# 3D Fresnel Volumizing for Efficient Implicit Velocity Field Reconstruction

**Sihan Chen***
School of Qilu Transportation
Shandong University
Jinan, China 250061
sihanc12@gmail.com

**Haochen Sun***
School of Computer Science and Technology
Shandong University
Qingdao, China 266237
sunhaochent@gmail.com

**Peng Jiang**†
School of Qilu Transportation
Shandong University
Jinan, China 250061
sdujump@gmail.com

**Anthony G. Cohn**
School of Computer Science
University of Leeds
Leeds LS2 9JT, U.K.
A.G.Cohn@leeds.ac.uk

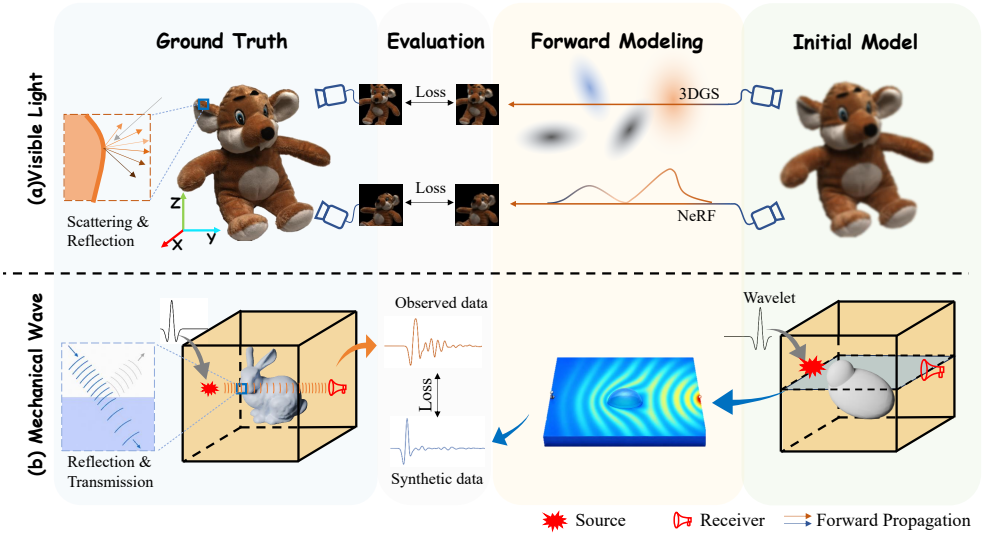


Figure 1: **Comparison of reconstruction paradigms.** (a) Appearance reconstruction based on visible images. (b) Internal velocity reconstruction based on wave signals.

## Abstract

Reconstructing velocity distributions using wavefield responses provides a powerful tool for probing the internal structures of objects and has been widely applied in nondestructive testing, geological exploration, and medical imaging. While full waveform inversion (FWI) provides high-resolution results, its severe ill-posedness and high computational cost hinder efficient 3D reconstruction. In this paper, we propose a novel framework for efficient 3D internal velocity field reconstruction. First, we parameterize the velocity field with a coordinate-based neural network as an implicit continuous function, enabling compact representation from sparse observations. Second, we adopt ray-based traveltime tomography with multi-level parallel forward modeling to accelerate computation. Third, we leverage

*Equal Contribution.
†Corresponding Author

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

a differentiable Fresnel volumizing method that extends 1D ray paths into 3D volumetric regions (*i.e.*, Fresnel volumes), enabling spatially dense and multi-scale gradient diffusion, thus alleviating gradient sparsity and improving reconstruction accuracy. To further improve efficiency, we propose a 2D slice-based construction of 3D Fresnel volumes using a lightweight neural network. Extensive experiments on synthetic datasets, including ablation studies, demonstrate up to $10\times$ speedup over FWI while maintaining comparable accuracy, validating the effectiveness of the proposed framework for efficient 3D velocity field reconstruction problems.

## 1 Introduction

Wave velocity characterizes the propagation speed of mechanical waves in a medium. Velocity reconstruction serves as a fundamental approach for inferring the internal structure and composition of objects, with broad applications in nondestructive testing [1, 2], geological exploration [3, 4], and medical imaging [5, 6]. The general workflow generates multi-frequency wave excitations and records the resulting signals (*e.g.* , amplitudes, phases, and traveltimes) using receivers for observed data acquisition [7]. The internal velocity distribution is then reconstructed by iteratively minimizing the misfit between observed and synthetic data generated through physics-based forward modeling [8].

As representative inverse problems, velocity reconstruction and image-based 3D reconstruction share similar overall paradigms, while the former focuses on internal physical properties and the latter emphasizes visible appearance properties [9, 10, 11, 12]. As shown in Fig. 1, internal velocity reconstruction is more challenging for three main reasons: **(1)** Compared with 2D image-based appearance reconstruction, recovering 3D velocity distributions from 1D wave signals is significantly more ill-posed. **(2)** Forward modeling typically requires solving partial differential equations (PDEs) to simulate wave propagation [7, 13], which is substantially more complex than visible-light transport. **(3)** Wave signals are typically acquired through sparse sampling, further increasing the ill-posedness of the reconstruction process.

Recently, Neural Radiance Fields (NeRF) have greatly advanced image-based 3D reconstruction [14]. NeRF employs a multilayer perceptron (MLP) to parameterize a radiance field, providing an implicit and continuous scene representation [11]. However, due to the exclusive focus on appearance reconstruction and the demand for higher efficiency, implicit representations have gradually been replaced by hybrid or explicit representations [15, 16, 17, 18, 19]. In particular, 3D Gaussian Splatting (3DGS) [19] has significantly improved both reconstruction quality and rendering efficiency. However, we argue that the continuity and regularization properties of neural implicit representations remain highly valuable for internal reconstruction tasks [20, 21].

Widely regarded as a state-of-the-art method for velocity reconstruction, full waveform inversion (FWI) leverages full waveform information and thus achieves theoretically the highest resolution [7]. However, the strong nonlinearity of the governing equation, together with practical limitations such as noise contamination and sparse acquisition, makes high-accuracy convergence difficult [13, 22]. Moreover, repeated forward modeling incurs substantial computational and memory costs, leading to significant limitations for large-scale 3D reconstruction [23].

Our goal is to achieve an effective trade-off between **performance** and **efficiency** in internal properties 3D reconstruction. In this paper, we propose an efficient framework for large-scale velocity reconstruction, built upon four key components:

**(1)** A coordinate-based MLP is used to reparameterize the velocity field as an implicit function, enabling a compact and continuous representation. Compared with discrete grids, it is better suited for reconstructing internal structures from sparse observations by enforcing spatial continuity and implicit regularization, thereby improving stability.

**(2)** We adopt transmitted wave-based traveltime tomography (TTT) instead of FWI to reduce computational complexity, and develop a multi-level parallel ray-based forward modeling scheme. By exploiting both ray-level and hierarchical parallelism, this design enables efficient large-scale simulation and achieves substantial acceleration over wave-equation-based solvers.

**(3)** To alleviate gradient sparsity in TTT, we leverage a differentiable Fresnel volumizing approach that extends 1D ray paths into spatially distributed 3D regions (Fresnel volumes), leading a Fresnel-TTT method. This converts the sensitivity associated with sparse 1D ray paths into 3D volumetric

regions, enabling dense gradient propagation. Furthermore, it supports multi-scale gradient diffusion, yielding coarse-to-fine inversion behavior and improved accuracy.

**(4)** To further improve efficiency, we propose Fresnel Slice Net (FS-Net), a lightweight network that predicts local 2D Fresnel slices and aggregates them into 3D volumes. This slice-based strategy avoids the high computational of direct 3D volumetric construction while maintaining flexibility.

Experiments on representative synthetic benchmarks in computer graphics demonstrate that the proposed framework reduces computational time by up to **10**$\times$ compared to FWI-based methods while maintaining high-fidelity reconstructions **(SSIM up to 0.96, PSNR up to 26 dB)**. Especially under noisy conditions, our framework demonstrates superior robustness and achieves optimal overall performance. Furthermore, the proposed FS-Net exhibits effective generalization and flexibility. Overall, the results demonstrate an efficient and physically consistent alternative to FWI for large-scale 3D velocity field reconstruction problems.

## 2 Related works and motivation

**Appearance property reconstruction.** As illustrated in Fig. 1(a), appearance property reconstruction leverages 2D images to model 3D scenes, either implicitly or explicitly. Neural Radiance Fields (NeRF) utilizes multi-view images to construct a continuous function of color and volume density over spatial locations [14]. The success of NeRF in novel view synthesis has inspired numerous follow-up works aimed at improving its performance and expanding its applications[24, 25, 26, 27, 28]. Despite achieving high-quality results, NeRF remains computationally expensive in both training and inference. To address this limitation, recent methods have progressively shifted from purely implicit representations to hybrid implicit-explicit or fully explicit formulations [15, 18, 29]. This trend has led to the emergence of 3D Gaussian Splatting (3DGS) [19], which represents scenes using anisotropic Gaussian primitives and achieves high rendering efficiency and high-fidelity reconstruction.

**Internal velocity reconstruction.** Internal velocity reconstruction relies on sparse 1D signals acquired from the target surface to infer its internal properties, as shown in Fig. 1(b). Depending on the type of wave information utilized, existing methods can be broadly categorized into wave-theory-based approaches (*e.g.* , full waveform inversion, FWI) and ray-theory-based approaches (*e.g.* , traveltime tomography, TTT). FWI exploits full wavefield information and demonstrates strong capability in recovering fine-scale structures [4, 8, 30, 31]. However, this high resolution comes at the cost of reduced robustness and computational complexity, particularly in 3D settings. In contrast, TTT relies only on first-arrival traveltime information [32], making it more stable and computationally efficient for large-scale reconstruction [33, 34], but it typically lacks the ability to resolve fine details.

**Motivation.** Although implicit neural representations have become less appealing for appearance reconstruction, their continuity and regularization remain valuable for internal velocity reconstruction with sparse observations. Recent studies have attempted to integrate such representations with FWI to enhance inversion quality [13, 35, 36, 37]. However, the high computational cost of FWI fundamentally limits their applicability to large-scale 3D problems. A natural alternative is to integrate implicit representations with ray-theory-based methods; however, this is non-trivial due to the inherent gradient sparsity of ray-based modeling, the lack of multi-scale inversion capability, and the need for efficient large-scale forward simulation. In this work, we address these challenges in combining implicit neural representations with traveltime tomography (TTT), with the goal of achieving an effective balance between reconstruction accuracy and computational efficiency.

## 3 Preliminaries

TTT is grounded in ray theory under the high-frequency approximation, where wave propagation is modeled as rays [32]. It mainly consists of two processes: **forward modeling** and **inversion**.

**Forward modeling.** According to Fermat's principle, rays propagate in a medium along paths that minimize travel time. Thus, wave propagation in TTT can be described by the Eikonal equation:

$$\begin{cases} |\nabla T(\mathbf{x})| = \dfrac{1}{v(\mathbf{x})} = s(\mathbf{x}), & \forall \mathbf{x} \in \Omega \subset \mathbb{R}^3, \\ T(\mathbf{x}_s) = 0. \end{cases} \tag{1}$$

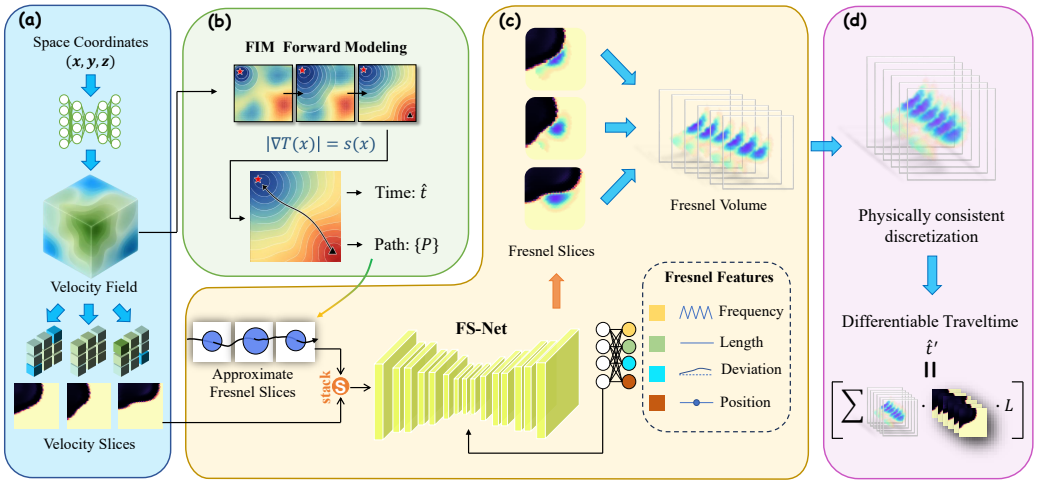


Figure 2: **Overview of proposed velocity reconstruction framework**. (a) Implicit Velocity representation. (b) Parallel FIM in forward modeling. (c) Slice-based Fresnel volume construction. (d) Gradient diffusion via differentiable Fresnel volumes.

where $\Omega$ represents the propagation domain, $\mathbf{x}$ is the spatial coordinate, $T(\mathbf{x})$ is the first-arrival traveltime from the source to each spatial point (*i.e.*, the traveltime field), $\nabla T(\mathbf{x})$ is the gradient of the traveltime field, $v(\mathbf{x})$ denotes the velocity field, and $s(\mathbf{x})$ denotes the slowness.

Common numerical solvers for Eq. 1 include the Fast Marching Method (FMM) [38], Fast Sweeping Method (FSM) [39], and Fast Iterative Method (FIM) [40]. These methods are generally based on the Godunov upwind scheme, which preserves monotonicity and causality during numerical updates. Under Fermat's principle, ray directions are aligned with the gradient of the traveltime field, as both are normal to the wavefronts [41]. Therefore, ray paths can be efficiently traced from receivers by following the steepest descent direction of the traveltime field, i.e., the negative gradient $-\nabla T(\mathbf{x})$.

**Inversion.** In the inversion process, gradient computation mainly depends on the Jacobian matrix. In a continuous medium, the traveltime of the $i$-th ray is defined as the line integral of slowness along the ray path $\Gamma$: $T_i = \int_\Gamma s(\mathbf{x})dl$. In discrete form, it can be approximated by a finite Riemann sum over grid cells: $T_i = \sum_j s_j L_{ij}$, where $L_{ij}$ is the ray length within the $j$-th cell and $s_j$ denotes the corresponding slowness. The TTT gradient can therefore be written as:

$$\frac{\partial \mathcal{L}}{\partial s_j} = \frac{\partial \mathcal{L}}{\partial T_i}\frac{\partial T_i}{\partial s_j} = \frac{\partial \mathcal{L}}{\partial T_i} L_{ij} \tag{2}$$

where $\mathcal{L}$ is the loss function. This formulation indicates that non-zero gradients exist only in cells intersected by rays, resulting in severe gradient sparsity and non-unique inversion solutions, particularly in 3D scenarios.

## 4 Method

Fig. 2 illustrates the proposed velocity reconstruction framework, which consists of four components. Considering their relative importance, we present the modules in reverse order, from (d) to (a). As the core innovations, **(d) Gradient diffusion via differentiable Fresnel volumes** and **(c) Slice-based Fresnel volume construction** are designed to improve performance and efficiency, respectively. To further support these objectives, we integrate **(b) Parallel FIM** and **(a) Implicit velocity representation**.

### 4.1 Gradient diffusion via differentiable Fresnel volumes

**Fresnel volume.** Eq. (2) reveals that TTT gradients are highly sparse, leading to non-unique inversion solutions. In contrast to ray theory, which relies on the infinite-frequency assumption, real

waves are inherently band-limited, meaning that their propagation is affected not only by structures along the ray path but also by those in its surrounding regions. Therefore, we introduce the Fresnel volume to extend 1D ray paths into 3D volumetric regions, which can be viewed as finite-width beam-like regions and offer a more physically consistent approach to extending the spatial influence of gradient. This process is also referred to as **3D Fresnel volumizing**, as illustrated in Fig. 3. We further refer to this Fresnel volume-enhanced traveltime tomography as **Fresnel-TTT**.

Formally, the Fresnel volume is defined as the set of rays whose traveltime delay relative to the shortest path is less than half a wave period. The 2D plane perpendicular to the ray path is referred to as a **Fresnel slice** (or Fresnel zone). In 3D discretized space, the Fresnel volume corresponding to a source–receiver $(S\text{-}R)$ pair is represented by the points $\mathbf{x}$ satisfying the following condition [42, 43, 44]:

$$| T_{S\mathbf{x}} + T_{\mathbf{x}R} - T_{SR} | \leq 1/2f \tag{3}$$

where $f$ is the frequency, $T_{SR}$ is the traveltime between the source $S$ and receiver $R$, and $T_{S\mathbf{x}}$ and $T_{\mathbf{x}R}$ are the traveltimes from $S$ to $\mathbf{x}$ and from $\mathbf{x}$ to $R$, respectively.

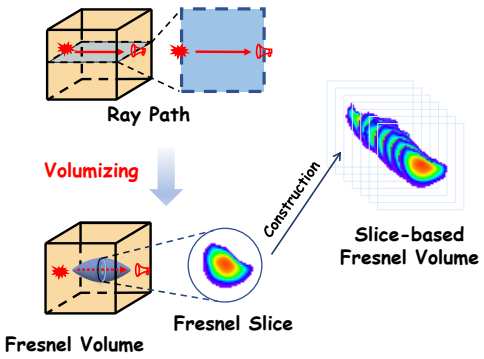


Figure 3: **Illustration of Fresnel Volumizing**. (a) Gradient diffusion from 1D rays to 3D volumes. (b) Frequency-dependent multi-scale effects. (c) Slice-based Fresnel volume construction.

The Fresnel volume can be computed using the paraxial approximation method [45] or traveltime field-based method [42]. The former approximates the Fresnel volume as an ellipsoid in a homogeneous medium [46], resulting in circular Fresnel slices. In contrast, traveltime field-based method [42] is more accurate and widely used. It evaluates forward and backward traveltime fields by placing sources at both the source and receiver, and computes the traveltime difference as $\Delta T(\mathbf{x}) = T_S(\mathbf{x}) + T_R(\mathbf{x}) - T_{SR}$. The effective Fresnel volume is then defined as the region satisfying $\Delta T(\mathbf{x}) \leq 1/2f$. Therefore, we adopt the traveltime field-based method for Fresnel volume computation and will further improve it in Sec. 4.2.

**Differentiable Fresnel volumizing method.** For Fresnel-TTT, the traveltime of $i\text{-}th$ ray is:

$$T_i = \int_{\Gamma} A(l)^{-1} \int_{\mathcal{F}} \frac{1}{v(\mathbf{x})} d\sigma dl \tag{4}$$

where $\mathcal{F}$ denotes the Fresnel slices (zones), $A(l)$ represents the area of the Fresnel slice at position $l$ along the ray, and $d\sigma$ denotes the area element. The traveltime under the Fresnel volume formulation can be interpreted as averaging the slowness over each cross-sectional area via surface integration, followed by a line integral along the ray path.

To enable gradient diffusion while ensuring correct gradient computation and physical consistency in the inversion process, and ultimately achieve a differentiable workflow, we derive the explicit gradient formulation for all cells within the Fresnel volume. We begin by discretizing Eq. 4 as:

$$T_i = \sum_j \sum_k \frac{a_k^{ij}}{A_{ij}} s_k^{ij} L_{ij} \tag{5}$$

where $a_k^{ij}$ denotes the $k\text{-}th$ cell that lies within the same Fresnel slice with area $A_{ij}$ as the $j\text{-}th$ cell intersected by the ray. To ensure physical consistency, we exploit the following insights: **(1) Grid cells closer to the ray (*i.e.*, with smaller traveltime differences) contribute more to the traveltime. (2) The traveltime computed based on the Fresnel volume should be equal to that computed based on rays. (3) The accumulated gradient of each Fresnel slice in Fresnel-TTT should be equal to the gradient of the corresponding cell in TTT.** This indicates that two key considerations are required: (1) assigning weights $w$ to each grid cell using a monotonically decreasing function of the traveltime difference, and (2) normalizing the effective weights within each Fresnel slice. Therefore, Eq. 4 can be discretized as:

$$T_i = \sum_j \sum_k \bar{w}_k^{ij} s_k^{ij} L_{ij}, \quad \text{where } \bar{w}_k^{ij} = \frac{w_k^{ij}}{\sum_k w_k^{ij}} \tag{6}$$

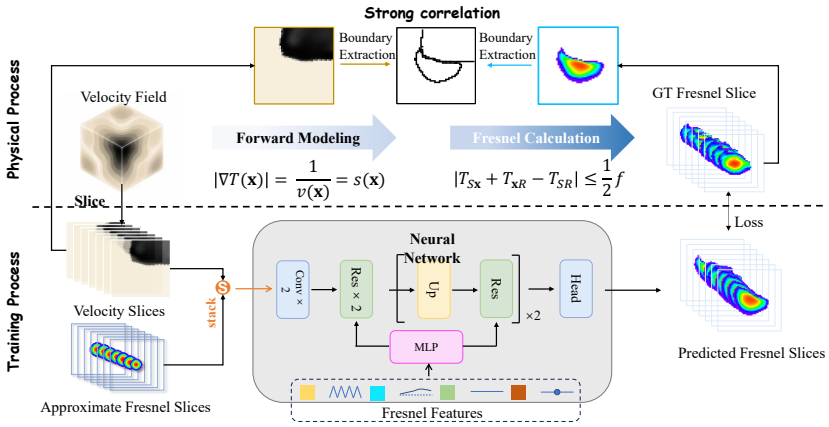


Figure 4: **Overview of slice-based Fresnel volume construction.** FS-Net maps approximate Fresnel slices from the ellipsoidal approximation Fresnel Slices, local velocity slices, and auxiliary features to refined 2D Fresnel slices, which are supervised by slices derived from the physical ray traveltime process and stacked to form the 3D Fresnel volume.

We refer to this formulation as the **physically consistent Fresnel formulation**, which constitutes one of the key contributions of this work. The formulation of $w$ is not fixed, and we define it as: $w = \cos^2(\frac{\pi}{2}\Delta T f)$.

Based on the above, we derive the explicit formulation of the diffused gradients:

$$\frac{\partial \mathcal{L}}{\partial s_k^{ij}} = \frac{\partial \mathcal{L}}{\partial T_i}\frac{\partial T_i}{\partial s_k^{ij}} = \frac{\partial \mathcal{L}}{\partial T_i}\bar{w}_k^{ij} L_{ij}, \quad \text{where } \bar{w}_k^{ij} = \frac{w_k^{ij}}{\sum_k w_k^{ij}} \tag{7}$$

Eq. 7 allows us to explicitly compute the gradient of the $k$-$th$ grid cell within the Fresnel slice associated with the $j$-$th$ cell traversed by the $i$-$th$ ray. Compared to Eq. 2, **we expand the gradients, which were originally restricted to units intersected by the ray, throughout the entire Fresnel volume region**. Furthermore, the derived explicit gradient formulation allows us to establish a differentiable workflow, while alleviating the memory overhead introduced by automatic differentiation.

Meanwhile, Eq. 3 shows that the extent of the Fresnel volume is controlled by frequency. This allows us to achieve multi-scale gradient diffusion by varying the frequency at different stages of optimization, which is crucial for improving inversion performance.

### 4.2 Slice-based Fresnel volume construction

As indicated in Sec. 4.1, accurate computation of the 3D Fresnel volume based on traveltime field [42] requires additional forward simulations, as it involves evaluating traveltime fields by imposing excitations at the source and receiver locations, respectively. In contrast, the paraxial approximation [45] provides a computationally efficient 3D ellipsoidal approximation, albeit with limited accuracy. To bridge the gap between efficiency and accuracy, we propose a neural network–based approach that learns the mapping from the 3D ellipsoidal approximation and velocity distribution to the full 3D Fresnel volume, enabling the recovery of fine structural details conditioned on the velocity field.

In implementation, we adopt a slice-based strategy to further improve efficiency. We introduce Fresnel Slice Net (FS-Net), a lightweight neural network that predicts 2D Fresnel slices $f_s$ from the corresponding 2D approximate Fresnel slices $\hat{f}_s$ derived from the ellipsoidal approximation and local velocity slices $v_s$. These predicted 2D slices are then stacked to construct the full 3D Fresnel volume, as illustrated in Fig. 3. For a point $\mathbf{x}$ in the ray path, both $\hat{f}_s(\mathbf{x})$ and $v_s(\mathbf{x})$ are extracted by slicing the 3D ellipsoid and the velocity model along the plane orthogonal to the ray path at $\mathbf{x}$. The approximate Fresnel slice $\hat{f}_s(\mathbf{x})$ is modeled as a circle with radius $r(\mathbf{x}) = \sqrt{d(L-d)/fLs(\mathbf{x})}$, where $d$ denotes the distance from point $\mathbf{x}$ to the source along the ray path $L$.

In addition, we incorporate four auxiliary features, collectively denoted as $\theta$, into the training process: the frequency $f$, the relative position $d(L-d)/L^2$, the total path length $L$, and the average cosine

similarity between the ray path and the source–receiver direction. The first three features characterize the local conditions at each slice location, thereby alleviating the non-uniqueness issue. The last feature reflects the degree of deviation of the current ray path, which is related to the deformation of the effective region in the Fresnel slice. Notably, ground truth Fresnel slices are obtained through the physical ray traveltime process, and FS-Net is designed to approximate this physical process. Consequently, FS-Net ($N$) predicts the refined 2D Fresnel slice $f_s$ at point $\mathbf{x}$ as

$$f_s(\mathbf{x}) = N(\hat{f}_s(\mathbf{x}), v_s(\mathbf{x}), \theta(\mathbf{x})) \tag{8}$$

The overall workflow is illustrated in Fig. 4. The provided inputs exhibit strong correlations with the target Fresnel slices. Fresnel slices are tightly aligned with the velocity interfaces in the velocity slices, and the approximate Fresnel slices provide a strong prior that largely determines the predicted region. Such a strong correlation allows FS-Net, with only **8,317** parameters, to learn an effective mapping and achieve efficient inference. Furthermore, the proposed Fresnel volume construction method depends only on local information, making it independent of the data acquisition setup. The local-information-driven mechanism of FS-Net provides strong flexibility and interpolation capability, suggesting the potential to generalize to unseen test scenarios.

### 4.3 Parallel FIM

Forward modeling requires solving the Eikonal equation, which is a major computational bottleneck in TTT. Common solvers include the FMM, FSM, FIM, which differ in global update strategies and thus exhibit different trade-offs in accuracy and parallel efficiency. FMM enforces strict wavefront causality via a global priority queue, achieving high accuracy but limited parallel efficiency [47]. FSM reduces global ordering through multi-directional sweeping and offers a balance between stability and efficiency, though residual dependencies still restrict scalability [48]. In contrast, FIM adopts a local active-set iteration scheme without global ordering, making it well-suited for parallel computation [49, 50].

In this work, we adopt FIM as the core forward modeling engine and develop a parallel implementation to fully exploit its scalability. We enable both intra-source and inter-source parallelism, allowing simultaneous traveltime computation from multiple sources on single- and multi-GPU platforms. This is significantly accelerates multi-source forward modeling while preserving the accuracy and stability of FIM.

### 4.4 Implicit velocity representation

Implicit neural representations provide a powerful and flexible solution for general data reconstruction tasks [14, 51, 52]. We consider reparameterizing the velocity field using implicit neural representations to be particularly important for internal velocity reconstruction. The primary reason lies in the continuity and spectral bias of neural implicit representations [53, 54], which significantly improve the robustness of inversion under sparse observation conditions and suppress non-physical velocity values. This motivates us to parameterize the velocity field using a coordinate-based MLP.

## 5 Experiments

### 5.1 Experimental settings

**Problem Setup and Dataset.** We consider the reconstruction of low-velocity targets embedded in a homogeneous high-velocity background, as this setting is commonly encountered in practice and presents greater challenges. Experiments are conducted on ShapeNetSem [55], ABC [56], and Thingi10k [57] datasets, where representative benchmarks are treated as reconstruction targets within a $128 \times 128 \times 128$ voxel space. The reconstruction target is assigned a velocity of $1000\ \mathrm{m/s}$, while the homogeneous background is set to $3000\mathrm{m/s}$. The acquisition system is configured with three through-transmission setups in the standard setting, each consisting of 9 sources and 9 receivers placed on opposite faces. In total, 243 travel-time measurements are collected for each model.

**Baseline, Metrics and Implementation Details.** The proposed framework is evaluated against several FWI-based methods, including total variation regularized FWI (TV-FWI), multi-scale FWI

Table 1: **Quantitative comparison on five representative benchmarks.** The ↑ indicates the larger value achieved, the better performance is, while ↓ indicates the smaller, the better. We mark out best and second best, and third best for metrics.

| | Duck | | | Gerotor | | | Seal | | | Geometry | | | Bunny | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *PSNR* ↑ | *SSIM* ↑ | *Time* ↓ | *PSNR* ↑ | *SSIM* ↑ | *Time* ↓ | *PSNR* ↑ | *SSIM* ↑ | *Time* ↓ | *PSNR* ↑ | *SSIM* ↑ | *Time* ↓ | *PSNR* ↑ | *SSIM* ↑ | *Time* ↓ | *Memory* ↓ |
| TV-FWI | 19.94 | 0.7480 | 37.29 | 19.82 | 0.7707 | 37.84 | 18.09 | 0.6801 | 27.97 | 25.72 | 0.8875 | 21.67 | 18.51 | 0.7978 | 37.80 | 75.7 |
| MS-FWI | 20.29 | 0.7471 | 50.56 | 21.54 | 0.8580 | 50.57 | 20.68 | 0.7093 | 33.88 | 23.27 | 0.8248 | 32.86 | 22.96 | 0.9051 | 40.05 | 75.7 |
| IFWI | 25.08 | 0.9244 | 105.53 | 30.15 | 0.9666 | 55.14 | 22.45 | 0.9262 | 79.21 | 32.04 | 0.9855 | 58.04 | 27.95 | 0.9573 | 84.43 | 91.6 |
| IG-FWI | 16.62 | 0.8829 | 77.70 | 28.17 | 0.9568 | 51.28 | 28.76 | 0.9524 | 77.74 | 18.01 | 0.7629 | 53.35 | 21.77 | 0.8348 | 49.99 | 89.4 |
| **Ours** | 25.16 | 0.9292 | 3.36 | 24.01 | 0.9253 | 2.89 | 25.97 | 0.9545 | 6.25 | 26.39 | 0.9559 | 5.60 | 25.81 | 0.9418 | 20.83 | 25.9 |

(MS-FWI), implicit FWI with SIREN activations [36] (IFWI), and IG-FWI [13], which combines a hybrid representation based on multi-resolution hash encoding. Peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM) are adopted to evaluate reconstruction quality. In addition, runtime (Min) and memory (GB) consumption are reported to assess computational efficiency. We adopt the standard SIREN architecture [51] to reparameterize the velocity field in our framework. For explicit reconstruction methods (TV-FWI and MS-FWI), high-quality initial models are provided, while implicit methods (IFWI, IG-FWI, and our framework) are initialized from a homogeneous background model. All experiments are conducted on two NVIDIA A100 GPUs.

### 5.2 Comparison results

**Discussion on quantitative results.** Table 1 reports the quantitative comparison between our framework and FWI-based methods under the standard setting. Our framework achieves comparable reconstruction quality while significantly reducing computational time and memory consumption, and even attains the best performance on several models. Importantly, it shows no significant performance degradation across all cases, indicating stronger robustness than FWI-based approaches. In particular, on the **"Duck"** model, our framework achieves the best performance while being more than **10**× faster than competing methods. We further observe variations in runtime across different tasks, mainly due to the increased number of optimization iterations required for fine-grained structures. Notably, unlike FWI-based methods that often stagnate at suboptimal solutions, our framework continues to improve with additional iterations, albeit with increased computational cost.

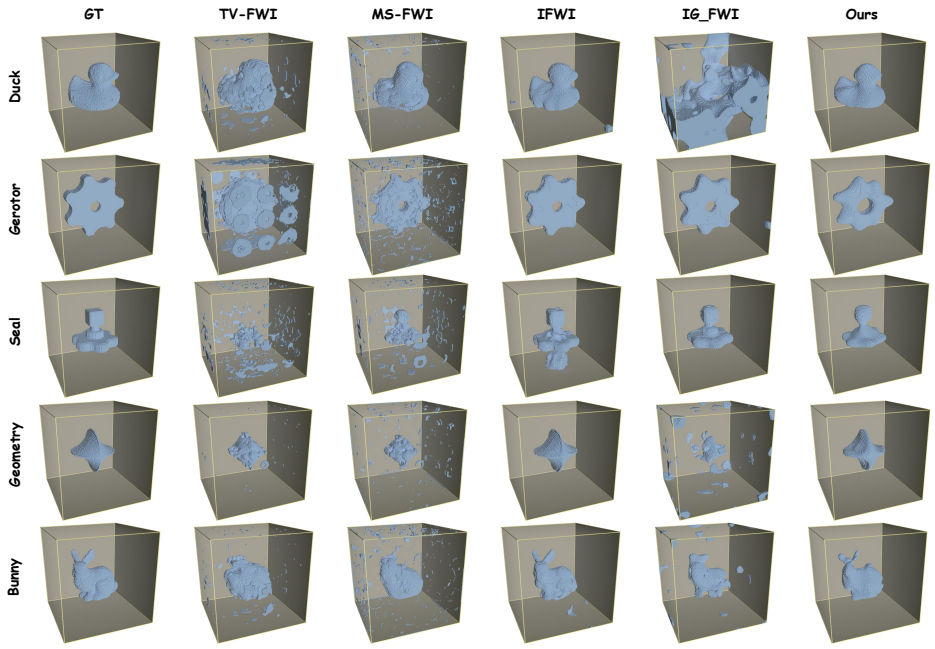


Figure 5: **Qualitative comparison on five representative benchmarks.** The visualization is primarily intended to highlight the structural accuracy of the reconstructed results.

**Discussion on qualitative results.** We provide qualitative comparisons in Fig. 5. Visually, the results of our framework contain almost no artifacts. By introducing Fresnel volumizing method, a multi-scale effect is naturally incorporated into the reconstruction process, effectively improving reconstruction accuracy. It should be acknowledged that, although our framework demonstrates stable performance, it still underperforms state-of-the-art FWI methods in capturing fine-grained details. This limitation is primarily determined by the inherent characteristics of ray-based methods.

### 5.3 Ablation study

To comprehensively evaluate the performance of our framework, we investigate the impact of key components, and assess its robustness under noisy scenarios.

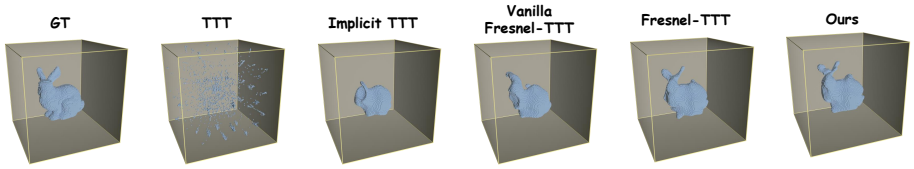


Figure 6: **Qualitative results of ablation on components of the proposed framework.**

Table 2: **Quantitative comparison of ablation on components.**

| | *PSNR* ↑ | *SSIM* ↑ | *Time* ↓ |
|---|---|---|---|
| TTT | - | - | - |
| Implicit TTT | 24.00 | 0.9315 | 16.08 |
| vanilla Fresnel-TTT | 24.64 | 0.9374 | 38.75 |
| Fresnel-TTT | 25.82 | 0.9447 | 40.92 |
| **Ours** | 25.81 | 0.9418 | 20.83 |

Table 3: **Quantitative comparison under noisy conditions.**

| | **10 dB** | | **5 dB** | | **0 dB** | |
|---|---|---|---|---|---|---|
| | *PSNR* ↑ | *SSIM* ↑ | *PSNR* ↑ | *SSIM* ↑ | *PSNR* ↑ | *SSIM* ↑ |
| TV-FWI | 20.59 | 0.7418 | 20.53 | 0.7372 | 20.43 | 0.7248 |
| MS-FWI | 21.85 | 0.7792 | 21.90 | 0.7682 | 21.71 | 0.7562 |
| IFWI | 22.93 | 0.8733 | 20.08 | 0.8255 | 19.29 | 0.7800 |
| IG-FWI | 19.52 | 0.8078 | 17.87 | 0.7758 | 16.06 | 0.7218 |
| **Ours** | 24.22 | 0.9302 | 23.84 | 0.9241 | 23.16 | 0.9196 |

**Component analysis.** We conduct comparisons among ray-based methods, including: (1) conventional TTT, (2) implicit TTT, (3) vanilla Fresnel-TTT [42], (4) Fresnel-TTT, and (5) our complete framework. This progressive comparison clearly validates the effectiveness of: (1) implicit neural representations, (2) gradient diffusion via Fresnel volume, (3) the physically consistent Fresnel formulation, and (4) slice-based Fresnel volume construction using FS-Net.

Table 2 and Fig. 6 present the quantitative and qualitative results of component ablation. As the method progresses from TTT to Fresnel-TTT through incremental module integration, the reconstruction performance is substantially improved, albeit at the cost of increased computational complexity. By introducing the slice-based Fresnel volume construction, we achieve a significant improvement in computational efficiency while maintaining reconstruction quality comparable to Fresnel-TTT.

**Noise robustness.** We use waveform data added with Gaussian noise at three SNRs as the observations for the FWI methods, and pick first-arrival traveltimes from noisy data using conventional cross-correlation as the observations for the TTT methods. As shown in Table 3, our framework consistently achieves the best average PSNR and SSIM values as the noise level increases, demonstrating superior noise robustness and reconstruction performance. This is mainly due to the fundamental difference in the mechanisms by which noise affects TTT and FWI methods.

## 6 Limitations

Our method relies on through-transmission acquisition geometries, which provide sufficient ray coverage for stable inversion; therefore, it is not directly applicable to reflection-based setups for internal structure reconstruction. In addition, the current study is primarily validated on synthetic datasets. In real-world scenarios, factors such as measurement noise, modeling errors, and more complex heterogeneous media may further affect reconstruction performance.

## 7 Conclusion

We present an efficient framework for 3D velocity reconstruction, achieving a better balance between reconstruction accuracy and computational efficiency than FWI-based methods. Specifically, we adopt a continuous implicit velocity representation and a parallel ray-based forward modeling scheme for efficient computation. To address gradient sparsity in ray-based inversion, we leverage differentiable Fresnel volumizing, which extends ray paths into 3D Fresnel volumes and enables dense, multi-scale gradient propagation. Furthermore, we design a slice-based Fresnel construction network (FS-Net) to efficiently approximate Fresnel volumes. Experiments demonstrate competitive reconstruction accuracy with significantly reduced computational cost, highlighting the effectiveness of combining implicit representations with ray-based inversion for large-scale 3D inverse problems.

## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (62472260), in part by the Natural Science Foundation of Shandong Province (ZR2025MS1015), and in part by the R&D Program of Jinan City, Shandong, China (202428002).

## References

[1] Robert Seidl. *Full waveform inversion for ultrasonic nondestructive testing*. PhD thesis, Technische Universität München, 2018.

[2] Luan T Nguyen and Ryan T Modrak. Ultrasonic wavefield inversion and migration in complex heterogeneous structures: 2D numerical imaging and nondestructive testing experiments. *Ultrasonics*, 82:357–370, 2018.

[3] Peng Jin, Xitong Zhang, Yinpeng Chen, Sharon Xiaolei Huang, Zicheng Liu, and Youzuo Lin. Unsupervised learning of full-waveform inversion: Connecting CNN and partial differential equation in a loop. *arXiv preprint arXiv:2110.07584*, 2021.

[4] Chengyuan Deng, Shihang Feng, Hanchen Wang, Xitong Zhang, Peng Jin, Yinan Feng, Qili Zeng, Yinpeng Chen, and Youzuo Lin. OpenFWI: Large-scale multi-structural benchmark datasets for full waveform inversion. *Advances in Neural Information Processing Systems*, 35:6007–6020, 2022.

[5] Lluís Guasch, Oscar Calderón Agudo, Meng-Xing Tang, Parashkev Nachev, and Michael Warner. Full-waveform inversion imaging of the human brain. *NPJ digital medicine*, 3(1):28, 2020.

[6] Jiahao Ren, Jian Li, Chang Liu, Shili Chen, Lin Liang, and Yang Liu. Deep learning with physics-embedded neural network for full waveform ultrasonic brain imaging. *IEEE transactions on medical imaging*, 43(6):2332–2346, 2024.

[7] Jean Virieux and Stéphane Operto. An overview of full-waveform inversion in exploration geophysics. *Geophysics*, 74(6):WCC1–WCC26, 2009.

[8] Albert Tarantola. Inversion of seismic reflection data in the acoustic approximation. *Geophysics*, 49(8):1259–1266, 1984.

[9] Ziyi Yin, Rafael Orozco, Philip Witte, Mathias Louboutin, Gabrio Rizzuti, and Felix J Herrmann. Extended source imaging—a unifying framework for seismic and medical imaging. In *SEG International Exposition and Annual Meeting*, page D041S078R003. SEG, 2020.

[10] Youzuo Lin, Shihang Feng, James Theiler, Yinpeng Chen, Umberto Villa, Jing Rao, John Greenhall, Cristian Pantea, Mark A Anastasio, and Brendt Wohlberg. Survey of deep learning and physics-based approaches in computational wave imaging. *Proceedings of the IEEE*, 2026.

[11] Kyle Gao, Yina Gao, Hongjie He, Dening Lu, Linlin Xu, and Jonathan Li. Neural radiance fields in 3D vision: A comprehensive review. *Computational Visual Media*, 2026.

[12] Zhangjin Huang, Yuxin Wen, Zihao Wang, Jinjuan Ren, and Kui Jia. Surface reconstruction from point clouds: A survey and a benchmark. *IEEE transactions on pattern analysis and machine intelligence*, 46(12):9727–9748, 2024.

[13] Ruihua Chen, Yisi Luo, Bangyu Wu, and Deyu Meng. Unveiling the mechanism of continuous representation full-waveform inversion: A wave based neural tangent kernel framework. *arXiv preprint arXiv:2603.22362*, 2026.

[14] 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.

[15] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. TensoRF: Tensorial radiance fields. In *European conference on computer vision*, pages 333–350. Springer, 2022.

[16] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5459–5469, 2022.

[17] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5501–5510, 2022.

[18] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022.

[19] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, et al. 3D Gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

[20] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8628–8638, 2021.

[21] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9446–9454, 2018.

[22] Yinan Feng, Yinpeng Chen, Shihang Feng, Peng Jin, Zicheng Liu, and Youzuo Lin. An intriguing property of geophysics inversion. In *International Conference on Machine Learning*, pages 6434–6446. PMLR, 2022.

[23] Qili Zeng, Shihang Feng, Brendt Wohlberg, and Youzuo Lin. InversionNet3D: Efficient and scalable learning for 3-D full-waveform inversion. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–16, 2021.

[24] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5855–5864, 2021.

[25] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5470–5479, 2022.

[26] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7210–7219, 2021.

[27] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. MVSNeRF: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14124–14133, 2021.

[28] Yuanhao Cai, Jiahao Wang, Alan Yuille, Zongwei Zhou, and Angtian Wang. Structure-aware sparse-view X-ray 3D reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11174–11183, 2024.

[29] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5752–5761, 2021.

[30] Zhigang Zhang, Zedong Wu, Zhiyuan Wei, Jiawei Mei, Rongxin Huang, and Ping Wang. Fwi imaging: Full-wavefield imaging through full-waveform inversion. In *SEG International Exposition and Annual Meeting*, page D031S027R004. SEG, 2020.

[31] Jonathan Fincke, Xiang Zhang, Bonghun Shin, Gregory Ely, and Brian W Anthony. Quantitative sound speed imaging of cortical bone and soft tissue: Results from observational data sets. *IEEE Transactions on Medical Imaging*, 41(3):502–514, 2021.

[32] Cédric Taillandier, Mark Noble, Hervé Chauris, and Henri Calandra. First-arrival traveltime tomography based on the adjoint-state method. *Geophysics*, 74(6):WCB1–WCB10, 2009.

[33] R Montelli, G Nolet, G Masters, FA Dahlen, and S-H Hung. Global P and PP traveltime tomography: rays versus waves. *Geophysical Journal International*, 158(2):637–654, 2004.

[34] Jie Zhang and M Nafi Toksoz. Nonlinear refraction traveltime tomography. *Geophysics*, 63(5):1726–1737, 1998.

[35] Zesong Wang, Weicheng Yan, Zhaohui Liu, Ming Yuchi, and Wu Qiu. $P^2$INR-FWI: An implicit neural representation method for speed of sound image reconstruction in ultrasound computed tomography. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 420–430. Springer, 2025.

[36] Jian Sun, Kristopher Innanen, Tianze Zhang, and Daniel Trad. Implicit seismic full waveform inversion with deep neural representation. *Journal of Geophysical Research: Solid Earth*, 128(3):e2022JB025964, 2023.

[37] Bo Du, Jian Sun, Anqi Jia, Ning Wang, and Huaishan Liu. Physics-informed robust and implicit full waveform inversion without prior and low-frequency information. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–12, 2024.

[38] James A Sethian. Fast marching methods. *SIAM review*, 41(2):199–235, 1999.

[39] Hongkai Zhao. A fast sweeping method for Eikonal equations. *Mathematics of computation*, 74(250):603–627, 2005.

[40] Won-Ki Jeong and Ross T Whitaker. A fast iterative method for Eikonal equations. *SIAM Journal on Scientific Computing*, 30(5):2512–2534, 2008.

[41] James G Berryman. Fermat's principle and nonlinear traveltime tomography. *Physical review letters*, 62(25):2953, 1989.

[42] Don W Vasco, John E Peterson, and Ernest L Majer. Beyond ray tomography; wavepaths and Fresnel volumes. *Geophysics*, 60(6):1790–1804, 1995.

[43] Toshiki Watanabe, Toshifumi Matsuoka, and Yuzuru Ashida. Seismic traveltime tomography using Fresnel volume approach. In *SEG Technical Program Expanded Abstracts*, volume 18, pages 1402–1405. Society of Exploration Geophysicists, 1999.

[44] Jianzhong Zhang, Bo Zhao, and Hua-wei Zhou. Fat ray tomography with optimal relaxation factor. In *SEG International Exposition and Annual Meeting*, pages SEG–2009. SEG, 2009.

[45] Vlastislav Cerveny and Jose Eduardo P Soares. Fresnel volume ray tracing. *Geophysics*, 57(7):902–915, 1992.

[46] Jesper Spetzler and Roel Snieder. The Fresnel volume and transmitted waves. *Geophysics*, 69(3):653–663, 2004.

[47] James A Sethian and A Mihai Popovici. 3-D traveltime computation using the fast marching method. *Geophysics*, 64(2):516–523, 1999.

[48] Jianliang Qian, Yong-Tao Zhang, and Hong-Kai Zhao. Fast sweeping methods for Eikonal equations on triangular meshes. *SIAM Journal on Numerical Analysis*, 45(1):83–107, 2007.

[49] Wei Cai, Peimin Zhu, and Guangchao Li. Improved fast iterative method for higher calculation accuracy of traveltimes. *Computers & Geosciences*, 174:105331, 2023.

[50] Jingqi Zhang, Zihao Zhou, Lixin Ren, Junyuan Liu, Ying Li, and Xiaowei He. A parallel multiscale FIM approach in solving the Eikonal equation on GPU. *Computer-Aided Design*, page 103949, 2025.

[51] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020.

[52] Vishwanath Saragadam, Daniel LeJeune, Jasper Tan, Guha Balakrishnan, Ashok Veeraraghavan, and Richard G Baraniuk. WIRE: Wavelet implicit neural representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18507–18516, 2023.

[53] Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yanyang Xiao, and Zheng Ma. Frequency principle: Fourier analysis sheds light on deep neural networks. *arXiv preprint arXiv:1901.06523*, 2019.

[54] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International conference on machine learning*, pages 5301–5310. PMLR, 2019.

[55] Manolis Savva, Angel X Chang, and Pat Hanrahan. Semantically-enriched 3D models for common-sense knowledge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 24–31, 2015.

[56] Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. ABC: A big cad model dataset for geometric deep learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9601–9611, 2019.

[57] Qingnan Zhou and Alec Jacobson. Thingi10K: A dataset of 10,000 3D-printing models. *arXiv preprint arXiv:1605.04797*, 2016.

[58] Alan Richardson. Seismic full-waveform inversion using deep learning tools and techniques. *arXiv preprint arXiv:1801.07232*, 2018.

# Supplementary Material

## A Theoretical Background

### A.1 Full waveform inversion

The forward wavefield propagation in full waveform inversion (FWI) is governed by the constant-density acoustic wave equation. Let $\Omega$ and $T$ denote the spatial and temporal coordinates, respectively, the forward modeling process can be formulated as:

$$\begin{cases} \dfrac{1}{v^2(\mathbf{x})}\dfrac{\partial^2 u(\mathbf{x},t)}{\partial t^2} = \nabla^2 u(\mathbf{x},t) + s(\mathbf{x}_s,t), & \forall \mathbf{x} \in \Omega \subset \mathbb{R}^3, \forall t \in T, \\ u(\mathbf{x},0) = 0, \quad \dfrac{\partial u(\mathbf{x},0)}{\partial t} = 0, & \forall \mathbf{x} \in \Omega \subset \mathbb{R}^3, \forall t \in T. \end{cases} \tag{9}$$

where $v(\mathbf{x})$ represents the velocity field, $u(\mathbf{x},t)$ denotes the wavefield, $\nabla^2$ is the Laplacian operator, $\mathbf{x}_s = 1, \ldots, N_s$ and $s(\mathbf{x_s})$ represent the source position and source function, respectively. In practice, the wave equation is commonly solved numerically with finite-difference schemes on a discretized computational domain. The synthetic data are obtained by sampling the forward wavefield at the receiver locations, and can be written as:

$$d_{\text{syn}}(\mathbf{x}_r,t) = \int_\Omega u(\mathbf{x},t)\delta(\mathbf{x}-\mathbf{x}_r)d\mathbf{x} \tag{10}$$

where $\mathbf{x}_r = 1, \ldots, N_r$ is the receiver location, and $\delta$ denotes the Dirac delta function for spatial sampling. FWI reconstructs the velocity distribution by iteratively minimizing the misfit between observed and synthetic data. The loss function of FWI is expressed in least-squares form as:

$$\mathcal{L}(v) = \frac{1}{2}\sum_{N_s}\sum_{N_r}\int_0^T \parallel d_{syn}(\mathbf{x}_r,t) - d_{obs}(\mathbf{x}_r,t) \parallel_2^2 dt. \tag{11}$$

where $d_{obs}$ denotes the observed data. The gradient is expressed as the zero-lag cross-correlation between the forward wavefield and the back-propagated residual wavefield (i.e., the adjoint wavefield), and can be computed using the adjoint-state method as follows:

$$\nabla\mathcal{L}(v) = \sum_{N_s}\int_0^T \frac{\partial^2 u(\mathbf{x},t)}{\partial t^2}\cdot q(\mathbf{x},t)dt \tag{12}$$

The adjoint wavefield $q(\mathbf{x},t)$ is obtained by injecting the data residuals $(d_{syn} - d_{obs})$ as source terms at the receiver locations and propagating them backward in time. When the governing equation or boundary conditions change, the adjoint equation and the form of the gradient must be manually derived. Therefore, a more efficient strategy is to compute the gradient using automatic differentiation, which is mathematically equivalent to the adjoint-state method. Let $v^{[k]}$ denote the velocity field at the $k$-th iteration and $\alpha$ be the learning rate. The gradient-descent-based optimization process can then be expressed as:

$$v^{[k+1]} = v^{[k]} - \alpha\cdot\nabla\mathcal{L}(v^{[k]}) \tag{13}$$

By discretizing the above formulation, we define the wavefield $u$ and the velocity field $v \in \mathbb{R}^{N_x\times N_y\times N_z}$, time steps $t \in [0, N_t]$. Considering the **computational time cost** and **computational memory cost** under the worst-case single-source scenario:

- **Computational time cost**: Each FWI iteration involves a forward and a backward wavefield propagation across all time steps, and each propagation requires evaluating the wavefield over the full computational domain. The **computational time cost** can be expressed as $\mathcal{O}(2 \times N_t \times N_x \times N_y \times N_z)$
- **Computational memory cost**: Since FWI gradient computation requires storing the entire forward-propagated wavefield, the **computational memory cost** can be expressed as $\mathcal{O}(N_t \times N_x \times N_y \times N_z)$.

### A.2 Traveltime tomography

Traveltime tomography (TTT) models wave propagation under the high-frequency ray approximation, with the propagation process characterized by the Eikonal equation:

$$\begin{cases} |\nabla T(\mathbf{x})| = \dfrac{1}{v(\mathbf{x})} = s(\mathbf{x}), & \forall \mathbf{x} \in \Omega \subset \mathbb{R}^3, \\ T(\mathbf{x}_s) = 0. \end{cases} \tag{14}$$

Numerical discretization of the Eikonal equation is commonly based on the Godunov upwind finite-difference scheme. We consider a 3D regular Cartesian grid with spacing $h$, where each point is given by $\mathbf{x} = (ih, jh, kh), (i, j, k \in \mathbb{N})$. Eq. (14) can be discretized as:

$$[\frac{(T_{i,j,k} - T^{xmin}_{i,j,k})^+}{h}]^2 + [\frac{(T_{i,j,k} - T^{ymin}_{i,j,k})^+}{h}]^2 + [\frac{(T_{i,j,k} - T^{zmin}_{i,j,k})^+}{h}]^2 = \frac{1}{v^2_{i,j,k}} \tag{15}$$

where $T_{i,j,k}$ represents the discretized approximation of traveltime at point $(ih, jh, kh)$ in $T(\mathbf{x})$, and $(\xi)^+ = max(\xi, 0)$. $T^{xmin}_{i,j,k}$, $T^{ymin}_{i,j,k}$, and $T^{zmin}_{i,j,k}$ are defined as:

$$\begin{cases} T^{xmin}_{i,j,k} = min(T_{i-1,j,k}, T_{i+1,j,k}) = a_1 \\ T^{ymin}_{i,j,k} = min(T_{i,j-1,k}, T_{i,j+1,k}) = a_2 \\ T^{zmin}_{i,j,k} = min(T_{i,j,k-1}, T_{i,j,k+1}) = a_3 \end{cases} \tag{16}$$

The local solver can be formulated as:

$$T_{i,j,k} = \begin{cases} \frac{1}{6}[2\sum_{i=1}^3 a_i + \sqrt{4(\sum_{i=1}^3 a_i)^2 - 12(\sum_{i=1}^3 a_i^2 - \frac{h^2}{v^2_{i,j,k}})}], & |a_1 - a_3| < h, \\ \frac{1}{2}[a_1 + a_2 + \sqrt{2\frac{h^2}{v^2_{i,j,k}} - (a_1 - a_2)^2}], & |a_1 - a_2| < h, \\ a_1 + \frac{h}{v_{i,j,k}}, & otherwise. \end{cases} \tag{17}$$

After computing the traveltime field associated with the source, the ray path from the receiver to the source can be traced using the steepest traveltime-gradient method. The ray tracing direction can be expressed as:

$$\mathbf{d}(\mathbf{x}) = -\frac{\nabla T(\mathbf{x})}{|\nabla T(\mathbf{x})|}, \quad \nabla T(\mathbf{x}) = \left[\frac{\partial T}{\partial x}, \frac{\partial T}{\partial y}, \frac{\partial T}{\partial z}\right]^{\mathrm{T}} \tag{18}$$

According to the Radon transform, the synthetic traveltime from the source $\mathbf{x}_s$ to the receiver $\mathbf{x}_s$ can be expressed as:

$$T_{syn}(\mathbf{x}_s, \mathbf{x}_r) = \int_{L_{(\mathbf{x}_s, \mathbf{x}_r)}} s(\mathbf{x}) dl \tag{19}$$

For the $i$-th ray, the discretized form of the traveltime computation can be expressed as:

$$T_i = \sum_j s_j L_{ij} \tag{20}$$

where $L_{ij}$ is the ray length within the $j$-th cell and $s_j$ denotes the corresponding slowness. In discrete form, we define the slowness vector as $\mathbf{s} \in \mathbb{R}^{N_\mathfrak{s} \times 1}$ and the traveltime vector as $\mathbf{t} \in \mathbb{R}^{N_\mathfrak{t} \times 1}$, where $N_\mathfrak{s}$ and $N_\mathfrak{t}$ denote the number of slowness cells and traveltime measurements (i.e., rays), respectively.. TTT reconstructs the velocity (slowness) field by minimizing the misfit between observed traveltimes and synthetic traveltimes. In this work, the loss function is defined as:

$$\mathcal{L}(\mathbf{s}) = \sum_{i=1}^{N_\mathfrak{t}} |T^i_{syn} - T^i_{obs}| = \|\Delta \mathbf{t}\|_1 \tag{21}$$

where $\Delta \mathbf{t} \in \mathbb{R}^{N_t \times 1}$ denotes the traveltime residual vector. The gradient can be written as:

$$\nabla \mathcal{L}(\mathbf{s}) = \frac{\partial \mathcal{L}}{\partial \mathbf{t}} \frac{\partial \mathbf{t}}{\partial \mathbf{s}} = \mathbf{J}^\top \operatorname{sign}(\Delta \mathbf{t}), \quad \text{where} \operatorname{sign}(x) = \begin{cases} 1, & x > 0, \\ 0, & x = 0, \\ -1, & x < 0. \end{cases} \tag{22}$$

where $\mathbf{J} \in \mathbb{R}^{N_t \times N_s}$ denotes the Jacobian matrix, which characterizes the sensitivity of traveltimes to changes in slowness. For the $j$-th slowness cell, its gradient can be expressed as:

$$\nabla \mathcal{L}(s_j) = \sum_{i=1}^{N_t} J_{ij} \cdot \operatorname{sign}(\Delta T_i), \quad \text{where } J_{ij} = \begin{cases} L_{ij}, & \Gamma_i \cap \Omega_{s_j} \neq \emptyset, \\ 0, & otherwise. \end{cases} \tag{23}$$

Considering the **computational time cost** and **computational memory cost** under the worst-case single-source scenario:

- **Computational time cost**: The primary computational cost of TTT arises from solving the Eikonal equation. The **computational time cost** can be expressed as $\mathcal{O}(N_x \times N_y \times N_z)$
- **Computational memory cost**: The gradient descent optimization entails storing the Jacobian matrix. The **computational memory cost** can be expressed as $\mathcal{O}(N_r \times N_x \times N_y \times N_z)$.

### A.3 Fresnel volume-enhanced traveltime tomography

#### A.3.1 Fresnel volume

Compared with traditional TTT, the Fresnel volumizing method extends wave propagation from 1D ray paths to 3D volumetric regions, effectively mitigating the reconstruction non-uniqueness caused by gradient sparsity. This finite volume is referred to as the Fresnel volume. There are two methods for determining the effective region of the Fresnel volume: (1) The exact computation method and (2) The paraxial approximation method.

**Exact computation method.** In the exact computation method, the effective region of the Fresnel volume is determined according to its definition given in Eq. 3. This process requires separately calculating the traveltimes ($T_{S\mathbf{x}}$ and $T_{R\mathbf{x}}$) from the source and the receiver to all grid cells. Although this method provides a rigorously defined Fresnel volume, it requires additional forward modeling computations equal to the number of receivers. In TTT, the dominant computational burden comes from forward modeling, making the exact computation method computationally inefficient.

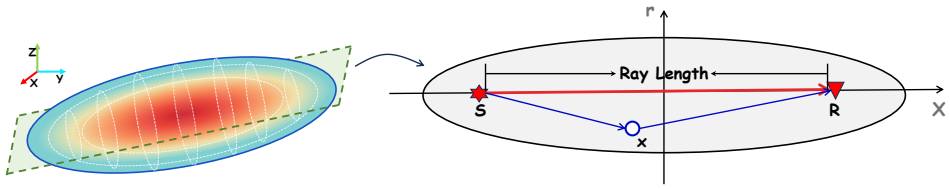


Figure 7: 2D cross-section of the paraxially approximated Fresnel volume along its major axis.

**Paraxial approximation method.** In the paraxial approximation method, the Fresnel volume is commonly modeled as an ellipsoid in a homogeneous medium. The ray path is regarded as the major axis of the ellipsoid, and the Fresnel slice perpendicular to this axis is a standard circle. The 2D profile along the major axis of the 3D Fresnel volume is an ellipse, as shown in Fig. 7. Let $a$ denote the semi-major axis and $b$ represent the semi-minor axis of the ellipse. The standard equation of the ellipse is written as:

$$\frac{x^2}{a^2} + \frac{r^2}{b^2} = 1 \tag{24}$$

where $r$ is the radius of the Fresnel slice. By combining Eq. 3 with the homogeneous velocity field $v(\mathbf{x})$, the boundary point $\mathbf{x}_e$ of the Fresnel volume satisfies:

$$L_{S\mathbf{x}_e} + L_{\mathbf{x}_e R} - L_{SR} = \frac{\lambda}{2} \tag{25}$$

where $L_{SR} = T_{SR} \times v = L$, and $\lambda$ denotes the wavelength. From the geometric relationship, it follows that:

$$a = \frac{L}{2} + \frac{\lambda}{4}, \qquad b = \sqrt{\frac{L\lambda}{4} + \frac{\lambda^2}{16}} \tag{26}$$

When $\lambda \ll L$, Eq. 26 can be simplified as:

$$a \approx \frac{L}{2}, \qquad b \approx \sqrt{\frac{L\lambda}{4}} \tag{27}$$

Therefore, the radius of the paraxially approximated Fresnel slice is given by:

$$r(\mathbf{x}) \approx \sqrt{x\lambda(1 - \frac{x}{L})} = \sqrt{\frac{d(L-d)}{fLs(\mathbf{x})}} \quad (0 \leq d \leq L) \tag{28}$$

where $d$ denotes the distance from point $\mathbf{x}$ to the source or receiver along the ray path $L$. Clearly, the paraxial approximation method is more computationally efficient but less accurate.

The above two methods jointly motivate us to propose FS-Net for slice-based Fresnel volume construction. Eq. 28 clarifies why the FS-Net architecture is designed in this manner. Specifically, the radius of the approximated Fresnel slice is determined by three factors: (1) frequency $f$, (2) path length $L$, and (3) the relative position of the slice $d(L-d)/L^2$. Therefore, without including these factors as auxiliary features in the training process, the approximate Fresnel slice suffers from non-uniqueness.

Obviously, the paraxial approximation method cannot accurately represent the actual situation. Therefore, the exact computation method based on the traveltime field is widely adopted. Considering the **computational time cost** and **computational memory cost** under the worst-case single-source scenario:

- **Computational time cost**: Computing accurate Fresnel volumes requires applying excitations at the source and receiver locations separately to perform forward modeling. The **computational time cost** can be expressed as $\mathcal{O}((N_r + 1) \times N_x \times N_y \times N_z)$
- **Computational memory cost**: Besides the Jacobian matrix, Fresnel-TTT also requires additional storage for traveltime fields. The **computational memory cost** can be expressed as $\mathcal{O}(2 \times N_t \times N_x \times N_y \times N_z)$.

**Summary.** FWI can achieve the best inversion performance but its prohibitive computational cost and extreme instability significantly limit its application in 3D velocity field reconstruction. In contrast, conventional TTT offers high computational efficiency and improved stability, but its applicability is still constrained by the severe non-uniqueness induced by sparse gradient information. For the Fresnel-TTT adopted in this work, it effectively alleviates the sparse gradient issue and improves inversion performance with an acceptable increase in computational cost.

# B Details of Proposed Framework

## B.1 Physically consistent gradient diffusion

**The corresponding component in the proposed framework is primarily designed to improve reconstruction accuracy.** Specifically, based on the simultaneous iterative reconstruction technique (SIRT), previous work [43, 44] has derived the gradient formulation of vanilla Fresnel-TTT:

$$\nabla \mathcal{L}(s_j) = \frac{1}{M} \sum_{i=1}^{M} \Delta T_i \bar{w}_k^{ij}, \quad \text{where } \bar{w}_k^{ij} = \frac{w_k^{ij}}{\sum_j \sum_k w_k^{ij}} \tag{29}$$

where $M$ represents the number of Fresnel volumes that include cell $s_j$. For each grid cell within the Fresnel volume, the above formulation introduces averaging effects: (1) It treats the ray path lengths $L_{ij}$ within individual cells as identical. (2) It normalizes each effective Fresnel weight within the entire Fresnel volume, thereby treating all Fresnel volumes as having equal importance.

(3) It computes the average gradient of cell $s_j$ within one iteration. This formulation reduces the influence of anomalous gradients, thereby improving the stability of the inversion. However, this averaging process also introduces deviations from the true physical gradient, resulting in inaccurate reconstruction of detailed structures.

In contrast, we derive a physically consistent gradient computation formulation:

$$\nabla \mathcal{L}(s_j) = \sum_{i=1}^{M} \operatorname{sign}(\Delta T_i) \bar{w}_k^{ij} L_{ij}, \quad \text{where } \bar{w}_k^{ij} = \frac{w_k^{ij}}{\sum_k w_k^{ij}} \tag{30}$$

This physically consistent gradient diffusion increases gradient density while ensuring the accuracy of both forward traveltime computation and backward gradient calculation.

### B.2 Slice-based Fresnel Volume construction

**The corresponding component in the proposed framework is primarily designed to improve reconstruction efficiency.** Although the Eikonal equation can be solved efficiently, the computation of Fresnel volumes requires extra forward modeling, which represents the dominant computational cost during reconstruction. To further improve efficiency, we adopt a slice-based Fresnel volume construction method using FS-Net to approximate the physical process of accurately computing Fresnel volumes.

On the one hand, we stack the paraxially approximated Fresnel slices and local velocity slices, both closely correlated with the accurate Fresnel slices, as inputs to the network. In addition, we incorporate auxiliary features $\theta(\mathbf{x})$ into the training process by performing affine transformations on the feature maps, where the affine transformation coefficients are generated by the MLP. The feature modulation process can be expressed as:

$$[\boldsymbol{\gamma}_i, \boldsymbol{\beta}_i] \xleftarrow{\text{split}} \mathbf{h}_i \leftarrow f_{\text{MLP}}(\theta(\mathbf{x})), \qquad \theta(\mathbf{x}) \in \mathbb{R}^4, \mathbf{h}_i \in \mathbb{R}^{2C}, \boldsymbol{\gamma}_i, \boldsymbol{\beta}_i \in \mathbb{R}^C \tag{31}$$

$$\mathbf{F}'_{i,c} = \gamma_{i,c} \mathbf{F}_{i,c} + \beta_{i,c} \tag{32}$$

where $\mathbf{F}_{i,c}$ denotes the $c$-th feature map in the $i$-th output, $\mathbf{h}$ represents the modulation coefficient generated by the MLP, with $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ acting on the weights and biases, and $C$ is the total number of output channels.

Furthermore, leveraging our understanding of 3D velocity field reconstruction, the proposed framework incorporates implicit neural representations and multi-level parallel FIM to enhance reconstruction performance and efficiency, respectively.

## C Experimental Details

### C.1 Dataset and standard settings

In our experiments, we choose representative benchmarks widely used in Computer Graphics from the ShapeNetSem, ABC, and Thingi10K datasets as reconstruction targets. **(1) The ShapeNetSem dataset** [55] is a large-scale 3D object dataset that consists of common indoor objects. **(2) The ABC dataset** [56] contains a large-scale collection of 3D CAD models, covering diverse object categories and geometric structures. (3) **The Thingi10K dataset** [57] contains thousands of real-world models collected from online 3D repositories. It includes diverse object categories with various geometric structures, complex topologies, and irregular mesh characteristics.

We perform internal velocity reconstruction and evaluation on a $128 \times 128 \times 128$ 3D voxel grid with the spatial interval of $1\ \mathrm{m}$. The 3D objects selected from the dataset are positioned at the center of the voxel grid and serve as the primary reconstruction targets. Compared with cases where the target object has a higher wave velocity than the background, the reconstruction of low-velocity targets is more prevalent in real-world applications. Typical examples include damage detection in non-destructive testing, as well as faults and cavities in geological exploration. So the background velocity is set to $3000\ \mathrm{m/s}$, while the reconstruction target is assigned a velocity of $1000\ \mathrm{m/s}$.

In this study, the wavefield responses of transmitted waves are primarily utilized as the basis for internal wave velocity reconstruction. As shown in Fig. 8, the acquisition system consists of three

identical direct-transmission configurations, each containing 9 transmitters and 9 receivers uniformly distributed on opposite sides of the reconstruction domain with a spacing of 30 grid cells. For each cross-transmission configuration, a single source is excited, and nine receivers simultaneously acquire the corresponding wavefield responses. In our framework, there are $3 \times 9 \times 9 = 243$ available traveltime measurements in total.

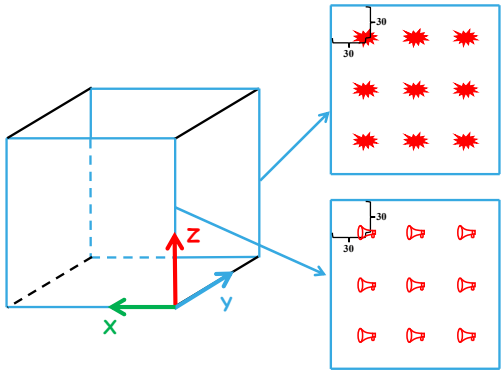


Figure 8: Illustration of the observation configuration with the y-axis as the principal axis.

### C.2 Implementations of our framework and baselines

For FWI-based methods, we use the Deepwave package to perform forward modeling with a fourth-order finite-difference scheme [58]. We use a Ricker wavelet with a dominant frequency of 100 Hz as the source wavelet. The sampling interval of the receivers is set to 0.00015 s, with a total of 500 time steps. To mitigate artificial boundary reflections, a perfectly matched layer (PML) boundary condition with a thickness of 20 grid points is implemented surrounding the entire model domain. To alleviate the memory burden introduced by 3D FWI while maintaining computational efficiency as much as possible, we incorporate two techniques: accumulating gradients over batches and CPU-based wavefield storage. All experiments are conducted using PyTorch with the Adam optimizer on two NVIDIA A100 GPUs.

The implementation details of the baselines and our proposed reconstruction framework are described as follows:

- **Our framework**: We employ an MLP with the SIREN architecture to reparameterize the velocity field. We set the frequency factor controlling the sinusoidal activation functions to 30. The employed MLP contains four layers, each with 128 neurons, and the learning rate is set to 8e-5. An L1-norm loss function is adopted to accelerate the convergence speed. The frequency parameter controlling the Fresnel volume region is gradually increased from 300 to 900 Hz to introduce a multi-scale effect. We pretrain the MLP using a homogeneous background velocity field to introduce prior. The pretraining process consists of 2000 iterations with a learning rate of 1e-3.
- **Total variation regularized FWI (TV-FWI)**: We normalize all velocity values and set the learning rate to 1. An L2-norm isotropic total variation (TV) regularization is applied to constrain the inversion process, with the regularization coefficient set to 6e-5. Since conventional FWI methods are highly sensitive to the choice of the initial model, we generate the initial model for TV-FWI by smoothing the ground-truth model with a Gaussian filter of standard deviation 10.
- **Frequency multi-scale FWI (MS-FWI)**: Following the same settings as TV-FWI, MS-FWI adopts the same normalization procedure, learning rate, and initial model. At each iteration, we apply a sixth-order Butterworth low-pass filter to the observed data, with the cutoff frequency gradually increased from [30, 50, 70, 90, 120, 150] Hz.
- **Implicit FWI (IFWI)**: For a fair comparison, IFWI and our framework use the same MLP configuration and incorporate a homogeneous background velocity model as prior information. Based on experimental evaluation, we select the optimal training strategy:

using an L2 loss function during pretraining, with 2000 iterations and a learning rate of 2e-4. For the subsequent optimization stage, an L1 loss is adopted with a learning rate set to 6e-4.

- **INR-Grid based FWI (IG-FWI)**: IG-FWI employs a hybrid representation combining SIREN and multi-resolution hash grids to model the velocity field. Following the settings in [13], the hash grid employs 16 levels of encoding, with a base resolution of 50 and a per-level scaling factor of 1.05. The features extracted from the hash grid are concatenated with those generated by an MLP (2 layers with 128 neurons each). The combined features are then fed into a compact MLP (2 layers with 64 neurons each) to finally output the velocity field. IG-FWI adopts the same training settings as IFWI, with learning rates of 5e-4 and 4e-4 for the pretraining and main training stages, respectively.

### C.3 Additional evaluation scenario settings

Beyond the standard velocity reconstruction setting adopted in the main experiments, we further investigate the robustness of our framework in the presence of noise. It is worth noting that FWI and TTT methods are affected by noise in different ways. For FWI methods, the use of full wavefield data may cause the optimization process to inadvertently fit noise present in the observations. In contrast, TTT methods rely only on first-arrival traveltimes. Thus, noise primarily affects TTT methods by introducing errors in first-arrival picking. We add zero-mean Gaussian noise at SNRs of 10 dB, 5 dB, and 0 dB to the waveform data ,and use the resulting noisy data as observations for the FWI methods. To ensure a fair comparison, we apply the conventional cross-correlation method to pick first-arrival traveltimes from the corresponding noisy data and use them as observations for the TTT methods.

### C.4 Fresnel Slice Net settings

**Architecture.** With Fresnel Slice Net (FS-Net), the Fresnel slices can be predicted by performing only local 2D convolution operations within the 3D space, and the predicted slices are then used to construct Fresnel volumes. Given the strong correlation between the effective weights in Fresnel slices and network inputs, as well as the need for computational efficiency, FS-Net is designed with a lightweight architecture, as shown in Table 4. Specifically, we employ an encoder-decoder convolutional neural network (CNN) as the backbone. A multi-branch MLP is used to generate modulation coefficients for incorporating auxiliary features into the training process. The total number of parameters in FS-Net is only 8,317.

Table 4: Architecture of the proposed FS-Net.

| **Backbone CNN** | | | |
|---|---|---|---|
| Stage | Operation | Channels | Parameters |
| Input Block | Conv $3 \times 3$ + BN + ReLU | $2 \rightarrow 4$ | stride=1 |
| Input Block | Conv $3 \times 3$ + BN + ReLU | $4 \rightarrow 4$ | stride=1 |
| Encoder Block 1 | Feature Modulation Block | $4 \rightarrow 8$ | stride=2 |
| Encoder Block 2 | Feature Modulation Block | $8 \rightarrow 16$ | stride=2 |
| Decoder Block 1 | Upsampling $\times 2$ + Feature Modulation Block | $16 \rightarrow 8$ | stride=1 |
| Decoder Block 2 | Upsampling $\times 2$ + Feature Modulation Block | $8 \rightarrow 4$ | stride=1 |
| Output Head | Conv $3 \times 3$ + BN + ReLU + Conv $1 \times 1$ + Sigmoid | $4 \rightarrow 1$ | stride=1 |
| **Modulation Parameter Generator** | | | |
| Module | Shared MLP | $4 \rightarrow 16$ | Linear + ReLU |
| Branch 1 | Linear Layer | $16 \rightarrow 16$ | $\boldsymbol{\gamma}_1, \boldsymbol{\beta}_1 \in \mathbb{R}^8$ |
| Branch 2 | Linear Layer | $16 \rightarrow 32$ | $\boldsymbol{\gamma}_2, \boldsymbol{\beta}_2 \in \mathbb{R}^{16}$ |
| Branch 3 | Linear Layer | $16 \rightarrow 16$ | $\boldsymbol{\gamma}_3, \boldsymbol{\beta}_3 \in \mathbb{R}^8$ |
| Branch 4 | Linear Layer | $16 \rightarrow 8$ | $\boldsymbol{\gamma}_4, \boldsymbol{\beta}_4 \in \mathbb{R}^4$ |

**Training.** The training details of FS-Net are presented in Table 5. The samples are split into the training and validation sets with a ratio of 99:1. The test set comprehensively covers samples with different frequencies. According to Eq. 28, we set the input and output sizes of FS-Net to $51 \times 51$. The voxels along the ray path are positioned at the center. We train the FS-Net for 20 epochs.

Table 5: Training dataset configurations and optimization settings.

| Scenario | Background Velocity (m/s) | Target Velocity (m/s) | Number of Samples | Learning Rate | Batch Size | Validation Loss |
|---|---|---|---|---|---|---|
| Standard & Noisy | 3000 | 1000 | ∼5.9M | $5 \times 10^{-3}$ | 2048 | 0.0091 |

### C.5 Metric

To evaluate the reconstruction accuracy, we adopt the commonly used metrics Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM), formulated as:

$$\mathrm{PSNR}(v, \hat{v}) = 10 \log_{10} \left( \frac{D_{\max}^2}{\frac{1}{N} \sum_{i=1}^{N} (v_i - \hat{v}_i)^2} \right) \tag{33}$$

$$\mathrm{SSIM}(v, \hat{v}) = \frac{(2\mu_v \mu_{\hat{v}} + C_1)(2\sigma_{v\hat{v}} + C_2)}{(\mu_v^2 + \mu_{\hat{v}}^2 + C_1)(\sigma_v^2 + \sigma_{\hat{v}}^2 + C_2)} \tag{34}$$

where $v$ and $\hat{v}$ denote the velocity values of the reconstructed result and the ground truth, respectively. $D_{\max}$ represents the maximum possible value. $\mu_v/\mu_{\hat{v}}$ and $\sigma_v^2/\sigma_{\hat{v}}^2$ denote the local mean and local variance, respectively. $\sigma_{v\hat{v}}$ represents the local covariance.

## D Additional Experimental Results

This section presents detailed quantitative and qualitative results for additional noisy scenarios.

The quantitative and qualitative comparisons under different noise levels are summarized in Table 6 and Fig. 9.As the noise level increases, the FWI baselines are severely degraded by fitting the noise in the observations, leading to noticeable artifacts and convergence to local minima. In contrast, our framework consistently maintains stable target reconstruction across different noise levels, with a particularly clear advantage under high-noise scenarios.This is mainly because the signal peak times extracted from the noisy data are used as the observed traveltimes for the TTT methods, which remain highly stable against noise interference for most data traces. For FWI methods, fitting the noise information during reconstruction leads to erroneous gradients. The best overall performance on the benchmark confirms that our framework can maintain robust and stable reconstruction under noisy scenarios.

Table 6: **Quantitative comparison under noisy scenarios.**

| Method | Duck | | | | | | Gerotor | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 10dB | | 5dB | | 0dB | | 10dB | | 5dB | | 0dB | |
| | *PSNR* ↑ | *SSIM* ↑ | *PSNR* ↑ | *SSIM* ↑ | *PSNR* ↑ | *SSIM* ↑ | *PSNR* ↑ | *SSIM* ↑ | *PSNR* ↑ | *SSIM* ↑ | *PSNR* ↑ | *SSIM* ↑ |
| TV-FWI | 21.75 | 0.7776 | 21.69 | 0.7706 | 21.61 | 0.7565 | 24.02 | 0.8275 | 23.87 | 0.8200 | 23.41 | 0.7974 |
| MS-FWI | 22.23 | 0.7977 | 22.13 | 0.7868 | 22.05 | 0.7728 | 23.70 | 0.8179 | 23.96 | 0.8050 | 23.52 | 0.7941 |
| IFWI | 23.19 | 0.8665 | 20.77 | 0.8411 | 20.87 | 0.8029 | 22.37 | 0.8512 | 18.84 | 0.7889 | 17.56 | 0.7531 |
| IG-FWI | 22.51 | 0.8629 | 20.34 | 0.8125 | 17.34 | 0.7527 | 18.13 | 0.7909 | 17.31 | 0.7624 | 16.81 | 0.7427 |
| **Ours** | 23.14 | 0.9311 | 22.55 | 0.9251 | 22.42 | 0.9173 | 26.19 | 0.9540 | 25.87 | 0.9529 | 25.84 | 0.9528 |
| **Method** | **Geometry** | | | | | | **Bunny** | | | | | |
| | 10dB | | 5dB | | 0dB | | 10dB | | 5dB | | 0dB | |
| | *PSNR* ↑ | *SSIM* ↑ | *PSNR* ↑ | *SSIM* ↑ | *PSNR* ↑ | *SSIM* ↑ | *PSNR* ↑ | *SSIM* ↑ | *PSNR* ↑ | *SSIM* ↑ | *PSNR* ↑ | *SSIM* ↑ |
| TV-FWI | 17.38 | 0.6461 | 17.40 | 0.6444 | 17.49 | 0.6378 | 19.19 | 0.7160 | 19.16 | 0.7139 | 19.19 | 0.7073 |
| MS-FWI | 21.10 | 0.7565 | 21.23 | 0.7499 | 21.12 | 0.7392 | 20.37 | 0.7446 | 20.28 | 0.7309 | 20.15 | 0.7186 |
| IFWI | 22.02 | 0.8774 | 19.74 | 0.8377 | 17.94 | 0.7435 | 24.14 | 0.8981 | 20.98 | 0.8342 | 20.79 | 0.8206 |
| IG-FWI | 22.11 | 0.8986 | 18.58 | 0.8252 | 15.72 | 0.7135 | 16.16 | 0.7073 | 14.41 | 0.6747 | 14.35 | 0.6782 |
| **Ours** | 23.79 | 0.9182 | 23.54 | 0.9064 | 22.15 | 0.9027 | 23.75 | 0.9174 | 23.40 | 0.9121 | 22.24 | 0.9054 |

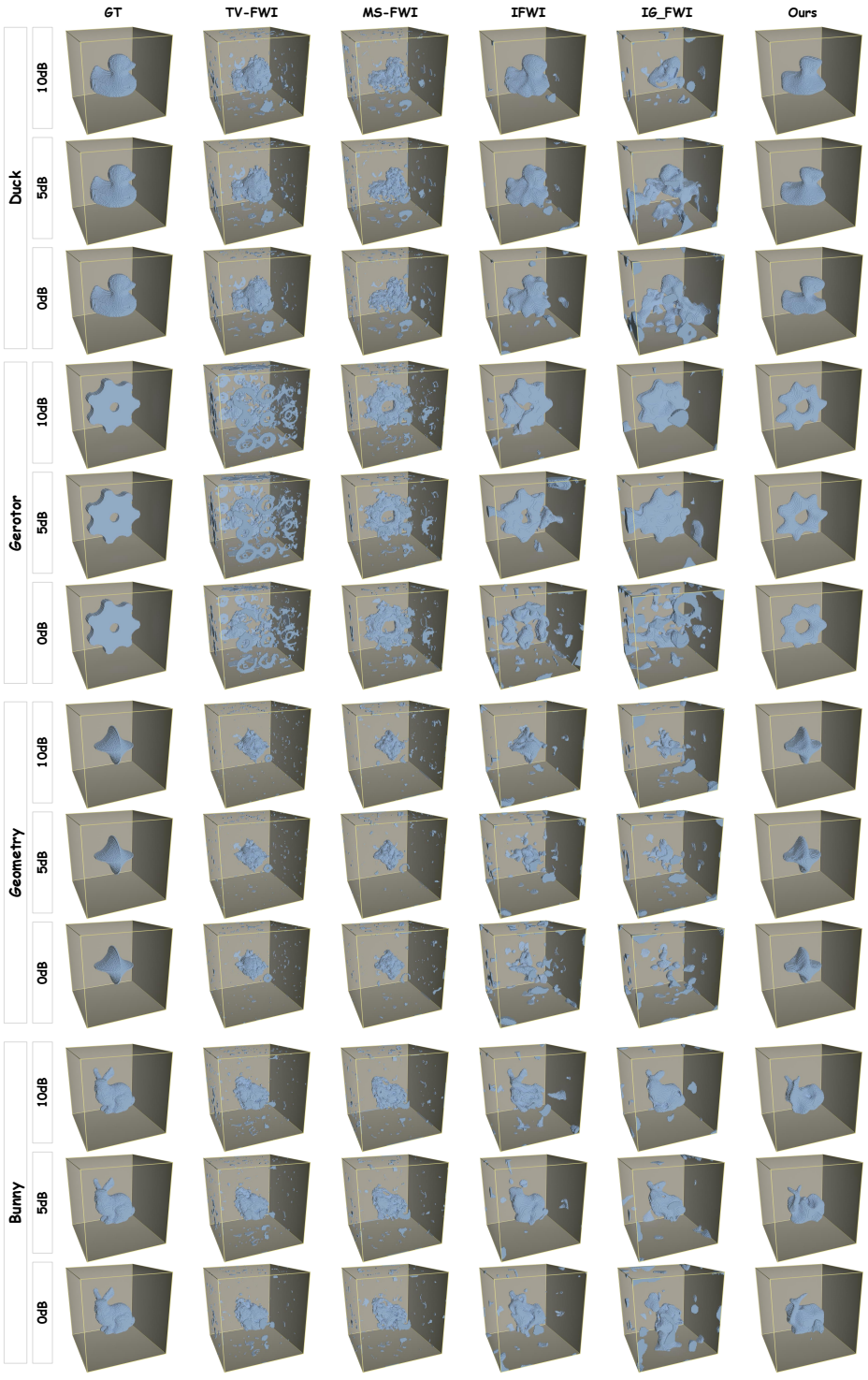


Figure 9: **Qualitative comparison under noisy scenarios.**

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately reflect the paper's contributions, which are further detailed in Sec.4 Method. All claimed improvements are experimentally validated in Sec.5 Experiments.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of the proposed method are discussed in Sec.6 Limitations.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: For each theoretical result, we provide the corresponding assumptions and detailed derivations, along with supporting references to prior work in Sec.4 Method.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We fully describe the main experimental procedure and settings necessary to reproduce the reported results in Sec.4 Method and Sec.5 Experiments. Additional implementation details follow standard practices and are clarified via references to prior work.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer:[No]

   Justification: The code will be released upon acceptance. All experiments are conducted on data generated using publicly available models, which ensures that the experimental setup is reproducible once the code is released.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide descriptions of the experimental settings in Sec.5.1. Certain implementation details follow standard protocols from prior work and are clarified through appropriate references.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The paper provides extensive results across five models, multiple metrics, and ablation studies. As our method is an optimization-based reconstruction approach rather than a large-scale data-driven training model, the experiments are largely deterministic and do not include error bars or statistical significance measures.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide sufficient information on computational resources, including the type of computing hardware (server configuration), memory consumption, and runtime for the experiments, as described in Sec.5 Experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research presented in this paper conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

Justification: The work is methodological in nature and focuses on optimization-based reconstruction without direct deployment in human-centered or real-world decision-making scenarios; therefore, it does not explicitly address societal impact.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The work does not involve high-risk models or datasets that require special safeguards for responsible release.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All data and code used in this work are properly credited to their original sources, and we explicitly follow and respect their respective licenses and terms of use as cited in the paper.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This work does not introduce any new assets; all datasets and models used in the experiments are existing and publicly available.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or human subject studies.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve human subjects or studies requiring IRB approval.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not used in the proposed method or experiments, and are only potentially relevant for writing-related assistance.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.