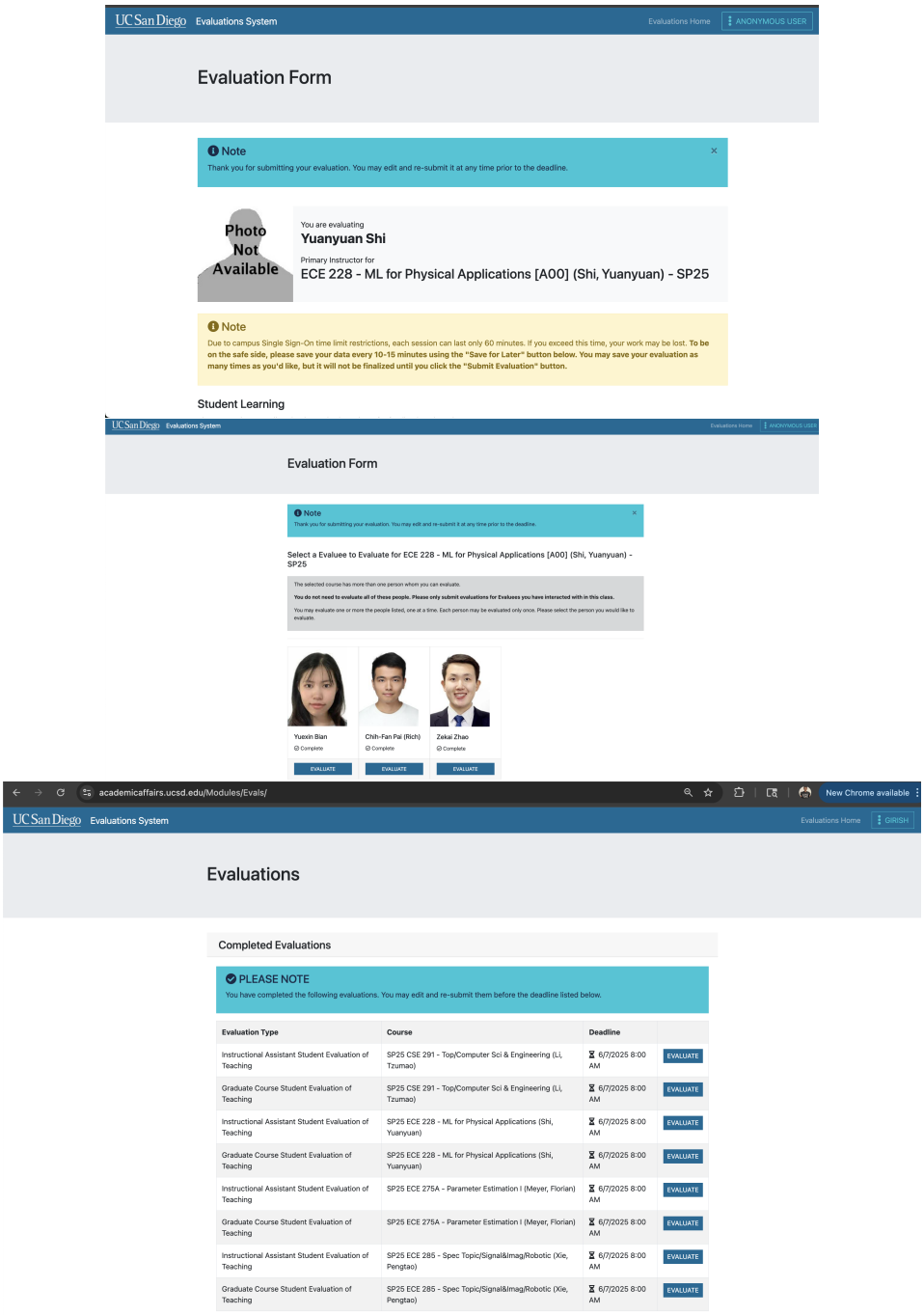


Figure 11: Teaching Evaluation Submission Confirmation

Thanks for teaching us this awesome course!

Metric	Description	Formula
MAE	Measures average absolute difference between predicted and ground truth values. Also the primary metric used in the Kaggle competition.	$\frac{1}{N}\sum_{i=1}^{N}\lvert y_i - \hat{y}_i\rvert$
RMSE	Penalizes large errors more than MAE; helpful for evaluating overall accuracy.	$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$
SSIM	Captures perceptual similarity by comparing luminance, contrast, and structure. Useful for assessing structural fidelity.	$\frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$
Relative ℓ_2 Error	Measures the relative difference scaled by the ground truth magnitude. Provides a normalized global error measure.	$\frac{\lVert y - \hat{y}\rVert_2}{\lVert y\rVert_2}$

Table 6: Evaluation metrics used to quantify model performance on velocity map prediction. In the SSIM formula, μ_x and μ_y denote the local means of the predicted and ground truth images respectively, σ_x^2 and σ_y^2 are their local variances, σ_{xy} is the local covariance, and C_1, C_2 are small constants that stabilize the division to avoid numerical instability.

Evaluation Metrics Formulae

Table 6 summarizes the evaluation metrics used to quantify model performance on velocity map prediction.

References

[1] E. Dupont, A. Doucet, and Y. W. Teh. Augmented neural odes, 2019. URL `https://arxiv.org/abs/1904.01681`.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[3] E. Jones, R. Smith, and J. Doe. Waveform seismic event classification using modern machine learning techniques. *Nuclear Engineering and Design*, 395:112456, 2023. doi: 10.1016/j.nucengdes.2023.112456.

[4] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020. URL `https://arxiv.org/abs/2010.08895`.

[5] Z. Li, N. B. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. L. Stuart, and A. Anandkumar. Fourier neural operator for parametric partial differential equations. *Journal of Computational Physics*, 447:110668, 2021.

[6] F. Meng, W. Sun, and et al. Openfwi: Benchmark datasets for data-driven seismic full waveform inversion. arXiv:2112.06742, 2021.

[7] F. Meng, W. Sun, and et al. Openfwi 2.0: Benchmark datasets for elastic full-waveform inversion. In *SEG IMAGE*, 2023.

[8] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 9351: 234–241, 2015.

[9] G. Wen, Z. Li, K. Azizzadenesheli, A. Anandkumar, and S. M. Benson. U-FNO—an enhanced fourier neural operator-based deep-learning model for multiphase flow. *Advances in Water Resources*, 163:104180, 2022. doi: 10.1016/j.advwatres.2022.104180.

[10] Y. Wu and Y. Lin. Inversionnet: An efficient and accurate data-driven full waveform inversion. *IEEE Transactions on Computational Imaging*, 6:419–433, 2020. doi: 10.1109/TCI.2019.2956866.

[11] Z. Zhang, Y. Wu, Z. Zhou, and Y. Lin. Velocitygan: Subsurface velocity image estimation using conditional adversarial networks. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 705–714, 2019. doi: 10.1109/WACV.2019.00080.

[12] M. Zhu, Y. Hu, C. Wu, X. Song, Z. Zhang, Z. Li, Y. Fu, and G. E. Karniadakis. Fourier-deeponet: Fourier-enhanced deep operator networks for full waveform inversion with improved accuracy, generalizability, and robustness, 2023. URL `https://arxiv.org/abs/2305.17289`.