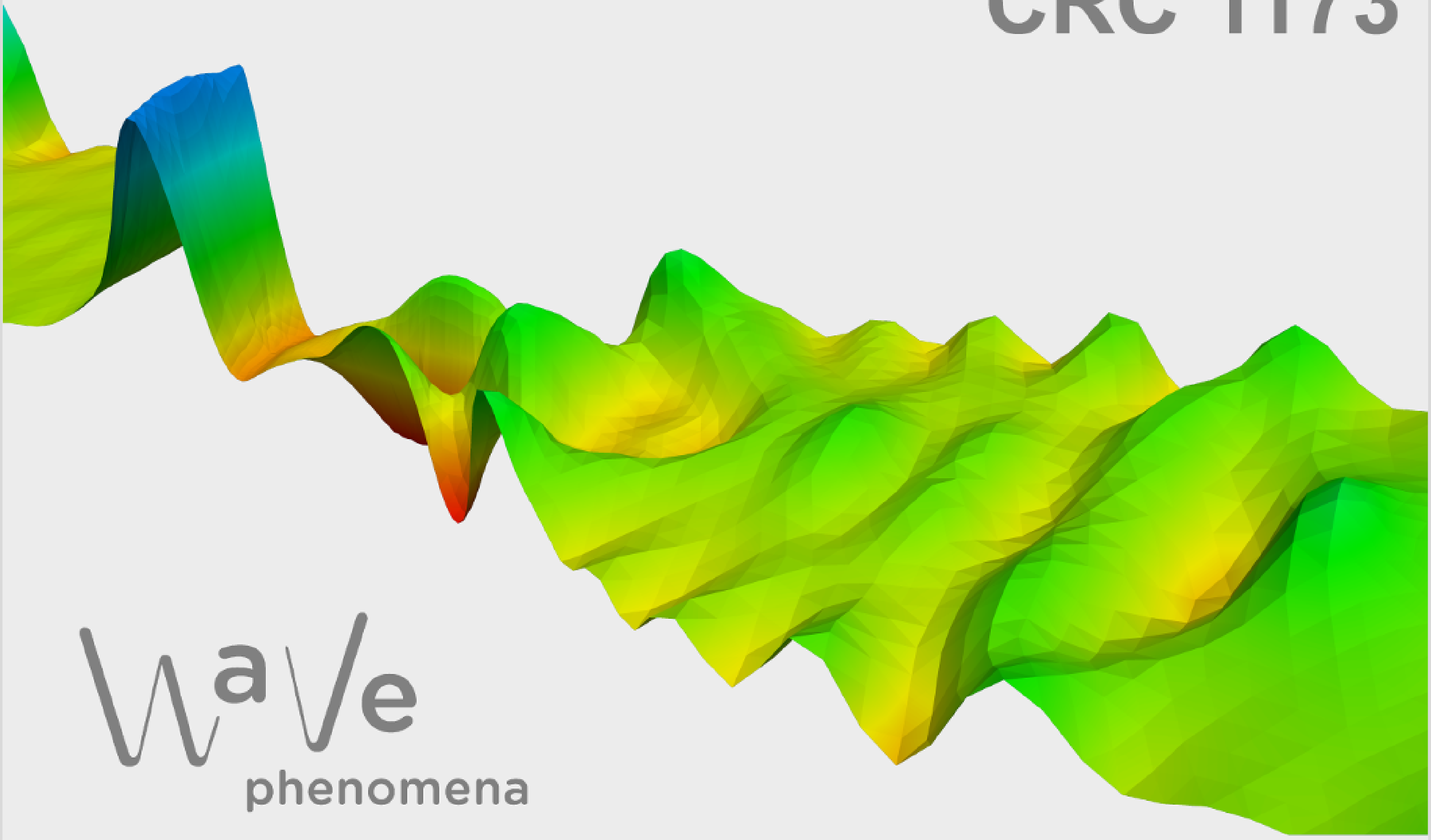
Learned distance functionals and the Landweber method for inverse problems with an application to full waveform inversion

David Hämmerling, Lukas Pieronek, Andreas Rieder

CRC Preprint 2026/8, March 2026

KARLSRUHE INSTITUTE OF TECHNOLOGY

CRC 1173



Participating universities

UNIVERSITÄT BONN

Universität Stuttgart

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

TU
WIEN

Funded by

DFG

ISSN 2365-662X

LEARNED DISTANCE FUNCTIONALS AND THE LANDWEBER METHOD FOR INVERSE PROBLEMS WITH AN APPLICATION TO FULL WAVEFORM INVERSION

DAVID HÄMMERLING[1], LUKAS PIERONEK[2], AND ANDREAS RIEDER[1]

Abstract. Nonlinear inverse problems are typically solved by minimizing a data-misfit functional, which is often non-convex that leads minimization algorithms to stagnate at a local minimum. A typical example is the cycle-skipping phenomenon in full waveform inversion (FWI) of seismic reflection or transmission data. To overcome this difficulty, distance functionals are constructed by training neural networks to emulate a convex distance measure. Two construction strategies are discussed: (i) a data-converter network that simplifies the forward map so that a standard quadratic loss becomes convex, and (ii) a scalar-valued distance-network based on the data residual. Training samples can be generated from measured data exploiting an approximate invariance of the forward operator. Under a set of structural assumptions, it is proven that applying the Landweber iteration to the learned functional is a well-defined regularization method that guarantees monotone error reduction and convergence to the exact solution as the noise level vanishes. Given certain restrictions on the trained network and the nonlinear forward operator, it is validated that these structural assumptions are satisfied by the learned distance. This methodology is numerically validated on the Camembert benchmark model for FWI in the acoustic regime. Replacing the conventional L^2-misfit with the learned convexified functional considerably mitigates cycle-skipping and enlarges the domain of convergence for gradient-based optimization. Numerical experiments show that the Landweber scheme with the learned misfit reaches a lower reconstruction error than the standard least-squares approach. For comparison, in some experiments convergence has been accelerated by a limited-memory BFGS optimizer leading to slightly larger reconstruction errors. Overall, the work provides a systematic way to embed learned, convex distance measures into inverse problem solvers, supplies a solid theoretical foundation for their use, and demonstrates practical gains in a seismic imaging benchmark model.

This work is dedicated to the memory of Alfred K. Louis. He was both academic mentor and friend of the third author who is deeply grateful for his unconditional support, his enduring trust, and his inspiring lessons.

[1] Department of Mathematics, Karlsruhe Institute of Technology (KIT), D-76128 Karlsruhe, Germany

[2] Carl Zeiss AG, Corporate Research and Technology, Carl-Zeiss-Straße 22, D-73447 Oberkochen, Germany

E-mail address: david.haemmerling@kit.edu, pieronek.lukas@gmail.com, andreas.rieder@kit.edu.

Date: March 12, 2026.

2020 *Mathematics Subject Classification.* 65J20, 86A22.

Key words and phrases. nonlinear inverse and ill-posed problem, regularization, learned distance function, time domain full waveform inversion.

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - Project-ID 258734477 - SFB 1173.

1. Introduction

We are concerned with the solution of a general nonlinear inverse problem

$$F(\cdot) = y \tag{1}$$

where the forward map $F\colon \mathrm{D}(F) \subset X \to Y$ acts between real Hilbert spaces X and Y. We assume that (1) is locally ill-posed at any point in $\mathrm{D}(F)$, see [27, Def. 1.1]. Concrete examples include electric impedance tomography, inverse electromagnetic scattering, and seismic imaging, see [14, Part IV].

A common way to approach (1) is to frame it as the generally non-convex optimization problem of minimizing the data misfit

$$\mathrm{D}(F) \ni x \mapsto \|F(x) - y^\delta\|_Y^2$$

for measured data $y^\delta \in Y$ by gradient descent-like schemes, which are stopped in time to avoid noise amplification. However, due to the non-convexity of the misfit, these schemes may become trapped in a local minimum unless they are started sufficiently close to a global minimum. Therefore, it is crucial to replace the non-convex data misfit with a convex function

$$d_y\colon X \to [0, \infty),$$

that measures the distance of $F(x)$ to a fixed $y \in Y$.

A particularly interesting and topical nonlinear inverse problem is the seismic inverse problem for reconstructing the internal structure of the Earth using seismograms, i.e., measurements of reflected wave fields. Full waveform inversion (FWI) is the most advanced solution technique for this task, as it can exploit the entire information content of the seismograms, see, e.g., [21, 44]. However, in its standard formulation, FWI is heavily affected by the previously explained non-convexity, known as cycle-skipping in the geophysical community. Therefore, many attempts have been made to mitigate cycle-skipping by replacing the image space norm $\|\cdot\|_Y$ with an improved misfit functional $\Delta\colon Y \times Y \to \mathbb{R}$ such that the associated cost functional

$$d_y(x) := \Delta(F(x), y)$$

for a given set of seismograms $y = F(x^+) \in Y$ is convex or has a wider range of convergence for standard optimization methods in a neighborhood of the ground truth x^+.

Approaches to finding an FWI-adapted distance Δ include, for example, optimal transport-based misfits (OT) [18, 19, 24, 34, 35, 36], objective functions obtained from reduced order modeling (ROM) [10], and data-driven neural network distances [13, 43]. Although OT offers some interesting convexity properties, its evaluation is computationally demanding and seismic data do generally not fulfill the requirements from OT naturally, such as positivity. This is because seismograms are interpreted as "mass" distributions therein and computing Δ amounts to solving a (regularized) transport optimization problem under a chosen ground cost. Applicability is usually achieved through hand-crafted data transformations which may in turn reintroduce non-convex behavior, see [19], or by using higher-dimensional data embeddings which further increase the computational burden, see [24, 35]. Finally, this procedure must be repeated across shots/receivers and iterations, making it far more expensive than pointwise norms. On the other hand, using the original least-squares formulation, [10] proposes a ROM-based preconditioner to transform the measurement data. This transformation is shown to improve the optimization landscape in practice, but due to its implicit definition, the method lacks a theoretical justification specifying the assumptions under which convexity can be expected.

In contrast, learned misfit functionals can be regarded as cheap-to-evaluate data-driven surrogates of such convex metrics, prepending the computational costs to the initial one-time training and thus providing fast inference throughout the iteration afterwards. Moreover, they allow an analysis as provided in [13] for a Tikhonov-type regularization by using the learned network's output as data-fitting term. In this context, we would like to mention the work [31], where the penalty term in Tikhonov regularization is defined via a neural network. The training process enforces small regularization values on plausible solutions and larger values on implausible ones. This approach is versatile and theoretically backed up as, under suitable assumptions, the regularization property is validated. A more refined approach to learned regularization terms in the context of frequency domain FWI can be found in [38], where a denoising network is combined with a wavelet basis to construct a regularization term. The resulting regularization objective is optimized via a multiparameter approach in both the wavelet and standard bases simultaneously. While suitable regularization terms and enlarged parameter spaces can decrease the non-convexity of the optimization objective, they do not address the nonlinearity of the forward operator F directly.

From our discussion of alternative misfit functionals above, it became clear that two important shortcomings remain:

(1) Existing learned misfits are usually introduced heuristically and lack a rigorous regularization analysis.
(2) To the best of our knowledge, there is currently no general regularization theory for Landweber/gradient-type iterations driven by a *general* convex objective functional d_y (beyond the classical quadratic/Hilbert-space misfit and special structured choices).

Consequently, it is unclear under which conditions the learned functional guarantees monotone error reduction, convergence to the exact solution, or stability with respect to noise.

In this paper we address the above gap and make the following contributions.

1. Under a set of structural assumptions (convexity, Lipschitz continuity of the gradient, bounded linearization error) we prove in Section 2 that the Landweber iteration applied to the learned functional is a regularization scheme with monotone error reduction and convergence as the noise level decreases (Theorems 2.4 and 2.5, Corollary 2.6).

2. In Section 3 we propose two construction strategies for learned convex distance functionals:

(i) a conversion network Φ_θ that simplifies the forward map so that a standard quadratic loss becomes convex, i.e.,

$$\frac{1}{2}\left\|\Phi_\theta \circ F(x) - \Phi_\theta(y)\right\|_Y^2$$

is convex in x (Section 3.1). Further we present a training concept in Section 3.1.1 to stably train such a simplifier network. For the case $\Phi_\theta \circ F \approx \mathrm{Id}$, i.e., Φ_θ is an approximate inverse, there exists a large variety of research in the field of FWI [3, 13, 32, 45, 46] and other tomography problems [28, 47] as well. However those approaches use the learned inverter Φ_θ to directly reconstruct the parameters from the perturbed data y^δ and thus require an extensive and well-chosen training set like OpenFWI [12]. We, however, avoid these issues by learning a network as a more general simplifier for an optimization method and thus require a less refined training process and choice of data.

(ii) a scalar-valued distance network that directly predicts a convex distance between two measurements (Section 3.2).

Furthermore, given certain restrictions on the trained network and the nonlinear forward operator, we prove that both versions of the learned distance meet a majority of the structural assumptions (Lemmas 3.1, 3.3, and 3.5). Since we could only validate a weak convexity of our learned functionals, we therefore provide an adapted regularization theory in Appendix A.

3. We suggest practical guidelines for generating training samples from measured data by exploiting approximate invariances of the forward operator (Remark 3.4).

4. In Section 4 we give a proof of concept by demonstrating our theory on the Camembert benchmark model for time domain FWI in the acoustic regime. Replacing the conventional L^2-misfit with the learned convex functional dramatically reduces reconstruction error, mitigates cycle-skipping, and enlarges the convergence domain. In some experiments, a limited-memory BFGS optimizer accelerates convergence while preserving the accuracy gain over the L^2-misfit.

Section 5 concludes our exposition with a summary and an outlook on future research topics.

2. Misfit Landweber method

In the next section, we will present concepts from machine learning to construct a misfit functional $d_y\colon X \to [0,\infty)$ for $y \in Y$ such that $d_{F(x^+)}(x) \approx \frac{1}{2}\|x - x^+\|_X^2$. The object we are seeking then is the minimizer of d_y, which we compute by an adapted steepest descent method, terminated by the discrepancy principle, see Algorithm 1. The goal of the present section is to analyze the convergence of this algorithm under meaningful assumptions on d_y which we discuss and justify in Remark 2.2 below.

Assumption 2.1. *Let $y \in Y$. We assume that our misfit functional $d_y\colon X \to [0,\infty)$ matches the following conditions:*

(i) *The misfit functional is Fréchet-differentiable at any $x \in X$ with Riesz-representation $d_y'(x) \in X$ such that*

$$\partial_x d_y(x)[h] = \langle d_y'(x), h\rangle_X.$$

(ii) *The misfit d_y is convex and has a unique minimizer $x^+ \in X$.*

(iii) *It holds that*

$$d_{F(x)}(x) = 0 \quad \textit{for all } x \in \mathrm{D}(F).$$

(iv) *There is a constant $c_y > 0$ such that*

$$c_y \left\| d_y'(x) \right\|_X^2 \leq d_y(x) \quad \textit{for all } x \in X.$$

(v) *There exists a functional $d\colon Y \times X \to [0,\infty)$ such that $d_y(\cdot) = d(y,\cdot)$ and which is Fréchet-differentiable with respect to y at $(F(x^+), x^+)$. Set*

$$\eta_y := \left\| \partial_y d(F(x^+), x^+) \right\|_{Y \to \mathbb{R}}.$$

(vi) *There is a constant* $\zeta_y > 0$ *such that*

$$\zeta_y \, \|x - x^+\|_X^2 \leq d_y(x) \quad \textit{for all } x \in X.$$

Assumption 2.1 needs some motivation and explanation, which we provide in the following remark.

Remark 2.2. a) *In the ideal case, we would like to have a misfit measure* d_y *yielding* $d_{F(x^+)}(x) = \frac{1}{2}\|x - x^+\|_X^2$. *So, we try to learn a distance concept* d_y *such that* $d_{F(x^+)}(x) \approx \frac{1}{2}\|x - x^+\|_X^2$ *for* x *in a ball about* x^+ *with a radius as large as possible. To this end we train an artificial neural network* Φ *to achieve* $\Phi \approx F^{-1}$. *Then, we set* $d_y(x) = d(y,x) := \frac{1}{2}\|x - \Phi(y)\|_X^2$. *For this distance concept, it is reasonable to expect the requirements in Assumption* 2.1 *to be met. To see this, accept for the moment that* F^{-1} *is well defined on* Y *and Fréchet-differentiable. Then,* $\widetilde{d}_y(x) = \widetilde{d}(y,x) = \frac{1}{2}\|x - F^{-1}(y)\|_X^2$ *satisfies Assumption* 2.1. *In fact,* (i)-(iii) *are obvious with* $x^+ = F^{-1}(y)$ *and* $\widetilde{d}_y'(x) = x - F^{-1}(y)$. *Further,* (iv) *holds as an equality for* $c_y = 1/2$. *Finally, we have* (v) *and* (vi) *with* $\eta_y = 0$ *and* $\zeta_y = 1/2$, *respectively. The knowledgeable reader may object here, that the latter assumptions on* F *make the inverse problem* (1) *well-posed. This objection is of course true, but a large variety of inverse problems are conditionally well-posed on certain closed subspaces of* X, *see, e.g.,* [1, 2, 5, 6, 7, 8, 9, 11, 15, 16, 26, 30, 42], *and so it is reasonable to consider our inverse problem* (1) *to be conditionally well-posed.*

b) *We would like to point out that we can delete condition* (vi) *of Assumption* 2.1 *when we replace* (ii) *by*

(ii') *The misfit* d_y *is uniformly convex, that is, for all* $u, v \in X$ *and all* $\lambda \in [0,1]$,

$$d_y(\lambda u + (1-\lambda)v) + \lambda(1-\lambda)\phi_y(\|u - v\|_X) \leq \lambda d_y(u) + (1-\lambda)d_y(v)$$

with $\phi_y \colon [0,\infty) \to [0,\infty)$ *non-decreasing and vanishing only at* 0.

As a consequence of (ii'), *the minimizer of* d_y *is unique and*

$$\frac{1}{2}\phi_{F(x^+)}\big(\|x - x^+\|\big) \leq d_{F(x^+)}(x) \tag{2}$$

which is the substitute for (vi). *All subsequent results remain correct under this modification of Assumption* 2.1 *(except for Corollary* 2.6, *see Remark* 2.7*). Note that* $\widetilde{d}_y$ *from part a) of this remark also fulfills* (ii') *with* $\phi_y(t) = t^2/2$.

c) *Also one could change condition* (iv) *to a slightly stronger Lipschitz-condition for* d_y':

(iv') *There is a constant* $\widetilde{c}_y > 0$ *such that*

$$\big\|d_y'(x)\big\|_X = \big\|d_y'(x) - d_y'(x^+)\big\|_X \leq \widetilde{c}_y \|x - x^+\|_X \quad \textit{for all } x \in X.$$

Indeed, (iv') *together with* (vi) *yields* (iv).

A straightforward gradient descent for d_y leads to Algorithm 1 where $\delta \geq 0$ is a bound for possible noise in the data, i.e., $\|y - F(x^+)\|_Y \leq \delta$. The used stopping criterion is the discrepancy principle. We now prove that the steepest decent iteration of Algorithm 1 is a regularization following three steps:

1. well-definedness and convergence for every $y = F(x^+)$ to the exact data x^+ (Theorem 2.3),
2. monotone error decrease under perturbed data until termination (Theorem 2.4),
3. regularization property (Theorem 2.5).

Algorithm 1 Misfit steepest descent

```
Input: $y \in Y,\ \delta \geq 0$    %input data with noise level
       $\tau > 0$    %parameter for the discrepancy principle
       $x_0 \in X,\ \{\lambda_n\}_{n\in\mathbb{N}_0} \in (0,\infty)^{\mathbb{N}}$    %inital guess and sequence of step sizes
  $n := 0$
  while $d_y(x_n) > \tau\delta$ do
      $x_{n+1} := x_n - \lambda_n d_y'(x_n)$
      $n++$
  end while
```

Theorem 2.3. *Under Assumption 2.1 consider Algorithm 1 with input $y = F(x^+)$ for $x^+ \in \mathrm{D}(F)$, $\delta = 0$, and $\{\lambda_n\}_{n\in\mathbb{N}_0} \in (0, 2c_y]^{\mathbb{N}_0}$ where $c_y > 0$ is the constant from* (iv) *in Assumption 2.1. Moreover,*

$$\sum_{n=0}^{\infty} \lambda_n = \infty \quad \textit{and} \quad \sum_{n=0}^{\infty} \lambda_n^2 < \infty. \tag{3}$$

Then, for $x_0 \in X$, the sequence

$$x_{n+1} := x_n - \lambda_n d_y'(x_n), \quad n \in \mathbb{N}_0,$$

generated by Algorithm 1 either stops after a finite number of iterations with x^+ or converges to x^+: $\lim_{n\to\infty} x_n = x^+$. In both cases, we have a monotone error decrease

$$\|x_{n+1} - x^+\|_X \leq \|x_n - x^+\|_X. \tag{4}$$

Proof. If Algorithm 1 terminates with x_N we have that $d_y(x_N) = 0$. Since $y = F(x^+)$ and since the minimizer of d_y is unique by Assumption 2.1(ii), we conclude with Assumption 2.1(iii) that $x_N = x^+$.

Now, assume that $d_y(x_n) > 0$ for all n. Using $d_y(x^+) = 0$ and the convexity of d_y in the form of

$$\langle d_y'(u), v - u\rangle_X \leq d_y(v) - d_y(u), \tag{5}$$

we get

$$\begin{aligned} \|x_{n+1} - x^+\|_X^2 &= \|x_n - \lambda_n d_y'(x_n) - x^+\|_X^2 \\ &= \|x_n - x^+\|_2^2 - 2\lambda_n\langle d_y'(x_n), x_n - x^+\rangle_X + \lambda_n^2\|d_y'(x_n)\|_X^2 \\ &\leq \|x_n - x^+\|_X^2 + 2\lambda_n(d_y(x^+) - d_y(x_n)) + \lambda_n^2\|d_y'(x_n)\|_X^2 \\ &= \|x_n - x^+\|_X^2 - 2\lambda_n d_y(x_n) + \lambda_n^2\|d_y'(x_n)\|_X^2. \end{aligned} \tag{6}$$

By Assumption 2.1(iv) and $\lambda_n \leq 2c_y$,

$$\|x_{n+1} - x^+\|_X^2 \leq \|x_n - x^+\|_X^2 + 2\lambda_n(-d_y(x_n) + c_y\|d_y'(x_n)\|_X^2) \leq \|x_n - x^+\|_X^2$$

which is (4). Thus, $x_n \in B_\rho(x^+)$ for all $n \in \mathbb{N}_0$ where $\rho = \|x_0 - x^+\|_X$. By recursively repeating the estimate (6), we obtain

$$\|x_{n+1} - x^+\|_X^2 \leq \|x_0 - x^+\|_X^2 - 2\sum_{i=0}^{n} \lambda_i d_y(x_i) + c_y \sum_{i=0}^{n} \lambda_i^2 d_y(x_i) \tag{7}$$

where we have used again Assumption 2.1(iv). Now, by (i) of Assumption 2.1,

$$d_y(x_i) = d_y(x_i) - d_y(x^+) = \underbrace{d'_y(x^+)}_{=0}[x_i - x^+] + \mathrm{o}(\|x_i - x^+\|_X)$$

and by $x_i \in B_\rho(x^+)$ for all $i \in \mathbb{N}_0$, there exists a bound b_ρ such that $d_y(x_i) \le b_\rho$ for all $i \in \mathbb{N}_0$. Bounding the left hand side of (7) from below by zero implies first that

$$2\sum_{i=0}^{n} \lambda_i d_y(x_i) \le \rho^2 + c_y b_\rho \sum_{i=0}^{n} \lambda_i^2$$

and then

$$\min_{i\in\{0,\ldots,n\}} d_y(x_i) \le \frac{\rho^2 + c_y b_\rho \sum_{i=0}^{n} \lambda_i^2}{2\sum_{i=0}^{n} \lambda_i}.$$

Our assumptions on $\{\lambda_n\}$ and taking the limit as $n \to \infty$ yield

$$\inf_{i\in\mathbb{N}_0} d_y(x_i) = 0.$$

Hence, $\{d_y(x_i)\}_{i\in\mathbb{N}_0}$ has a subsequence $\{d_y(x_{i_k})\}_{k\in\mathbb{N}}$ which converges to 0. By (vi) of Assumption 2.1 the subsequence $\{x_{i_k}\}_{k\in\mathbb{N}}$ converges to x^+. Due to the monotonicity (4), the entire sequence must converge. □

In the following we have to distinguish clearly between quantities depending on perturbed data y^δ and on exact data y. To this end, the former are marked by a superscript δ.

Theorem 2.4. *Let $y = F(x^+)$ and let $y^\delta \in Y$ such that $0 < \|y^\delta - y\|_Y \le \delta$. Let both, d_y and d_{y^δ}, fulfill Assumption 2.1. Call Algorithm 1 with input y^δ, δ, $\tau > 2\eta_y$, $x_0^\delta \in X$, and step sizes $\{\lambda_n^\delta\} \subset (0, c_{y^\delta}]^{\mathbb{N}}$ which satisfy the left equation of (3). Here, c_{y^δ} and η_y are as in* (iv) *and* (v) *of Assumption 2.1, respectively. Then, for $\delta > 0$ sufficiently small, Algorithm 1 stops after a finite number $N(\delta)$ of iteration steps and the iterates satisfy*

$$\|x_{n+1}^\delta - x^+\|_X < \|x_n^\delta - x^+\|_X \quad \text{for all } n < N(\delta).$$

Proof. Relying again on (5) we can estimate

$$\begin{aligned}\|x_{n+1}^\delta - x^+\|_X^2 &= \|x_n^\delta - \lambda_n^\delta d'_{y^\delta}(x_n^\delta) - x^+\|_X^2 \\ &= \|x_n^\delta - x^+\|_X^2 - 2\lambda_n^\delta \langle d'_{y^\delta}(x_n^\delta), x_n^\delta - x^+\rangle_X + (\lambda_n^\delta)^2 \|d'_{y^\delta}(x_n^\delta)\|_X^2 \\ &\le \|x_n^\delta - x^+\|_X^2 + 2\lambda_n^\delta\big(d_{y^\delta}(x^+) - d_{y^\delta}(x_n^\delta)\big) + (\lambda_n^\delta)^2 \|d'_{y^\delta}(x_n^\delta)\|_X^2.\end{aligned}$$

With $d_y(x^+) = 0$ and $\lambda_n^\delta \le c_{y^\delta}$ we get

$$\begin{aligned}\|x_{n+1}^\delta - x^+\|_X^2 \le \|x_n^\delta - x^+\|_X^2 &+ 2\lambda_n^\delta\big(d_{y^\delta}(x^+) - d_y(x^+) - \partial_y d(y, x^+)[y - y^\delta] \\ &+ \partial_y d(y, x^+)[y - y^\delta] - d_{y^\delta}(x_n^\delta)\big) + \lambda_n^\delta c_{y^\delta} \|d'_{y^\delta}(x_n^\delta)\|_X^2.\end{aligned}$$

By (iv) of Assumption 2.1,

$$\begin{aligned}\|x_{n+1}^\delta - x^+\|_X^2 \le \|x_n^\delta - x^+\|_X^2 + \lambda_n^\delta \big(2\left[d_{y^\delta}(x^+) - d_y(x^+) - \partial_y d(y, x^+)[y - y^\delta]\right] \\ + 2\partial_y d(y, x^+)[y - y^\delta] - d_{y^\delta}(x_n^\delta)\big).\end{aligned}$$

In view of Assumption 2.1(v) we bound

$$\partial_y d(y, x^+)[y - y^\delta] \le |\partial_y d(y, x^+)[y - y^\delta]| \le \|\partial_y d(y, x^+)\|_{Y\to\mathbb{R}} \|y - y^\delta\|_Y \le \eta_y \delta.$$

Assume that Algorithm 1 does not terminate. Then, $d_{y^\delta}(x_n^\delta) > \tau\delta$ for all n. Thus,

$$\|x_{n+1}^\delta - x^+\|_X^2 \le \|x_n^\delta - x^+\|_X^2 + \lambda_n^\delta \big(2\left[d_{y^\delta}(x^+) - d_y(x^+) - \partial_y d(y,x^+)[y-y^\delta]\right] \\ + \delta(2\eta_y - \tau)\big).$$

Since the asymptotic behavior of $d_{y^\delta}(x^+) - d_y(x^+) - \partial_y d(y,x^+)[y-y^\delta] = \mathrm{o}(\delta)$ as $\delta \to 0$ is independent of n and since $\tau > 2\eta_y$ we have that

$$\|x_{n+1}^\delta - x^+\|_X^2 \le \|x_n^\delta - x^+\|_X^2 - \lambda_n^\delta D(\delta)$$

where $D(\delta) := \mathrm{o}(\delta) + \delta(\tau - 2\eta_y) > 0$ for $\delta > 0$ sufficiently small independent of n. Recursively, for all n,

$$\|x_n^\delta - x^+\|_X^2 \le \|x_0^\delta - x^+\|_X^2 - D(\delta)\sum_{k=0}^{n-1}\lambda_k^\delta,$$

which leads to a contradiction in view of (3). Hence, Algorithm 1 terminates with the claimed error monotonicity.

□

Now we are ready to validate the regularization property of Algorithm 1.

Theorem 2.5. *Adopt the notation and assumptions of Theorem 2.4. If $x_0^\delta \to x_0$ and $\lambda_n^\delta \to \lambda_n \in (0, c_y]$ for all n as $\delta \to 0$, then*

$$\lim_{\delta\to 0} x_{N(\delta)}^\delta = x^+.$$

Proof. We benefit from standard arguments as given in [25]. Without loss of generality, we can assume that $N(\delta) \to \infty$ as $\delta \to 0$. According to Theorem 2.3, for any $\varepsilon > 0$ there exists an $m_\varepsilon \in \mathbb{N}$ such that $\|x_{m_\varepsilon} - x^+\|_X < \varepsilon$. Let $\delta > 0$ be so small that $N(\delta) \ge m_\varepsilon$. By Theorem 2.4,

$$\|x_{N(\delta)}^\delta - x^+\|_X \le \|x_{m_\varepsilon}^\delta - x^+\|_X < \|x_{m_\varepsilon}^\delta - x_{m_\varepsilon}\|_X + \varepsilon.$$

Under our assumptions, the stability property readily follows, that is, $x_{m_\varepsilon}^\delta \to x_{m_\varepsilon}$ as $\delta \to 0$, thereby completing the proof. □

Under an additional weak assumption, we can even validate a convergence order. This result is hardly surprising: As explained in Remark 2.2, under Assumption 2.1 the learned misfit $d_y(x)$ behaves like $\frac{1}{2}\|x - x^+\|_X^2$, and since Algorithm 1 is terminated by a discrepancy principle, we expect $\|x_{N(\delta)}^\delta - x^+\|_X \sim \sqrt{\delta}$ to hold.

Corollary 2.6. *Adopt the assumptions of Theorem 2.5. If $\partial_y d(F(x^+), \cdot)$ is continuous at x^+ then*

$$\|x_{N(\delta)}^\delta - x^+\|_X = \mathrm{O}\left(\sqrt{\delta}\right) \qquad \text{as } \delta \to 0.$$

Proof. We begin with Assumption 2.1(vi) and (v),

$$\zeta_y \|x_{N(\delta)}^\delta - x^+\|_X^2 \le d_y(x_{N(\delta)}^\delta) = d(y, x_{N(\delta)}^\delta),$$

and proceed according to

$$\begin{aligned} d(y, x_{N(\delta)}^\delta) &= d(y, x_{N(\delta)}^\delta) - d(y^\delta, x_{N(\delta)}^\delta) + d(y^\delta, x_{N(\delta)}^\delta) \\ &\le d(y, x_{N(\delta)}^\delta) - d(y^\delta, x_{N(\delta)}^\delta) + \tau\delta. \end{aligned}$$

Moreover,

$$d(y, x_{N(\delta)}^\delta) - d(y^\delta, x_{N(\delta)}^\delta) = \partial_y d(F(x^+), x_{N(\delta)}^\delta)[y - y^\delta] + \mathrm{o}(\delta)$$

where the $\mathrm{o}(\delta)$-term depends on y and not on y^δ. Hence,

$$|d(y, x^\delta_{N(\delta)}) - d(y^\delta, x^\delta_{N(\delta)})| \leq \|\partial_y d(F(x^+), x^\delta_{N(\delta)})\|_{Y\to\mathbb{R}}\, \delta + \mathrm{o}(\delta)$$

and the result follows from $x^\delta_{N(\delta)} \to x^+$ as $\delta \to 0$ and the assumed continuity of $\partial_y(F(x^+), \cdot)$ at x^+. □

If both, Assumption 2.1(iv) and (vi), hold only in a ball $B_\rho(x^+)$, then above results carry over provided that the initial guess x_0 is chosen in this ball. The reason for this lies in the error monotonicity (Theorems 2.3 and 2.4) so that all iterates stay in the ball.

Remark 2.7. *The statement of the above corollary needs to be adjusted if Assumption* 2.1 *is modified as discussed in Remark* 2.2b)*. Since then we have* (2) *instead of Assumption* 2.1(vi) *such that the convergence rate is given by*

$$\|x^\delta_{N(\delta)} - x^+\|_X = \mathrm{O}\left(\phi^{-1}_{F(x^+)}(\delta)\right) \qquad \textit{as } \delta \to 0.$$

3. Learning misfit functionals

In this section, we specify how to construct misfit functionals that satisfy as many of the conditions of Assumption 2.1 as possible.

The underlying nonlinear inverse problem (1) is typically formulated in an infinite dimensional setting. Therefore, in the first step we have to discretize it: Let $X_n \subset X$ and $Y_n \subset Y$ be finite dimensional subspaces. Define $F_n \colon \mathrm{D}(F_n) \subset X_n \to Y_n$ where $F_n(x) := Q_n F(x)$ with $Q_n \colon Y \to Y$ being the orthogonal projector onto Y_n. We assume that $\mathrm{D}(F_n) := \mathrm{D}(F) \cap X_n$ has no empty interior. Now, the discrete version of (1) reads

$$F_n(\cdot) = \mathrm{M}_n\, y$$

where $\mathrm{M}_n \colon Y \to Y_n$ models the measurement process. In the next two subsections we present two ways to construct misfit functionals for this discrete setting.

3.1. Data converter.

Let Z be a finite dimensional normed space. We train a neural network

$$\Phi_\theta \colon Y_n \to Z,$$

as a "simplifier" of the forward problem, i.e. it should hold in some sense that

$$\Phi_\theta \circ F_n(x) \approx Ax, \quad x \in \mathrm{D}(F_n), \tag{8}$$

for some "simple", not necessarily linear, operator $A \colon X_n \to Z$. Here, $\theta \in \mathbb{R}^p$ represents the p parameters determining the neural network. For example, these parameters could be obtained through a standard training process by minimizing the mean-square error

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{k=1}^{N} \|\Phi_\theta(F_n(\xi_k)) - A\xi_k\|_Z^2$$

for given data points $\xi_1, \ldots, \xi_N \in \mathrm{D}(F_n)$, see Section 3.1.1 below for more details on an implementation of this training procedure.

Having found an optimal θ, we define

$$\mathrm{dist}_\theta \colon Y_n \times \mathrm{D}(F_n) \to [0, \infty), \quad (y, x) \mapsto \frac{1}{2}\|\Phi_\theta(F_n(x)) - \Phi_\theta(y)\|_Z^2,$$

and the resulting misfit is $d_y(\cdot) := \mathrm{dist}_\theta(y, \cdot)$. Please note that, under the optimality assumption $\mathcal{L}(\theta) \approx 0$, we have

$$d_y(x) \approx \frac{1}{2}\|Ax - \Phi_\theta(y)\|_Z^2.$$

Accordingly, there are two requirements on A: First, A should be chosen so that the above right-hand side is convex in x. Second, property (8) should be feasible with reasonable training effort. The obvious choice $A = \mathrm{Id}$, $Z = X_n$, to achieve the first requirement, as done in [13], contradicts the second requirement, at least for complex inverse problems. In general, constructing an appropriate A is challenging and requires a priori knowledge of the underlying inverse problem, see Sections 4.1 and 4.2 for concrete examples from seismic imaging. In the first example, the operator A is only implicitly given, and $A = \mathrm{Id}$ is realized in the second.

Now, we will discuss the theoretical justification of this approach with regard to Assumption 2.1 and the resulting constraints on the neural network and the training process. For our analysis below we introduce the notation $E_G(x, z)$ for the linearization error of a Fréchet-differentiable operator $G\colon \mathcal{X} \to \mathcal{Y}$ in $x \in \mathcal{X}$ at $z \in \mathcal{X}$:

$$E_G(x, z) = G(x) - G(z) - G'(z)[x - z].$$

Under certain restrictions on the trained network and the forward operator F_n the misfit $d_{F(x^+)}$ locally satisfies Assumption 2.1.

Lemma 3.1. *Let $y = F_n(x^+)$ for $x^+ \in \mathrm{D}(F_n)$. Further, let both F_n and Φ_θ be Fréchet-differentiable with locally bounded derivatives, i.e., for some $L_F, L_\Phi > 0$ and $\rho > 0$ we have*

$$\|F_n'(x)\| \le L_F, \quad \|\Phi_\theta'(F_n(x))\| \le L_\Phi, \quad x \in B_\rho(x^+) \subset \mathrm{D}(F_n). \tag{9}$$

Then, $d_y(\cdot) = \mathrm{dist}_\theta(y, \cdot)$ satisfies conditions (i), (iii), (iv), *and* (v) *of Assumption* 2.1 *locally in $B_\rho(x^+)$.*

Moreover, d_y is weakly convex on every line passing through x^+ in the following sense (compare, for instance, [4, Chap. 2]*):*

(ii') *The misfit d_y fulfills:*

$$d_y(x^+) - d_y(z) \ge d_y'(z)[x^+ - z] - C_z\|z - x^+\|_X^2, \quad z \in B_\rho(x^+),$$

where the constant $C_z > 0$ gets arbitrarily small for $z \to x^+$.

Proof. Conditions (i) and (iii) follow immediately by construction. Conditions (iv) and (v) are verified as follows.

(iv) For $x \in B_\rho(x^+)$ we have

$$d_y'(x) = F_n'(x)^* \left[\Phi_\theta'(F_n(x))^* \left[\Phi_\theta(F_n(x)) - \Phi_\theta(y)\right]\right]$$

and our assumptions yield

$$\|d_y'(x)\|^2 \le 2L_F L_\Phi d_y(x).$$

(v) The Fréchet-derivative of d_y in $y \in Y_n$ for $x \in \mathrm{D}(F_n)$ is

$$\partial_y \,\mathrm{dist}_\theta(y, x) = \Phi_\theta'(y)^* \left[\Phi_\theta(F_n(x)) - \Phi_\theta(y)\right].$$

Plugging in (y, x^+) we even get $\eta_y = \partial_y \,\mathrm{dist}_\theta(y, x^+) = 0$.

For the weak convexity (ii') let $z \in B_\rho(x^+)$ and conclude according to

$$\begin{aligned}
d_y(z) - d_y(x^+) - d_y'(z)[z - x^+] &= d_y(z) - d_y'(z)[z - x^+] \\
&= \frac{1}{2}\|\Phi_\theta(F_n(z)) - \Phi_\theta(y)\|_Z^2 - \left\langle \Phi_\theta(F_n(z)) - \Phi_\theta(y), (\Phi_\theta \circ F_n)'(z)\left[z - x^+\right]\right\rangle_Z \\
&= \frac{1}{2}\left\langle \Phi_\theta(F_n(z)) - \Phi_\theta(y), -E_{(\Phi_\theta \circ F)}(x^+, z)\right\rangle_Z \\
&\quad - \frac{1}{2}\left\langle \Phi_\theta(F_n(z)) - \Phi_\theta(y), (\Phi_\theta \circ F_n)'(z)\left[z - x^+\right]\right\rangle_Z \\
&= \frac{1}{2}\left\langle E_{(\Phi_\theta \circ F)}(x^+, z), E_{(\Phi_\theta \circ F)}(x^+, z)\right\rangle_Z \\
&\quad - \frac{1}{2}\left\langle (\Phi_\theta \circ F_n)'(z)\left[z - x^+\right], (\Phi_\theta \circ F_n)'(z)\left[z - x^+\right]\right\rangle_Z \\
&\le \frac{1}{2}\left\|E_{(\Phi_\theta \circ F)}(x^+, z)\right\|_Z^2 = \mathrm{o}(\|z - x^+\|_X^2),
\end{aligned}$$

which is the weak convexity. □

Remark 3.2. *Note that the constant C_z solely depends on the linearization error of $\Phi_\theta \circ F_n$, and as such, condition (8) is a natural requirement on the data converter to "convexify" the problem.*

The rigorous verification of (vi) must fail in this general setting as this condition is intertwined with the local conditionally well-posedness of the inverse problem (1). As such it depends strongly on the nonlinearity F and the discrete space X_n. However, if the discrete operator F_n is locally injective, and if the operator A, the architecture of the neural network and its training are designed carefully for X_n, we can reasonably expect

$$C\|x - x^+\|_X \le \|\Phi_\theta(F_n(x)) - \Phi_\theta(F_n(x^+))\|_Z, \quad x \in B_\rho(x^+) \subset \mathrm{D}(F_n), \tag{10}$$

to hold for one $C > 0$ and one $\rho > 0$. In other words, we have a local version of (vi) for d_y as in Lemma 3.1. In fact, the intention of the whole training process is to establish the above Lipschitz well-posedness of the composition $\Phi_\theta \circ F_n$ locally in a large neighborhood of any $x^+ \in \mathrm{D}(F_n)$. The local Lipschitz stability (10) holds, for instance, if the Fréchet-derivative $(\Phi_\theta \circ F)'(x^+)\colon X_n \to Z$ is injective, see [16, Lem. C.1(a)] (a similar result can be found in [11]). In Appendix A we provide a local convergence result for Algorithm 1 under (10) and assumptions that $d_y(\cdot) = \mathrm{dist}_\theta(y, \cdot)$ satisfies (Lemma 3.1).

Under a slightly stronger hypothesis on $(\Phi_\theta \circ F)'$ we can even verify uniform convexity of d_y. To this end, let $(\Phi_\theta \circ F)'(x)$ be uniformly injective with respect to x, that is, there exists a constant $\lambda_{\min} > 0$ such that

$$\lambda_{\min}\|h\|_X \le \|(\Phi_\theta \circ F)'(x)h\|_Z \quad \text{for all } x \in B_\rho(x^+) \text{ and for all } h \in X_n. \tag{11}$$

This injectivity guarantees the existence of some $L < 1$ and a possibly smaller $\rho > 0$ such that

$$\left\|E_{(\Phi_\theta \circ F)}(x, z)\right\|_Z \le L\,\|(\Phi_\theta \circ F)'(z)[x - z]\|_Z \quad \text{for all } x, z \in B_\rho(x^+), \tag{12}$$

see [16]. The estimate (12) is known under the name *tangential cone condition.*

Lemma 3.3. *Let both (11) and (12) be fulfilled. Then, the misfit $d_y(\cdot) = \mathrm{dist}_\theta(y, \cdot)$ is uniformly convex in $B_\rho(x^+)$. Moreover, items* (ii) *and* (vi) *from Assumption 2.1 are satisfied.*

Proof. As in the proof of the previous lemma,

$$d_y(z) - d_y(x^+) - d'_y(z)[z - x^+] = \frac{1}{2}\left\|E_{(\Phi_\theta \circ F)}(x^+, z)\right\|^2 \\ - \frac{1}{2}\left\|(\Phi_\theta \circ F_n)'(z)\left[z - x^+\right]\right\|^2.$$

Using (11) and (12) we estimate

$$d_y(z) - d_y(x^+) - d'_y(z)[z - x^+] \leq \frac{(L^2 - 1)\lambda_{\min}^2}{2}\left\|z - x^+\right\|^2,$$

which rearranges to uniform convexity since $L < 1$. □

We emphasize that we still get (local) strict convexity only under (12).

Our previous analysis indicates that, if F' is locally injective, the training process should mimic the constrained optimization problem

$$\min_{\theta \in \mathbb{R}^p} \frac{1}{2}\left\|E_{\Phi_\theta \circ F}(x^+, z)\right\|_Z^2, \quad \lambda_{\min}\left(\Phi'_\theta(F(z))^* \Phi'_\theta(F(z))\right) \geq c \quad \text{for all } z \in X_n, \tag{13}$$

for some $c > 0$ where $\lambda_{\min}(M)$ denotes the smallest eigenvalue of the symmetric matrix M. Furthermore, x^+ is the desired solution, which is typically unknown. Thus, a more sophisticated approach is required to train a conversion network, as we explain below. We stress the fact that the above constraint yields (11) since F' is supposed to be locally injective.

3.1.1. *Training.* Our goal is to train a neural net $\Phi_\theta\colon Y_n \to Z$ with parameters $\theta \in \mathbb{R}^p$ satisfying

$$\Phi_\theta \circ F_n(x) \approx Ax$$

for a simple operator $A\colon X_n \to Z$. This operator must be chosen such that the trained distance functional $\mathrm{dist}_\theta(\cdot, y)$ is convex, for example, $A = \mathrm{Id}$ and $Z = X_n$. However, flexibility in choosing A, and hence Z, is allowed to incorporate a priori knowledge of the underlying inverse problem and/or to reduce the training effort. We provide a concrete example in the context of seismic imaging in Section 4 below.

We restrict ourselves to the case where $Y_n = (\widetilde{Y_n})^N$ and $Z = \widetilde{Z}^N$ for some finite dimensional spaces $\widetilde{Y_n}$ and $\widetilde{Z}$. For instance, this setting for Y_n reflects the case where the observations are N finite time series, i.e., $\widetilde{Y_n} \subset L^2(0, T)$. Moreover, $\widetilde{Z}$ represents the (possibly high-dimensional) output layer of the neural network where the (now convex) distance will be evaluated. Thus, Z and $\widetilde{Z}$ can be understood as some sort of encoding spaces, where the decoding layer is the distance evaluation. In this setting, the resulting misfit functional d_y can be written as a sum,

$$d_y(x) = \sum_{i=1}^{N} \widetilde{d}_{y_i}(x),$$

where $\widetilde{d}$ denotes a distance on the lower-dimensional space and can therefore be parameterized by a smaller neural network. Decomposing the measurement $y = (y_1, \ldots, y_N) \in Y_n$ also increases the number of effective training samples, even when only a limited number of measurements y is available in practice.

Thus instead of training a network Φ_θ on the large space $(\widetilde{Y_n})^N$, we train a network $\widetilde{\Phi_\theta}$ on the smaller space $\widetilde{Y_n}$ such that

$$\widetilde{\Phi_\theta}(F_{n,i}(x)) \approx (Ax)_i, \quad \text{for all} \quad i \in \{1, \dots, N\},$$

Where $F_{n,i}$ denotes the i-th component of the fully discrete forward operator F_n. With this approach the whole convexifier $\Phi_\theta \colon Y_n \to Z$ is build componentwise by

$$\Phi_\theta(y) = \begin{pmatrix} \widetilde{\Phi_\theta}(y_1) \\ \vdots \\ \widetilde{\Phi_\theta}(y_N) \end{pmatrix}. \tag{14}$$

Then, the new distance Δ between two measurements $y^1, y^2 \in Y_n$ is

$$\Delta(y^1, y^2) = \frac{1}{2} \left\| \Phi_\theta(y^1) - \Phi_\theta(y^2) \right\|_Z^2 = \frac{1}{2} \sum_{i=1}^N \left\| \widetilde{\Phi_\theta}(y_i^1) - \widetilde{\Phi_\theta}(y_i^2) \right\|_{\widetilde{Z}}^2 =: \frac{1}{2} \sum_{i=1}^N d_\theta(y_i^1, y_i^2).$$

Here, $d_\theta \colon \widetilde{Y_n} \times \widetilde{Y_n} \to [0, \infty)$ denotes the learned distance between two given measurements and thus serves as our training objective. In the best case this would yield

$$d_\theta(F_{n,i}(x^1), F_{n,i}(x^2)) \approx \left\| (Ax^1)_i - (Ax^2)_i \right\|_{\widetilde{Z}}^2.$$

Since in general full knowledge of A, Z and the parameters x is not possible, we generate a total of $N_{\text{data}} \in \mathbb{N}$ training triplets $\{(y_k^1, y_k^2, \tau_k)\}_{k=1,\dots,N_{\text{data}}} \in (\widetilde{Y_n} \times \widetilde{Y_n} \times \mathbb{R})^{N_{\text{data}}}$, where the pairs (y_k^1, y_k^2) replace $(F_{n,i_k}(x_k^1), F_{n,i_k}(x_k^2))$ for some unknown $x_k^1, x_k^2 \in \mathrm{D}(F_n)$ and an $i_k \in \{1, \dots, N\}$. Further, the τ_k's should satisfy

$$\tau_k \approx \left\| (Ax_k^1)_{i_k} - (Ax_k^2)_{i_k} \right\|_{\widetilde{Z}}^2.$$

In this way, A is only known implicitly. However, should A be known explicitly, then one can generate the data triplets using test samples from X_n.

With $\{(y_k^1, y_k^2, \tau_k)\}_{k=1,\dots,N_{\text{data}}}$ we finally train the neural net by minimizing the L^1 loss functional

$$l(\theta) = \frac{1}{N_{\text{data}}} \sum_{k=1}^{N_{\text{data}}} \left| d_\theta(y_k^1, y_k^2) - \tau_k \right|.$$

Remark 3.4. *We sketch a way to generate the training pairs (y_k^1, y_k^2) from measurements. Suppose that the nonlinearity F satisfies approximately an invariance property $F(Rx) \approx SF(x)$ with operators $R \colon X \to X$ and $S \colon Y \to Y$. Then, we can choose y_k^1 as our measurement for $F_{n,i_k}(x_k^1)$ and set $y_k^2 := Sy_k^1$. Observe that R is not required explicitly.*

We apply this approach in Section 4.1.

Now, to achieve the constraint on the Jacobian, see (13), we restrict our net to be of the form $\widetilde{\Phi_\theta} \colon \widetilde{Y_n} \to \widetilde{Z} := \widetilde{Y_n}$,

$$\widetilde{\Phi_\theta}(y) = y + \mathrm{MLP}_\theta(y), \tag{15}$$

where MLP_θ is a standard multi-layer-perceptron[1] with the LeakyReLU[2] as activation function. Assuming $\|\mathrm{MLP}'_\theta(y)\|_1 < 1$ and employing Gershgorin's circle theorem, we can

[1] This is a standard, fully connected, feed-forward network with at least one inner layer of neurons, see, e.g., [39, Chap. 2.1].

[2] see e.g. [39, Chap. 2.2.3].

estimate the smallest absolute value of the eigenvalues of the Jacobian from below by

$$\rho_{\min}(\widetilde{\Phi_\theta'}(y)) = \rho_{\min}(I + \mathrm{MLP}_\theta'(y)) \geq 1 - \|\mathrm{MLP}_\theta'(y)\|_1 \geq 1 - \prod_{W \in \mathrm{weights}(\mathrm{MLP}_\theta)} \|W\|_1 .$$

For the last inequality we used that for a standard MLP given as a concatenation of activation functions and affine linear transformations,

$$\mathrm{MLP}_\theta = \sigma \circ \mathcal{A}_L \circ \cdots \circ \sigma \circ \mathcal{A}_1,$$

with $\mathcal{A}_i(y) = W_i y + b_i$ the Jacobian is given as a multiplication of the weights and Jacobians of the activation function:

$$\mathrm{MLP}_\theta'(y) = \sigma'(\dots)W_L \sigma'(\dots)W_{L-1} \cdots \sigma'(\dots)W_1.$$

Since in our case σ is the LeakyReLU the Jacobians $\sigma'(\dots)$ are diagonal matrices with diagonal entries being either 1 or $a < 1$ (the negative leakage parameter) and thus $\|\sigma'(\dots)\|_1 \leq 1$. Now an application of the submultiplicativity property for matrix norms yields the above estimate for the smallest eigenvalue.

If we restrict the weights in the parameter space such that $\|W\|_1 \leq C < 1$, then both $\widetilde{\Phi_\theta'}(y)$ and $\widetilde{\Phi_\theta'}(y)^* \widetilde{\Phi_\theta'}(y)$ are injective. To enforce this constraint, we project the weights after each training step onto the closed, convex 1-norm ball of radius C. That is, if $W^k = (W_1^k, \dots, W_L^k)$ and $b^k = (b_1^k, \dots, b_L^k)$ denote the weights and biases at the k-th training iteration, and if the (unprojected) update map is denoted by $\mathcal{U}$, then one projected training step consists of the following two stages:

$$\begin{aligned} (W^{k+1/2}, b^{k+1}) &= \mathcal{U}(W^k, b^k), \\ W_i^{k+1} &= P_C^1(W_i^{k+1/2}) \quad \text{for all } i \in \{1, \dots, L\}, \end{aligned}$$

where P_C^1 denotes the projection onto $\left\{W \in \mathcal{L}(\widetilde{Y_n}, \widetilde{Y_n})\colon \|W\|_1 \leq C\right\}$, as described above.

3.2. **Distance networks.** Another way to train a distance functional is to directly train a neural network $\Delta\colon Y_n \to [0, \infty)$, such that $d_y(x) := \Delta(F_n(x) - y)$ is a convex functional.

In this case we usually need stronger assumptions on the architecture of d_y to obtain a useful optimization objective. If we assume that d_y is similar to an MLP and the non-negativity constraint is achieved by taking the square of the output, our functional is of the form

$$d_y(x) = \left(b^\top \Phi_\theta(F_n(x) - y)\right)^2, \tag{16}$$

where $\Phi_\theta\colon Y \to Z$ is a neural net and $(b, \theta) \in Z \times \mathbb{R}^p$ is the parameter vector which we optimize during the training process.

In this distance concept we obtain a result similar to Lemma 3.1.

Lemma 3.5. *Let $y = F_n(x^+)$ for $x^+ \in \mathrm{D}(F_n)$. Further, let both F_n and Φ_θ be Fréchet-differentiable satisfying* (9). *Additionally, let* 0 *be a zero of Φ_θ ($\Phi_\theta(0) = 0$). Then d_y from* (16) *satisfies conditions* (i), (iii), (iv), *and* (v) *of Assumption 2.1 locally in $B_\rho(x^+)$. Moreover, d_y is weakly convex on every line passing through x^+, that is,* (ii') *as formulated in Lemma 3.1 holds as well.*

Proof. Let $z \in X_n$. Again (i) is obvious with

$$d_y'(z) = 2\left(b^\top \Phi_\theta(F_n(z) - y)\right) F'(z)^* \Phi_\theta'(F(z) - y)^*[b].$$

Using $\Phi_\theta(0) = 0$ we establish (iii). Item (iv) is fulfilled since

$$\left\| d_y'(z) \right\|_{X_n}^2 \leq 2L_\Phi^2 L_F^2 \left\| b \right\|_Z^2 d_y(z),$$

and (v) holds where $\eta_y = 0$.

It remains to validate (ii'). We have

$$\begin{aligned}
&d_y(z) - d_y(x^+) - d_y'(z)[z - x^+] \\
&\quad= \left(b^\top \Phi_\theta(F_n(z) - y)\right)^2 - 2\left(b^\top \Phi_\theta(F_n(z) - y)\right) \left\langle b, \Phi_\theta'(F_n(z) - y)F'(z)[z - x^+]\right\rangle \\
&\quad= \left(b^\top \Phi_\theta(F_n(z) - y)\right) \left\langle b, \Phi_\theta(F_n(z) - y) - 2\Phi_\theta'(F_n(z) - y)F'(z)[z - x^+]\right\rangle \\
&\quad= -2\left(b^\top \Phi_\theta(F_n(z) - y)\right) \left(b^\top E_{(\Phi_\theta(\cdot - y))\circ F_n}(x^+, z)\right) - d_y(z) \\
&\quad\leq -2\left(b^\top \Phi_\theta(F_n(z) - y)\right) \left(b^\top E_{(\Phi_\theta(\cdot - y))\circ F_n}(x^+, z)\right).
\end{aligned}$$

The claim follows now directly from $\left(b^\top \Phi_\theta(F_n(x) - y)\right) = O(\|z - x^+\|_{X_n})$ and the definition of the Fréchet-derivative. □

We guarantee the condition $\Phi_\theta(0) = 0$ by designing the net Φ_θ without biases. Then, training can be conducted as was done for the data converter by learning a smaller network $\widetilde{\Phi_\theta}$ on the smaller subspace $\widetilde{Y_n}$: With the training samples $\{(y_k^1, y_k^2, \tau_k)\}_{k=1,\dots,N_{\text{data}}} \in (\widetilde{Y_n} \times \widetilde{Y_n} \times \mathbb{R})^{N_{\text{data}}}$ introduced in Section 3.1.1, we minimize the L^1-loss

$$l(\widetilde{b}, \theta) = \frac{1}{N_{\text{data}}} \sum_{k=1}^{N_{\text{data}}} \left| \left(\widetilde{b}^\top \widetilde{\Phi_\theta}(y_k^1 - y_k^2)\right)^2 - \tau_k \right|$$

to obtain the optimal parameters (b^*, θ^*). Finally, the full distance functional is

$$d_y(x) = \sum_{i=1}^{N} \left((\widetilde{b^*})^\top \widetilde{\Phi_{\theta^*}}(F_{n,i}(x) - y_i) \right)^2.$$

The corresponding $b \in Z$ is composed as $b = (\widetilde{b}, \dots, \widetilde{b}) \in \widetilde{Z}^N = Z$ and $\Phi_\theta \colon Y_n \to Z$ is again the componentwise application on the product space $Y_n = \widetilde{Y_n}^N$.

Remark 3.6. *Under an additional assumption on b and Φ_θ, which is hard to verify, we can ensure the conditional well-posedness from Assumption 2.1. Suppose, there is a constant $c > 0$ such that*

$$\frac{\left|b^\top z\right|}{\|b\|_Z \, \|z\|_Z} \geq c \quad \text{for any } z \in \Phi_\theta(Y_n) \setminus \{0\},$$

that is, the angle between z and b is uniformly smaller than $\pi/2$. Then, for any $x \neq x^+$ using $\Phi_\theta(0) = 0$ we get

$$\begin{aligned}
d_y(x) = \left(b^\top \Phi_\theta(F_n(x) - y)\right)^2 &\geq c\, \|b\|_Z^2 \, \|\Phi_\theta(F_n(x) - y)\|_Z^2 \\
&= c\, \|b\|_Z^2 \, \|\Phi_\theta(F_n(x) - y) - \Phi_\theta(0)\|_Z^2.
\end{aligned}$$

If we now assume the Fréchet-derivative F_n' is locally injective in a ball $B_\rho(x^+) \subset \mathrm{D}(F_n)$ and Φ_θ' is locally injective in a ball $B_r(0) \subset Y_n$, we obtain the conditional well-posedness locally in a ball about x^+, see [16, Lem. C.1(a)].

4. Full waveform inversion in the acoustic regime

One well-known inverse problem where the non-convexity of the objective functional becomes an issue arises in time domain full waveform inversion (FWI) due to the phenomenon of cycle-skipping. FWI is the most advanced seismic imaging methodology to reconstruct the interior structure of the Earth from reflected wave fields. Here, cycle-skipping is caused by a phase mismatch between the measured wave fields (seismograms) and reconstructed wave fields [44]. When this mismatch exceeds half a period, optimization procedures using L^2-distances will become trapped in a local minimum, unless the starting guess is close to the ground truth.

From a mathematical point of view, FWI entails a parameter identification task for the underlying wave propagation model, that is, the used wave equation. For our numerical experiments, we rely on the acoustic wave equation with constant bulk density in a bounded Lipschitz-domain $\Omega \subset \mathbb{R}^2$. In this setting the forward operator $F = \mathrm{M}_n \mathcal{F}$ consists of the following two building blocks:

- The parameter-to-state map

$$\mathcal{F} \colon \mathsf{D}(\mathcal{F}) \subset L^\infty(\Omega, \mathbb{R}) \to L^2([0,T], L^2(\Omega)), \quad \nu \mapsto u,$$

  which maps a pressure wave velocity

$$\nu \in \mathsf{D}(\mathcal{F}) := \left\{ w \in L^\infty(\Omega, \mathbb{R}) : \nu_{\min} \leq w \leq \nu_{\max} \text{ a.e.} \right\}$$

  to the solution $u \colon [0,T] \to L^2(\Omega)$ of the acoustic wave equation,

$$\frac{1}{\nu^2} \partial_t^2 u - \Delta_x^2 u = f, \quad u(0) = 0, \ \partial_t u(0) = 0. \tag{17}$$

  Here, $0 < \nu_{\min} < \nu_{\max}$ are physically meaningful constants and $f \in L^2([0,T], L^2(\Omega))$ is the source term. If f is sufficiently regular in time and the boundary restriction $u|_\Omega = 0$ is satisfied, then (17) admits a unique weak solution. Thus, $\mathcal{F}$ is well defined and moreover Fréchet-differentiable, see [29].
- The measurement operator

$$\mathrm{M}_n \colon L^2([0,T], L^2(\Omega)) \to Y_n = \mathbb{R}^{N_T \times N_S}, \quad u \mapsto s,$$

  which maps a wave field u to the observed seismograms s at N_S receiver locations and N_T points in time.

As a benchmark model to compare the proposed distance functionals with the standard L^2-objective, we use the established Camembert model, also known as the circular inclusion model [22]. The resulting inverse problem is highly nonlinear and susceptible to cycle-skipping.

In our example we choose $\Omega = [-35, 35]^2$ and the sought-after velocity is given as

$$\nu_C(x) = \begin{cases} 120, & |x| \leq 20, \\ 100, & \text{else}, \end{cases}, \quad x \in \Omega. \tag{18}$$

The velocity of the Camembert is 20% higher than the background velocity, which renders the inverse problem severely ill-posed. As discrete space X_n for the velocities we choose piecewise constant functions on a 301×301 equidistant grid in the square $[-35, 35]^2$. So, $\dim X_n = 90601$ and the discretization step size in each coordinate direction is $h = 70/301 \approx 0.23$. We equip X_n with the Euclidean inner product with respect to the grid

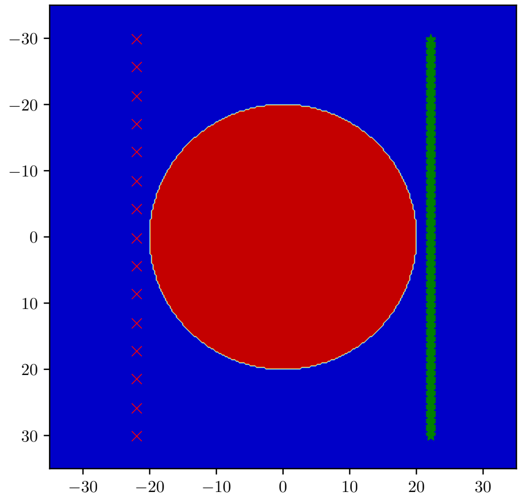


FIGURE 1. Test velocity (18) with source (red crosses) and receiver locations (green crosses).

values. Our fully discrete, Fréchet-differentiable forward operator between Hilbert spaces is then

$$F_n\colon \mathsf{D}(\mathcal{F}) \cap X_n \subset X_n \to Y_n, \quad F_n(\nu) \coloneqq \mathrm{M}_n\mathcal{F}(\nu).$$

For our transmission experiment, we place 15 equidistant point sources left of the Camembert and measure the excited wave fields at $N_S = 200$ receiver points on the right; see Figure 1 for the geometric setting. As the number of points in time is $N_T = 400$ for each receiver in any of our examples below, we have $\dim Y_n = 80000$ throughout.

As source signal in time, we applied the Ricker wavelet

$$r(t) = \big(1 - 2\sigma^2(t)\big)\,\mathrm{e}^{-\sigma^2(t)}, \quad \sigma(t) = \pi\,\omega_{\mathrm{c}}\,(t - t_{\mathrm{s}}),$$

with central frequency $\omega_{\mathrm{c}} = 10\,\mathrm{Hz}$ and time-shift $t_{\mathrm{s}} = 0.15\,\mathrm{s}$. The complete source term for (17) is then

$$f(x,t) = \sum_{i=1}^{15} r(t)\delta_{x_i}^h(x), \tag{19}$$

where $\{x_i\colon i = 1,\dots,15\} \subset \Omega$ is the set of source positions. In addition, $\delta_{x_i}^h$ is the characteristic function of the square $x_i + [-h/2, h/2]^2$, normalized such that $\int_\Omega \delta_{x_i}^h(x)\mathrm{d}x = 1$. Here, h is the discretization step size associated with X_n. Moreover, $\operatorname{supp}\delta_{x_j}^h \cap \operatorname{supp}\delta_{x_i}^h = \emptyset$ for $i \neq j$. Please note that we perform a one-shot experiment. Only one source is fired at 15 spatially separated locations simultaneously.

In our numerical examples below, we used PyTorch [37] to implement neural nets Φ_θ with architectures specified in (14) and (15). Further, we used its auto-differentiation utilities to avoid the expensive adjoint state methods. We relied on the wave solver Deepwave [40] to compute the forward map F_n. Following the work [33], Deepwave implements an absorbing layer around Ω to prevent waves from propagating back into the domain after reaching the boundary.

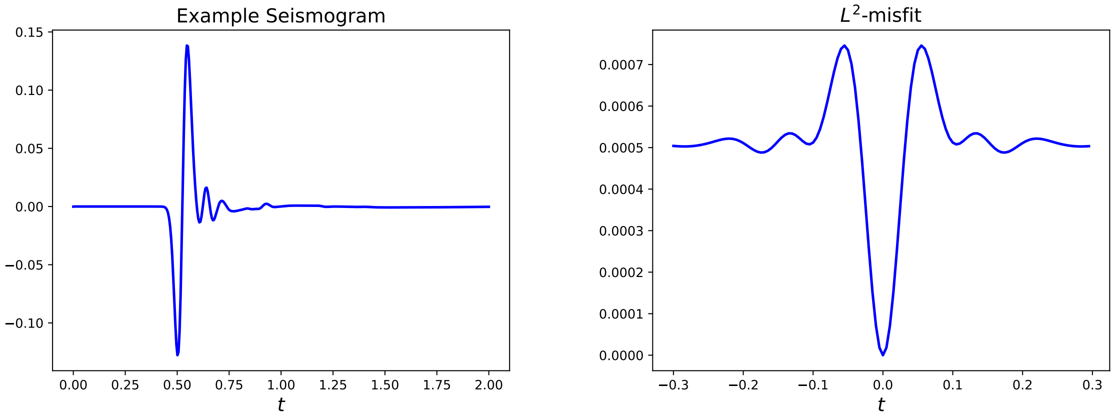


FIGURE 2. Example seismogram on the left with corresponding L^2-Misfit on the right.

The Python code for all of our experiments below and additional material can be downloaded from the GIT repository https://gitlab.kit.edu/david.haemmerling/learned-distance-functionals.

4.1. **Time-shift training data.** Within the seismic community, a common intuition for non-convexity of the standard L^2-objective $J_y(\nu) := \frac{1}{2}\,\|F(\nu) - y\|_{L^2}^2$ are time shifts in the seismic data. When the velocity ν changes, the arrival time of an emitted shot at receiver locations also changes, as well as the reflections of the signal, but not so much the shape of the signal itself. This phenomenon has already been described in the introduction of this section and is called cycle-skipping. It is illustrated in Figure 2 by a seismogram s measured in the Camembert experiment and the standard L^2-Misfit

$$t \mapsto \frac{1}{2}\,\|s(\cdot) - s(\cdot - t)\|_{L^2}^2 \tag{20}$$

as a function of time shifts (the norm above is the weighted Euclidean norm on $\mathbb{R}^{N_T}$ to approximate the indicated continuous norm). Motivated by this observation it has been the goal in the seismic community to construct a misfit functional $\Delta\colon L^2([0,T]) \times L^2([0,T]) \to [0,\infty)$ which is convex with respect to the time shifts (see e.g. [18, 34]). In the best case scenario such a misfit functional would be given as the normal parabola, i.e. for a given seismogram $s \in L^2([0,T])$, extended by 0 outside of $[0,T]$, we would like to have

$$\Delta(s, s(\cdot - t)) = t^2 \tag{21}$$

for any time shift $t \in [-T,T]$. This approach fits into the setting of Remark 3.4, when S is the shift-operator $S = S_t\colon L^2(\mathbb{R}) \to L^2(\mathbb{R})$, $S_t s(\cdot) = s(\cdot - t)$. Since a time shift of a seismogram corresponds to a change in wave speed, we postulate the existence of an operator $R_t\colon L^\infty(\Omega) \to L^\infty(\Omega)$ such that $S_t \circ \mathcal{F} = \mathcal{F} \circ R_t$. Accordingly, for any given seismogram s, we can generate a useful training triplet $(y^1, y^2, \tau) = (s, S_t s, t^2)$ for any t.

Now suppose we are given a collection of seismograms $y^\delta = (s_1, \ldots, s_{N_S}) \in Y_n$. In agreement with the discussion in the previous paragraph, our training samples are

$$\left\{(s_i, s_i(\cdot - t_l), t_l^2)\colon i = 1, \ldots, N_S,\ l = 1, \ldots, N_{\text{shift}}\right\} \subset \widetilde{Y_n} \times \widetilde{Y_n} \times \mathbb{R}$$

where $\widetilde{Y_n} = \mathbb{R}^{N_T} = \mathbb{R}^{400}$ and $t_1, \ldots, t_{N_{\text{shift}}} \in [-T,T]$. Thus, $N_{\text{data}} = N_S N_{\text{shift}}$. For all experiments, we set $N_{\text{shift}} = 60$ and randomly split the obtained dataset into training and

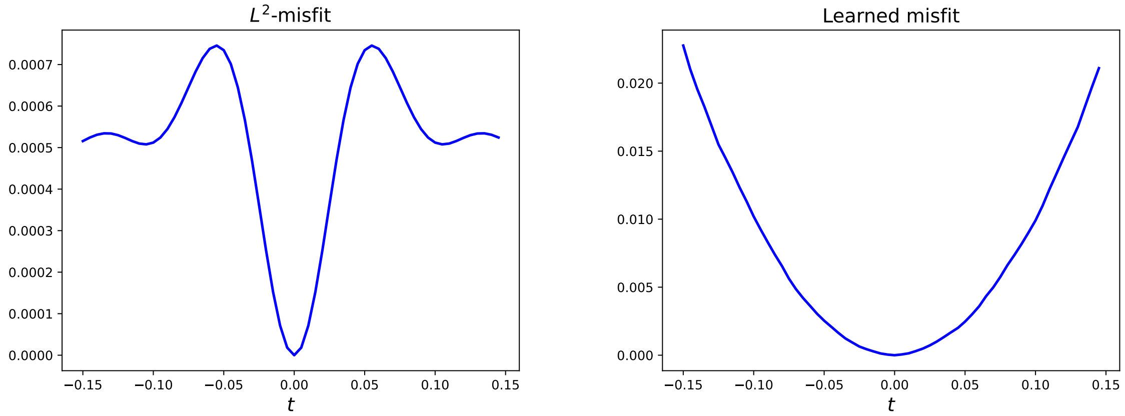


FIGURE 3. Comparison of the L^2-Misfit from (20) and the learned misfit from (21).

validation sets with a size ratio $9 : 1$ to avoid overfitting. All of the following experiments were run on an NVIDIA RTX A4000 GPU.

For our first experiment we rely on consistent data $y := F_n(P_n \nu_C)$ where the value of $P_n \nu_C \in X_n$ on a grid cell coincides with the value of ν_C at the cell's midpoint. When plotting the standard L^2-misfit for one of the seismograms "measured" at the receivers, we observe a non-convex L^2-misfit with local minima, which potentially give rise to the cycle-skipping phenomenon, see Figure 3 (left).

However, after training a data transformer with an architecture as described in Section 3.1.1 consisting of 25 linear layers with a total of 554000 parameters, the misfit functional loses its local extrema, see Figure 3 (right). Since we work with consistent data in a rather low-dimensional discretization, regularization is a minor issue in this experiment. Instead of the Landweber iteration, we therefore use the faster L-BFGS method from [20] for minimizing the L^2- and the learned misfit, starting with the constant background velocity $\nu_0 \equiv 100$, which we also use as initial velocity in the other experiments below. The outcome after 100 iterations can be viewed in Figure 4. Note that the reconstruction on the left is near a local minimum. Earlier iterates with the same starting value do not fundamentally differ in shape and quality from this result. A similar observation is reported in [23, Fig. 1].

We emphasize that all displayed reconstructions are clipped to values in the range $[100, 120]$ to show more detail.[3] So, some reconstructions may appear better than they actually are. Therefore, to quantitatively compare the different reconstructions, we define an error measure relative to the starting guess by

$$\operatorname{err}(\nu) := \frac{\|\nu - P_n \nu_C\|_{X_n}}{\|\nu_0 - P_n \nu_C\|_{X_n}}.$$

For the reconstruction ν_{L^2}, obtained by minimizing the L^2-misfit (Figure 4, left), we have $\operatorname{err}(\nu_{L^2}) \approx 1.53$. In other words, its error is about 53% worse than that of the initial guess! On the other hand, the velocity ν_Φ, fitted with the learned functional, has a relative error of only $\operatorname{err}(\nu_\Phi) \approx 0.61$.

[3] Instead of the reconstruction ν_{rec}, we plot $\max\{\min\{\nu_{\mathrm{rec}}, 120\}, 100\}$.

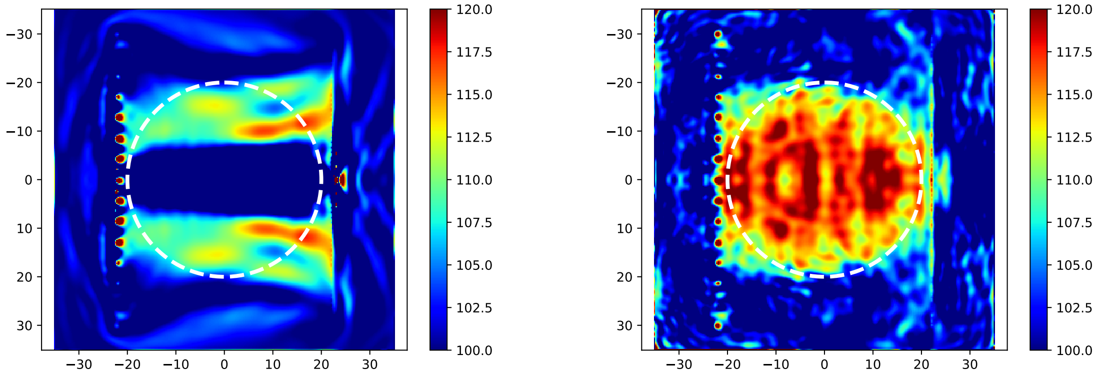


FIGURE 4. Reconstructions after 100 iterations of L-BFGS. Left: Reconstruction ν_{L^2} using the standard L^2-misfit with $\mathrm{err}(\nu_{L^2}) \approx 1.53$. Right: Reconstruction ν_Φ using the learned misfit Δ (21) with $\mathrm{err}(\nu_\Phi) \approx 0.61$.

In the previous example (Figure 4) we committed an inverse crime: We generated the seismogram with the same wave solver that we used in the L-BFGS minimization procedure. Now, we are training the same network layout based on inconsistent data that we generated using our own finite difference time domain solver. Additionally, we have perturbed the seismograms by 1% relative L^2-noise. Figure 5 displays the resulting misfit functional for one seismogram and the reconstruction after 5000 Landweber iterations[4]. This reconstruction is overfitted as indicated by the error plot in Figure 6 (left, blue curve), which reveals the typical semi-convergence of iterative regularization schemes applied to ill-posed problems, see, e.g., [17, 41]: The error first decreases and then increases after an optimal stopping index (iteration number) is reached. Here, the optimal stopping index is 1140 and the corresponding iterate is shown in the right of Figure 6. Please observe that the error decreases strictly monotonically up to the optimal stopping index, which is in full agreement with Theorem 2.4.

Employing the faster L-BFGS scheme to minimize the learned distance yields the best reconstruction with respect to the L^2-norm after only 49 iterations, see Figure 7 (right). However, its error $\mathrm{err}(\nu_{49}^{\mathrm{BFGS}}) \approx 0.62$ is larger then $\mathrm{err}(\nu_{1140}^{\mathrm{Land}}) \approx 0.58$ for the best Landweber reconstruction. Furthermore, as the error evolution in Figure 7 (left) reveals, the error does not decrease monotonically up to its minimal value at ν_{49}^{BFGS}. Hence, it is not straightforward to carry over the regularization theory for the Landweber method from Section 2 to the L-BFGS scheme, if it is even possible.

Remark 4.1. *A comment on computer time consumption is in order. To set up the neural net used for the reconstructions in Figure 4, we minimized the loss functional with the Adam optimizer over* 200 *training epochs and selected the set of parameters with the lowest validation loss. This process took about* 4:39 min[5]. *The* 100 L-BFGS *iterations to generate the reconstructions ν_{L^2} and ν_Φ required* 2:56 min *and* 2:20 min, *respectively.*

[4] We worked with a constant step size of 50000, which we also used for the reconstruction shown in Figure 6. The step sizes for the reconstructions in Figures 8 and 9 are 2500000 and 200, respectively. These large values compensate for the rather small magnitude of the source (19), that is, c_y in Assumption 2.1 (iv) can be large, and so the step size as well, see the hypothesis on the step sizes in Theorem 2.4.

[5] Training times for the other networks in this section are similar.

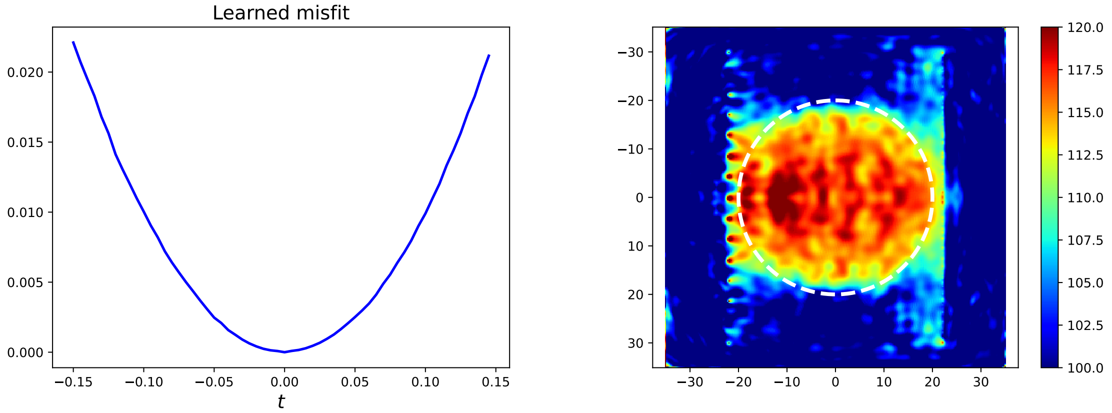


FIGURE 5. Left: Learned misfit Δ (21) for one seismogram based on data generated without inverse crime (inconsistent data). Right: Corresponding Reconstruction ν_{5000}^{Land} after 5000 Landweber iterations with $\text{err}(\nu_{5000}^{\text{Land}}) \approx 0.61$.

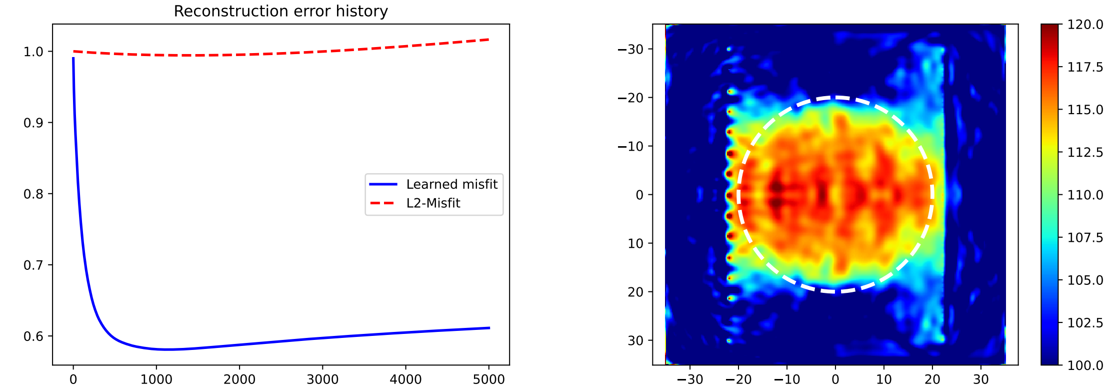


FIGURE 6. Left: Relative L^2-error of the Landweber iteration as a function of the iteration index for the learned (blue) and the L^2-misfit (red dashed). Right: Optimal reconstruction, i.e., the Landweber iterate based on the learned misfit after 1140 iterations which has the least error $\text{err}(\nu_{1140}^{\text{Land}}) \approx 0.58$.

Finally, we compare the runtime for computing the optimal reconstructions on display in Figures 6 *and* 7*. The* 1140 *Landweber iterations took* 19:57 min*, whereas the* 49 L-BFGS *iterations finished in only* 1:15 min.

With the modified Armijo rule for step size selection in the Landweber method, the optimal reconstruction is obtained in 725 *iterations, consuming* 18:28 min *of* GPU *time, while retaining the accuracy of the reconstruction in Figure* 6*. The more sophisticated Fletcher-Reeves (nonlinear conjugate gradients) scheme together with the Armijo rule delivered its optimal result in* 1:27 min *after* 49 *iterations with an error of* 0.64*. For more details, see the above-referenced* GIT *repository.*

For a further experiment, we trained the distance network from (16) relying on the same training data as above and obtain similar results for data with and without inverse crimes. Figure 8 (left) shows the L-BFGS reconstruction ν_{50}^{BFGS} based on consistent data yielding $\text{err}(\nu_{50}^{\text{BFGS}}) \approx 0.68$. Here, the stopping rule of the L-BFGS implementation

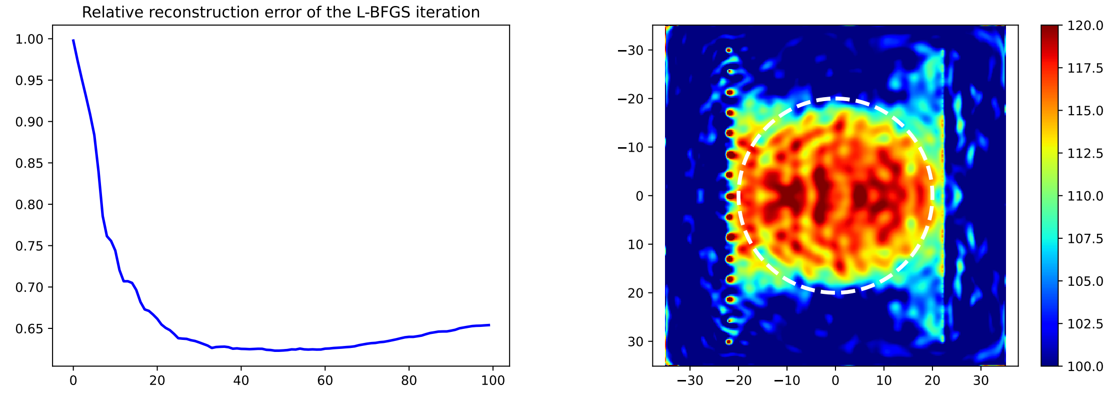


FIGURE 7. L-BFGS error history (left) minimizing the learned misfit and corresponding optimal reconstruction (right) after 49 iterations with $\mathrm{err}(\nu_{49}^{\mathrm{BFGS}}) \approx 0.62$.

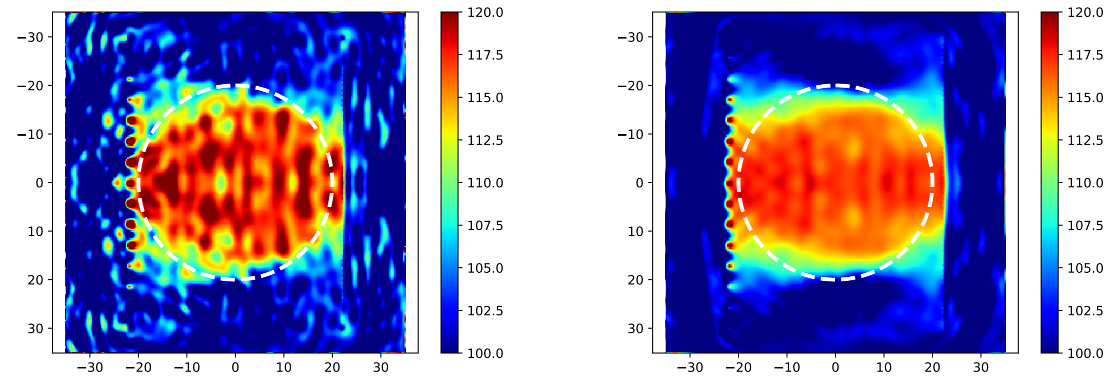


FIGURE 8. Reconstructions obtained from minimizing the direct misfit (16). Left: L-BFGS result ν_{50}^{BFGS} for consistent data with $\mathrm{err}(\nu_{50}^{\mathrm{BFGS}}) \approx 0.68$. Right: Landweber result $\nu_{1224}^{\mathrm{Land}}$ for inconsistent data with $\mathrm{err}(\nu_{1224}^{\mathrm{Land}}) \approx 0.52$.

from [20] terminated the algorithm after 50 iterations. On the right of this figure, the optimal Landweber iterate $\nu_{1224}^{\mathrm{Land}}$, using inconsistent data, is displayed which we picked by monitoring the error evolution. Its error is the lowest so far: $\mathrm{err}(\nu_{1224}^{\mathrm{Land}}) \approx 0.52$.

4.2. **Constant parameter training data.** Another way to set up a data converter is to make Φ_θ an approximate inverse of the discrete forward operator F_n, that is,

$$\Phi_\theta \circ F_n \approx A = \mathrm{Id},$$

see (8). To achieve this, we need to generate data samples $\big(\mu_i, F(\mu_i)\big)_{i=1,\dots,N_{\mathrm{param}}} \in (X_n \times Y_n)^{N_{\mathrm{param}}}$ for a total of N_{param} parameters μ_i in X_n. However since the dimension of X_n can be arbitrarily large (here, $\dim X_n = 90601$), it is nearly impossible to reflect all the small variations in the discrete materials by the training set. Therefore, we generate only a small number of constant parameters $\mu_i \in \mathbb{R}$ and the corresponding seismograms $F(\mu_i) = (y_{i,1}, \dots, y_{i,N_S})$, so that we can at least accomplish

$$(\Phi_\theta \circ F_n)|_{\mathcal{C}} \approx \mathrm{Id}\,|_{\mathcal{C}}$$

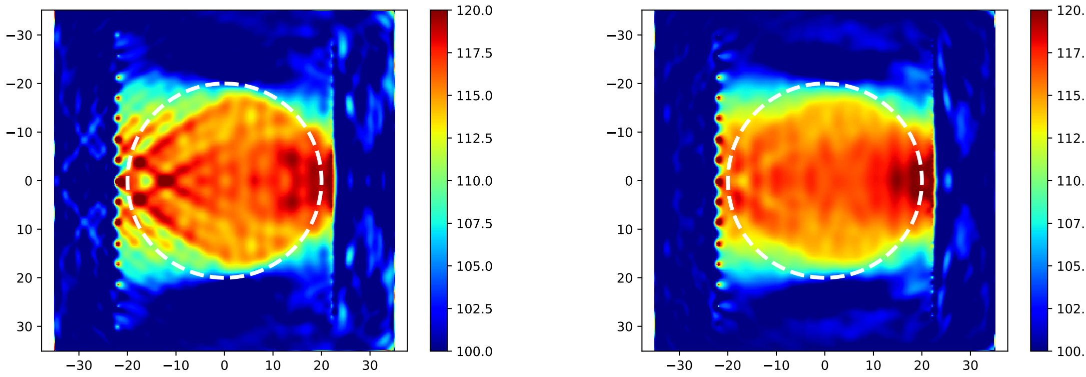


FIGURE 9. Reconstructions from constant velocity training data. Left: $\nu_{100}^{\mathrm{BFGS}}$ with $\mathrm{err}(\nu_{100}^{\mathrm{BFGS}}) \approx 0.53$, obtained after 100 iterations of L-BFGS using consistent data. Right: $\nu_{3131}^{\mathrm{Land}}$ with $\mathrm{err}(\nu_{3131}^{\mathrm{Land}}) \approx 0.53$, obtained after 3131 iterations of Landweber's method using inconsistent data.

for the subspace $\mathcal{C} \subset X_n$ of constant velocities. Then our training samples for this data converter, as described in Section 3.1, are

$$\big\{(y_{i,l}, y_{j,l}, (\mu_i - \mu_j)^2)\colon l = 1, \dots, N_S,\ i, j = 1, \dots, N_{\mathrm{param}}\big\} \subset \widetilde{Y_n} \times \widetilde{Y_n} \times \mathbb{R}.$$

This results in a total of $N_{\mathrm{data}} = N_{\mathrm{param}}^2 N_S$ training samples. After generating seismograms for the $N_{\mathrm{param}} = 13$ constant velocities $85, 90, 95, \dots, 145$, we train a neural net as described in Section 3.1.1 with $N_{\mathrm{data}} = 33800$ training samples, again randomly split into training and validation samples. The results are shown in Figure 9 where, on the left, the reconstruction $\nu_{100}^{\mathrm{BFGS}}$ from consistent data is the output of L-BFGS stopped after 100 steps, as further iterations decreased the error only insignificantly. We have $\mathrm{err}(\nu_{100}^{\mathrm{BFGS}}) \approx 0.53$. The image on the right is $\nu_{3131}^{\mathrm{Land}}$, which is the best outcome of the Landweber method from inconsistent data with respect to the error measure, yielding $\mathrm{err}(\nu_{3131}^{\mathrm{Land}}) \approx 0.53$.

5. CONCLUSION AND OUTLOOK

First, in Section 2, we established a regularization theory for the Landweber iteration applied to general convex distance functionals. Next, in Section 3 (and also in Appendix A), we theoretically justified the use of neural networks to construct such convex functionals. We also presented an application-driven training strategy to stabilize the training of these networks.

Finally, in Section 4 we investigated the proposed approach numerically on the Camembert benchmark model. We found that the distance functionals can be constructed and optimized without excessive computational cost. Moreover, the learned misfit functionals can convexify the problem and yield reconstructions that outperform the standard formulation. We also tested a more common optimization method, namely L-BFGS, and observed no notable differences in the reconstructions; however, we did not observe the error monotonicity required by the theory in Section 2. Hence, additional modifications would be needed to prove that a standard L-BFGS scheme is a regularization method.

This leads directly to our outlook. From a theoretical perspective, we have so far established convergence for noisy data only for (a suitable) gradient descent scheme. It is therefore natural to investigate other optimization methods and assess under which conditions they can be interpreted as regularization schemes. From a machine learning perspective, alternative constructions of convex functionals and different network architectures merit further study, since our experiments were restricted to standard MLPs. In the context of FWI, the proposed methodology could be applied to additional benchmark settings and, in particular, to multiparameter reconstruction and to field data. In these scenarios, our approach of training a network using only the measured seismograms (to mitigate cycle-skipping) could be especially beneficial.

Appendix A. Local convergence for weakly convex functionals

In this appendix we will show that Algorithm 1 still converges and has the regularization property when the learned distance functional is only weakly convex instead of convex. To this end, we modify Assumption 2.1 to be consistent with the discussion and results in Subsection 3.1 (Lemma 3.1 and explanations given in the context of (10)).

Assumption A.1. *Let $y \in Y$. We assume that our misfit functional $d_y \colon X \to [0, \infty)$ matches the following conditions:*

(i) *The misfit functional is Fréchet-differentiable at any $x \in X$ with Riesz-representation $d'_y(x) \in X$ such that*

$$\partial_x d_y(x)[h] = \langle d'_y(x), h \rangle_X .$$

(ii) *The misfit d_y is weakly convex in a ball $B_\rho(x^+)$ in the following sense:*

$$d_y(x^+) - d_y(z) \geq d'_y(z)[x^+ - z] - \beta_y \|z - x^+\|_X^2 \quad \text{for all } z \in B_\rho(x^+)$$

and for some constant $\beta_y > 0$.

(iii) *It holds that*

$$d_{F(x)}(x) = 0 \quad \text{for all } x \in \mathrm{D}(F).$$

(iv) *There is a constant $c_y > 0$ such that*

$$c_y \left\| d'_y(x) \right\|_X^2 \leq d_y(x) \quad \text{for all } x \in B_\rho(x^+)$$

(v) *There exists a functional $d \colon Y \times X \to [0, \infty)$ such that $d_y(\cdot) = d(y, \cdot)$ and which is Fréchet-differentiable with respect to y at $(F(x^+), x^+)$. Set*

$$\eta_y := \left\| \partial_y d(F(x^+), x^+) \right\|_{Y \to \mathbb{R}} .$$

(vi) *There is a constant $\zeta_y > 0$ such that*

$$\zeta_y \, \|x - x^+\|_X^2 \leq d_y(x) \quad \text{for all } x \in B_\rho(x^+)$$

with $\zeta_y > 2\beta_y$.

Under these assumptions we can again prove local convergence for starting values in $B_\rho(x^+)$ in the noise-free setting.

Theorem A.2. *Under Assumption A.1 consider Algorithm 1 with input $y = F(x^+)$ for $x^+ \in \mathrm{D}(F)$, $\delta = 0$, and $\{\lambda_n\} \subset [\lambda_{\min}, \lambda_{\max}]^{\mathbb{N}_0}$ with $0 < \lambda_{\min} \leq \lambda_{\max} < 2c_y(1 - \beta_y/\zeta_y)$. Further, assume that $\zeta_y - \sqrt{\zeta_y/c_y} < \beta_y < \zeta_y$. Then, for $x_0 \in B_\rho(x^+)$, the Landweber sequence*

$$x_{n+1} := x_n - \lambda_n d'_y(x_n), \quad n \in \mathbb{N}_0,$$

generated by Algorithm 1 either stops after a finite number of iterations with x^+, or monotonically converges to x^+: $\lim_{n\to\infty} x_n = x^+$ and

$$\|x_{n+1} - x^+\|_X \leq \|x_n - x^+\|_X. \tag{22}$$

Proof. For $x_n \in B_\rho(x^+)$, we estimate

$$\begin{aligned}\|x_{n+1} - x^+\|_X^2 &= \|x_n - \lambda_n d_y'(x_n) - x^+\|_X^2 \\ &= \|x_n - x^+\|_X^2 - 2\lambda_n \langle d_y'(x_n), x_n - x^+\rangle_X + \lambda_n^2 \|d_y'(x_n)\|_X^2 \\ &\leq \|x_n - x^+\|_X^2 + 2\lambda_n \big(\underbrace{d_y(x^+)}_{=0} - d_y(x_n) + \beta_y \|x_n - x^+\|_X^2\big) + \frac{\lambda_n^2}{c_y} d_y(x_n) \\ &\leq \|x_n - x^+\|_X^2 - \lambda_n \left(2 - 2\frac{\beta_y}{\zeta_y} - \frac{\lambda_n}{c_y}\right) d_y(x_n)\end{aligned}$$

where we used Assumption A.1(vi) for the last bound. By our hypotheses, $2\frac{\beta_y}{\zeta_y} + \frac{\lambda_n}{c_y} < 2$ which immediately implies (22). A further application of (vi)

$$\|x_{n+1} - x^+\|_X^2 \leq \left(1 - \lambda_n \zeta_y \left(2 - 2\frac{\beta_y}{\zeta_y} - \frac{\lambda_n}{c_y}\right)\right) \|x_n - x^+\|_X^2 \leq (1 - \Lambda) \|x_n - x^+\|_X^2$$

for

$$0 < \Lambda = \zeta_y \min\left\{\lambda \left(2 - 2\frac{\beta_y}{\zeta_y} - \frac{\lambda}{c_y}\right) : \lambda \in \{\lambda_{\min}, \lambda_{\max}\}\right\} < 1.$$

The claimed convergence follows now inductively. □

The error monotonicity and termination follow as in Section 2.

Theorem A.3. *Let $y = F(x^+)$ and let $y^\delta \in Y$ such that $0 < \|y^\delta - y\|_Y \leq \delta$ for all $\delta > 0$ sufficiently small. Let both, d_y and d_{y^δ}, fulfill Assumption 2.1. Call Algorithm 1 with input y^δ, δ, $x_0^\delta \in B_{\rho_{y^\delta}}(x^+)$, and step sizes $\{\lambda_n\} \subset (0, \lambda_{\max}]^{\mathbb{N}_0}$ with $0 < \lambda_{\max} < 2c_{y^\delta}(1 - 2\beta_{y^\delta}/\zeta_{y^\delta})$ which satisfy the left equation of (3). Here, c_{y^δ}, ρ_{y^δ}, η_y, and ζ_{y^δ} are as in* (iv), (v), and (vi) *of Assumption A.1, respectively. If*

$$\tau > \frac{2\Big(1 + 2\beta_{y^\delta}/\zeta_{y^\delta}\Big)\eta_y}{\Big(2 - 4\beta_{y^\delta}/\zeta_{y^\delta} - \lambda_{\max}/c_{y^\delta}\Big)}$$

then, for $\delta > 0$ sufficiently small, Algorithm 1 stops after a finite number $N(\delta)$ of iteration steps and the iterates satisfy

$$\|x_{n+1}^\delta - x^+\|_X \leq \|x_n^\delta - x^+\|_X \quad \textit{for all } n < N(\delta).$$

Proof. Let $x_n^\delta \in B_{\rho_{y^\delta}}(x^+)$. As in the previous proof we start with

$$\|x_{n+1}^\delta - x^+\|_X^2 \leq \|x_n^\delta - x^+\|_X^2 + 2\lambda_n^\delta \big(d_{y^\delta}(x^+) - d_{y^\delta}(x_n^\delta) + \beta_{y^\delta} \|x_n^\delta - x^+\|_X^2\big) + \frac{(\lambda_n^\delta)^2}{c_{y^\delta}} d_{y^\delta}(x_n^\delta).$$

Let x^δ be the unique minimizer of d_{y^δ} according to (ii) of Assumption A.1. By (vi),

$$\|x_n^\delta - x^+\|^2 \leq 2\big(\|x_n^\delta - x^\delta\|_X^2 + \|x^\delta - x^+\|_X^2\big) \leq \frac{2}{\zeta_{y^\delta}}\big(d_{y^\delta}(x_n^\delta) + d_{y^\delta}(x^+)\big),$$

so that

$$\|x_{n+1}^\delta - x^+\|_X^2 \le \|x_n^\delta - x^+\|_X^2 + \lambda_n^\delta \Bigg(2\Big(1 + \frac{2\beta_{y^\delta}}{\zeta_{y^\delta}}\Big) d_{y^\delta}(x^+) - \Big(2 - \frac{4\beta_{y^\delta}}{\zeta_{y^\delta}} - \frac{\lambda_n^\delta}{c_{y^\delta}}\Big) d_{y^\delta}(x_n^\delta) \Bigg).$$

The factor in front of $d_{y^\delta}(x_n^\delta)$ is positive. Hence, if $d_{y^\delta}(x_n^\delta) > \tau\delta$ then

$$\|x_{n+1}^\delta - x^+\|_X^2 \le \|x_n^\delta - x^+\|_X^2 + \lambda_n^\delta \Bigg(2\Big(1 + \frac{2\beta_{y^\delta}}{\zeta_{y^\delta}}\Big) d_{y^\delta}(x^+) - \Big(2 - \frac{4\beta_{y^\delta}}{\zeta_{y^\delta}} - \frac{\lambda_n^\delta}{c_{y^\delta}}\Big) \tau\delta \Bigg).$$

We proceed with

$$\begin{aligned} d_{y^\delta}(x^+) &= d_{y^\delta}(x^+) - d_y(x^+) - \partial_y d(y, x^+)[y^\delta - y] + \partial_y d(y, x^+)[y^\delta - y] \\ &\le \mathrm{o}(\delta) + \eta_y \delta \end{aligned}$$

yielding

$$\|x_{n+1}^\delta - x^+\|_X^2 \le \|x_n^\delta - x^+\|_X^2 - \lambda_n^\delta \Delta(\delta) \tag{23}$$

with

$$\Delta(\delta) = \mathrm{o}(\delta) + \Bigg(\Big(2 - \frac{4\beta_{y^\delta}}{\zeta_{y^\delta}} - \frac{\lambda_n^\delta}{c_{y^\delta}}\Big)\tau - 2\Big(1 + \frac{2\beta_{y^\delta}}{\zeta_{y^\delta}}\Big)\eta_y \Bigg)\delta.$$

By the choice of τ, $\Delta(\delta) > 0$ for $\delta > 0$ sufficiently small independent of n. From (23) we can finish exactly as in the proof of Theorem 2.4. □

The regularization property and the convergence order of Theorem 2.5 and Corollary 2.6, respectively, carry over to the present situation.

References

[1] G. Alessandrini, M. V. de Hoop, F. Faucher, R. Gaburro, and E. Sincich, *Inverse problem for the Helmholtz equation with Cauchy data: reconstruction with conditional well-posedness driven iterative regularization*, ESAIM Math. Model. Numer. Anal., 53 (2019), pp. 1005–1030, https://doi.org/10.1051/m2an/2019009.

[2] G. Alessandrini and S. Vessella, *Lipschitz stability for the inverse conductivity problem*, Adv. in Appl. Math., 35 (2005), pp. 207–241, https://doi.org/10.1016/j.aam.2004.12.002.

[3] M. Araya-Polo, J. Jennings, A. Adler, and T. Dahlke, *Deep-learning tomography*, The Leading Edge, 37 (2018), pp. 58–66, https://doi.org/10.1190/tle37010058.1.

[4] F. Atenas, C. Sagastizábal, P. J. S. Silva, and M. Solodov, *A unified analysis of descent sequences in weakly convex optimization, including convergence rates for bundle methods*, SIAM J. Optim., 33 (2023), pp. 89–115, https://doi.org/10.1137/21M1465445.

[5] V. Bacchelli and S. Vessella, *Lipschitz stability for a stationary 2D inverse problem with unknown polygonal boundary*, Inverse Problems, 22 (2006), pp. 1627–1658, https://doi.org/10.1088/0266-5611/22/5/007.

[6] L. Beilina, M. Cristofol, S. Li, and M. Yamamoto, *Lipschitz stability for an inverse hyperbolic problem of determining two coefficients by a finite number of observations*, Inverse Problems, 34 (2018), pp. 015001, 27, https://doi.org/10.1088/1361-6420/aa941d.

[7] E. Beretta, M. V. de Hoop, F. Faucher, and O. Scherzer, *Inverse boundary value problem for the Helmholtz equation: quantitative conditional Lipschitz stability estimates*, SIAM J. Math. Anal., 48 (2016), pp. 3962–3983, https://doi.org/10.1137/15M1043856.

[8] E. Beretta, M. V. de Hoop, E. Francini, S. Vessella, and J. Zhai, *Uniqueness and Lipschitz stability of an inverse boundary value problem for time-harmonic elastic waves*, Inverse Problems, 33 (2017), pp. 035013, 27, https://doi.org/10.1088/1361-6420/aa5bef.

[9] E. Beretta, M. V. de Hoop, and L. Qiu, *Lipschitz stability of an inverse boundary value problem for a Schrödinger-type equation*, SIAM J. Math. Anal., 45 (2013), pp. 679–699, https://doi.org/10.1137/120869201.

[10] L. Borcea, J. Garnier, A. V. Mamonov, and J. Zimmerling, *Waveform inversion via reduced order modeling*, Geophysics, 88 (2023), pp. R175–R191, https://doi.org/10.1190/geo2022-0070.1.

[11] L. Bourgeois, *A remark on Lipschitz stability for inverse problems*, C. R. Math. Acad. Sci. Paris, 351 (2013), pp. 187–190, https://doi.org/10.1016/j.crma.2013.04.004.

[12] C. Deng, S. Feng, H. Wang, X. Zhang, P. Jin, Y. Feng, Q. Zeng, Y. Chen, and Y. Lin, *OpenFWI: Large-scale multi-structural benchmark datasets for seismic full waveform inversion*, 2023, https://arxiv.org/abs/2111.02926, https://arxiv.org/abs/2111.02926.

[13] W. Ding, K. Ren, and L. X. Zhang, *Coupling deep learning with full waveform inversion*, ArXiv, abs/2203.01799 (2022), https://api.semanticscholar.org/CorpusID:247222862.

[14] W. Dörfler, M. Hochbruck, J. Köhler, A. Rieder, R. Schnaubelt, and C. Wieners, *Wave phenomena—mathematical analysis and numerical approximation*, vol. 49 of Oberwolfach Seminars, Birkhäuser/Springer, Cham, 2023, https://doi.org/10.1007/978-3-031-05793-9.

[15] M. Eller, R. Griesmaier, and A. Rieder, *Tangential cone condition for the full waveform forward operator in the viscoelastic regime: the nonlocal case*, SIAM J. Appl. Math., 84 (2024), pp. 412–432, https://doi.org/10.1137/23M1551845.

[16] M. Eller and A. Rieder, *Tangential cone condition and Lipschitz stability for the full waveform forward operator in the acoustic regime*, Inverse Problems, 37 (2021), pp. Paper No. 085011, 17, https://doi.org/10.1088/1361-6420/ac11c5.

[17] H. W. Engl, M. Hanke, and A. Neubauer, *Regularization of inverse problems*, vol. 375 of Mathematics and its Applications, Kluwer Academic Publishers Group, Dordrecht, 1996.

[18] B. Engquist, B. D. Froese, and Y. Yang, *Optimal transport for seismic full waveform inversion*, 2016, https://arxiv.org/abs/1602.01540.

[19] B. Engquist and Y. Yang, *Optimal transport based seismic inversion: beyond cycle skipping*, Comm. Pure Appl. Math., 75 (2022), pp. 2201–2244, https://doi.org/10.1002/cpa.21990.

[20] R. Feinman, *Pytorch-minimize: a library for numerical optimization with autograd*, 2021, https://github.com/rfeinman/pytorch-minimize.

[21] A. Fichtner, *Full seismic waveform modelling and inversion*, Advances in Geophysical and Environmental Mechanics and Mathematics, Springer-Verlag Berlin Heidelberg, 2011, https://doi.org/10.1007/978-3-642-15807-0.

[22] O. Gauthier, J. Virieux, and A. Tarantola, *Two-dimensional nonlinear inversion of seismic waveforms: Numerical results*, Geophysics, 51 (1986), pp. 1387–1403, https://doi.org/10.1190/1.1442188.

[23] A. Gholami, H. S. Aghamiry, and S. Operto, *Extended-space full-waveform inversion in the time domain with the augmented Lagrangian method*, Geophysics, 87 (2021), https://doi.org/10.1190/geo2021-0186.1.

[24] A. Górszczyk, R. Brossier, and L. Métivier, *Graph-space optimal transport concept for time-domain full-waveform inversion of ocean-bottom seismometer data: Nankai trough velocity structure reconstructed from a 1D model*, Journal of Geophysical Research: Solid Earth, 126 (2021), p. e2020JB021504, https://doi.org/10.1029/2020JB021504.

[25] M. Hanke, A. Neubauer, and O. Scherzer, *A convergence analysis of the Landweber iteration for nonlinear ill-posed problems*, Numerische Mathematik, 72 (1995), pp. 21–37, https://doi.org/10.1007/s002110050158.

[26] B. Harrach, *Uniqueness and Lipschitz stability in electrical impedance tomography with finitely many electrodes*, Inverse Problems, 35 (2019), pp. 024005, 19, https://doi.org/10.1088/1361-6420/aaf6fc.

[27] B. Hofmann and O. Scherzer, *Local ill-posedness and source conditions of operator equations in Hilbert spaces*, Inverse Problems, 14 (1998), pp. 1189–1206, https://doi.org/10.1088/0266-5611/14/5/007.

[28] I. Häggström, C. R. Schmidtlein, G. Campanella, and T. J. Fuchs, *DeepPET: A deep encoder–decoder network for directly solving the PET image reconstruction inverse problem*, Medical Image Analysis, 54 (2019), pp. 253–262, https://doi.org/10.1016/j.media.2019.03.013.

[29] A. Kirsch and A. Rieder, *On the linearization of operators related to the full waveform inversion in seismology*, Math. Methods Appl. Sci., 37 (2014), pp. 2995–3007, http://dx.doi.org/10.1002/mma.3037.

[30] A. Lechleiter and A. Rieder, *Newton regularizations for impedance tomography: convergence by local injectivity*, Inverse Problems, 24 (2008), pp. 065009, 18, https://doi.org/10.1088/0266-5611/24/6/065009.

[31] H. Li, J. Schwab, S. Antholzer, and M. Haltmeier, *NETT: solving inverse problems with deep neural networks*, Inverse Problems, 36 (2020), p. 065005, https://doi.org/10.1088/1361-6420/ab6d57.

[32] M. Li, X.-s. Yan, and M.-z. Zhang, *A comprehensive review of seismic inversion based on neural networks*, Earth Science Informatics, 16 (2023), pp. 2991–3021, https://doi.org/10.1007/s12145-023-01079-4.

[33] R. Martin, D. Komatitsch, and S. D. Gedney, *A variational formulation of a stabilized unsplit convolutional perfectly matched layer for the isotropic or anisotropic seismic wave equation*, Computer Modeling in Engineering & Sciences, 37 (2008), pp. 274–304, http://www.techscience.com/CMES/v37n3/26752.

[34] J. Messud, R. Poncet, and G. Lambaré, *Optimal transport in full-waveform inversion: analysis and practice of the multidimensional Kantorovich–Rubinstein norm*, Inverse Problems, 37 (2021), p. 065012, https://doi.org/10.1088/1361-6420/abfb4c.

[35] L. Métivier, A. Allain, R. Brassier, Q. Mérigot, E. Oudet, and J. Virieux, *A graph-space optimal transport approach for full waveform inversion*, in 80th EAGE Conference and Exhibition 2018, vol. 2018, European Association of Geoscientists & Engineers, 2018, pp. 1–5, https://doi.org/10.1190/segam2018-2997001.1.

[36] L. Métivier, R. Brossier, F. Kpadonou, J. Messud, and A. Pladys, *A review of the use of optimal transport distances for high resolution seismic imaging based on the full waveform*, 2022, https://arxiv.org/abs/2204.08514.

[37] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, *Pytorch: An imperative style, high-performance deep learning library*, 2019, https://arxiv.org/abs/1912.01703.

[38] H. Qi, Y. Li, Z. Fu, and B. Han, *Deep prior sparse representation for full waveform inversion*, Inverse Problems, 42 (2026), p. 025009, https://doi.org/10.1088/1361-6420/ae3637.

[39] D. Ray, O. Pinti, and A. A. Oberai, *Deep Learning and Computational Physics*, Springer Cham, 2024, https://doi.org/10.1007/978-3-031-59345-1.

[40] A. Richardson, *Deepwave*, Sept. 2023, https://doi.org/10.5281/zenodo.8381177.

[41] A. Rieder, *Keine Probleme mit inversen Problemen*, Friedr. Vieweg & Sohn, Braunschweig, 2003, https://doi.org/10.1007/978-3-322-80234-7.

[42] A. Rüland and E. Sincich, *Lipschitz stability for the finite dimensional fractional Calderón problem with finite Cauchy data*, Inverse Probl. Imaging, 13 (2019), pp. 1023–1044, https://doi.org/10.3934/ipi.2019046.

[43] B. Sun and T. Alkhalifah, *ML-misfit: A neural network formulation of the misfit function for full-waveform inversion*, Frontiers in Earth Science, Volume 10 - 2022 (2022), https://doi.org/10.3389/feart.2022.1011825.

[44] J. Virieux and S. Operto, *An overview of full-waveform inversion in exploration geophysics*, Geophysics, 74 (2009), pp. WCC1–WCC26, https://doi.org/10.1190/1.3238367.

[45] Y. Wu and Y. Lin, *InversionNet: An efficient and accurate data-driven full waveform inversion*, IEEE Transactions on Computational Imaging, 6 (2020), pp. 419–433, https://doi.org/10.1109/TCI.2019.2956866.

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

[47] L. Zhu, W. Lu, M. Soleimani, Z. Li, and M. Zhang, *Electrical impedance tomography guided by digital twins and deep learning for lung monitoring*, IEEE Transactions on Instrumentation and Measurement, 72 (2023), pp. 1–9, https://doi.org/10.1109/TIM.2023.3298389.