EFILN: The Electric Field Inversion-Localization Network for High-Precision Underwater Positioning

Yimian Ding[*,+], Jingzehua Xu[*,+], Guanwen Xie[*], Haoyu Wang[†], Weiyi Liu[*], Yi Li[*]
[*]Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen, 518055, China
[†]Ocean College, Zhejiang University, Hangzhou, 316021, China
Email: liyi@sz.tsinghua.edu.cn

Abstract—Accurate underwater target localization is essential for underwater exploration. To improve accuracy and efficiency in complex underwater environments, we propose the Electric Field Inversion-Localization Network (EFILN), a deep feedforward neural network that reconstructs position coordinates from underwater electric field signals. By assessing whether the neural network's input-output values satisfy the Coulomb law, the error between the network's inversion solution and the equation's exact solution can be determined. The Adam optimizer was employed first, followed by the L-BFGS optimizer, to progressively improve the output precision of EFILN. A series of noise experiments demonstrated the robustness and practical utility of the proposed method, while small sample data experiments validated its strong small-sample learning (SSL) capabilities. To accelerate relevant research, we have made the codes available as open-source[1].

Index Terms—Underwater target localization, Electric field inversion, Deep learning, Neural network.

I. Introduction

IN recent years, precise and efficient underwater target localization has become a significant challenge [1]. Most research on electric field inversion and localization has focused on traditional methods [2], [3]. Lee *et al*. [4] proposed a real-time localization approach using a precomputed direct current (DC) electric field template. Ji *et al*. [5] simulated a moving dipole's electromagnetic field and introduced a Kalman filter-based localization algorithm. Yu *et al*. [6] used an unscented particle filter to track ship-generated electric fields. However, these methods suffer from lower accuracy and limited applicability.

With advances in computer performance and forward modeling, intelligent nonlinear localization methods have emerged. Unlike traditional approaches, these methods address nonlinear inversion without relying on initial models or Jacobian matrices [7]. Algorithms like simulated annealing [8], [9], particle swarm optimization [10], [11], and genetic algorithms [12] have been applied to electric field inversion. However, they often suffer from low accuracy and slow convergence with large datasets, limiting their effectiveness.

Recently, deep learning (DL) algorithms have attracted attention for their potential in electric signal inversion [13], [14]. Puzyrev *et al*. [15] used convolutional neural networks (CNNs) for two-dimensional (2D) inversion of controlled-source electromagnetic data [16]. Liu *et al*. [17] applied

CNNs for 2D inversion of electrical resistivity data, mapping resistivity to geoelectric models. These studies highlight DL's potential in signal inversion, but few have explored its use in electric signal localization.

Building on the above analysis, this paper proposes a novel deep learning neural network, termed the Electric Field Inversion-Localization Network (EFILN), specifically designed for high-precision and efficient electric field inversion localization. The proposed framework takes normalized three-directional underwater electric field components as inputs and outputs the corresponding position coordinates. By incorporating the physical principles of the electric field, governed by the Coulomb law, into the loss calculation, the output of EFILN gradually converges to the true values, achieving accurate electric field inversion localization. The primary contributions of this paper are summarized as follows

- To the best of our knowledge, this is the first study to employ the deep learning model for target localization by inverting electric field signals based on the Coulomb law. This technique elucidates the relationship between electric field signals and target position coordinates, facilitating precise localization.
- We employ a joint optimization approach using Adam and L-BFGS to update the neural network parameters. Adam rapidly reduces the loss during the initial training phase, allowing the model to quickly reach a promising parameter region. In the later stages, when the loss function approaches convergence, L-BFGS fine-tunes parameters with more accurate second-order information, further enhancing the model's performance. This approach improves both training efficiency and effectiveness.
- Extensive simulation experiments demonstrate that EFILN can achieve high-precision electric field inversion localization in environments with varying noise intensities, exhibiting strong robustness and generalizability. Additionally, EFILN performs well even under small sample learning (SSL) conditions.

The remainder of this paper is structured as follows. In Section II, we provide a comprehensive description of the system model. Section III presents the principles and details of the proposed algorithm, followed by the experimental results in Section IV. Finally, Section V concludes the paper and outlines potential directions for future research. Additionally,

---

| Symbols	Definition
$\mathbf{E}$	Electric field strength
Q_{enc}	The total charge
ρ	The charge density
ϵ_0	Vacuum permittivity
δ	The dirac delta function
V	The electric potential
q	The point charge
r	The distance from the point charge
E_x, E_y, E_z	Electric field components
(x_s, y_s, z_s)	The electric field source position
$Loss$	The loss function
α, β, γ	The weighting coefficients
b_l	The bias term
h_k	The hidden layer node
v_k	The visible layer node
σ_{kl}	The connection weight matrix
$\mathcal{F}_a(\cdot)$	The activation function
$\ddagger_l$	The hidden layer output
$\mathcal{D}$	The training set
$\mathcal{G}$	The gradient
$\mathcal{C}$	The number of samples
$\mathcal{B}_1, \mathcal{B}_2$	The exponential decay rates
$\mathcal{A}$	The first moment variables
ς	The second moment variables
H_k	The Hessian matrices from k^{th} iterations
I	The identity matrix
s_k	The change in network's parameter
g_k	The change in gradients
θ	The neural network's parameter
$u(E_X, E_y, E_z, \theta)$	The neural network
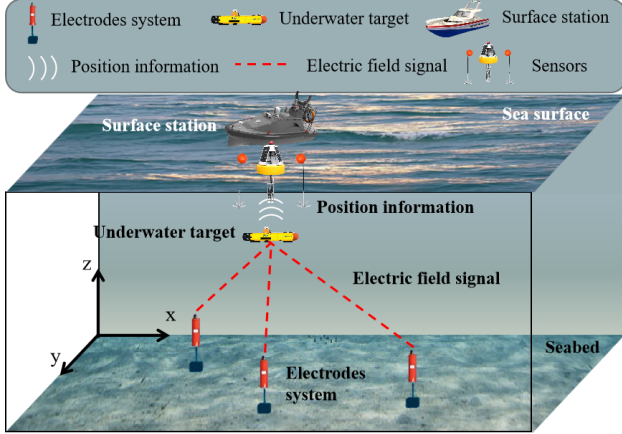
Fig. 1. Illustration of the underwater target localization scenario.

explanations of mainly used symbols are listed in Table I.

II. SYSTEM MODEL

In this section, we provide a detailed description of the underwater localization task and the principles of electric field inversion. The subsequent algorithm development will be based on these system models.

A. The Task Scenario of Underwater Localization

We consider an underwater target localization scenario, as illustrated in Fig. 1, where a target, denoted as T, is arbitrarily located within a 3D space defined by x, y, and z coordinates. Its position can be inferred through the inversion of electric field signals received by an electric field sensor network deployed on the target. Specifically, N electrode systems are deployed at various underwater locations, generating an electric field by applying a voltage between the electrodes. The target (such as an Autonomous Underwater Vehicle, or AUV) obtains the x, y, and z components of the electric field through onboard electric field sensors and inputs them into the EFILN localization system. EFILN performs inversion calculations on the input electric field signals and outputs the position coordinates of the target. These coordinates are then transmitted to a ground station via communication equipment, enabling the localization of the target.

B. Principles of Electric Field Localization

In electrostatics, Gauss's law from Maxwell's equations provides the relationship between the total charge enclosed within a closed surface and the integral of the electric flux over that surface:

$$\oint_{\partial V} \mathbf{E} \cdot d\mathbf{A} = \frac{Q_{\text{enc}}}{\epsilon_0}, \tag{1}$$

where $\mathbf{E}$ is the electric field strength, and $d\mathbf{A}$ is the differential area element vector, whose direction is aligned with the normal to the surface. ∂V is the closed surface in space V, Q_{enc} is the total charge enclosed by the surface, and ϵ_0 is the vacuum permittivity.

According to the Gauss-Ostrogradsky theorem, the electric flux through a closed surface can be related to the volume integral of the divergence of the electric field within that volume:

$$\oint_{\partial V} \mathbf{E} \cdot d\mathbf{A} = \int_V (\nabla \cdot \mathbf{E}) dV. \tag{2}$$

Meanwhile, the total charge Q_{enc} enclosed by a closed surface can be expressed as the volume integral of the charge density ρ:

$$Q_{\text{enc}} = \int_V \rho dV. \tag{3}$$

Substituting Equation (2) and Equation (3) into Equation (1) yields:

$$\int_V (\nabla \cdot \mathbf{E}) dV = \frac{1}{\epsilon_0} \int_V \rho dV, \tag{4}$$

thus

$$\frac{\partial E_x(x,y,z)}{\partial x} + \frac{\partial E_y(x,y,z)}{\partial y} + \frac{\partial E_z(x,y,z)}{\partial z} = \frac{\rho}{\epsilon_0}. \tag{5}$$

In this paper, since the electric field generated by the underwater electrode system can be approximated as a point charge, Equation (5) can be simplified. For a point charge q, its charge density ρ can be expressed using the Dirac δ function:

$$\rho = q\delta, \tag{6}$$

substituting Equation (6) into Equation (5) gives:

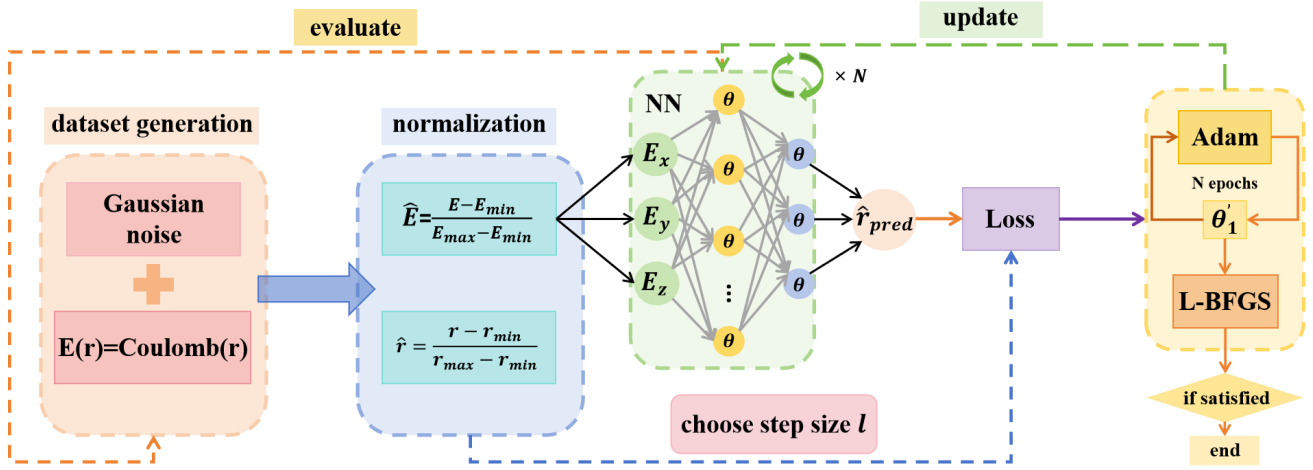$$\nabla^2 V = -\frac{q\delta}{\epsilon_0}, \tag{7}$$

Fig. 2. The overall architecture of our proposed EFILN algorithm.

where electric potential V satisfies the equation $\mathbf{E} = -\nabla V$. Since the electric potential V depends solely on the distance r from the point charge, it can be assumed that the potential exhibits spherical symmetry. In this case, Equation (7) can be simplified to:

$$\frac{1}{r^2}\frac{d}{dr}\left(r^2\frac{dV}{dr}\right) = -\frac{q\delta}{\epsilon_0}. \tag{8}$$

Based on the integral result from Gauss's law, the final solution for the electric potential distribution of a point charge is obtained as:

$$V = \frac{q}{4\pi\epsilon_0 r}. \tag{9}$$

Therefore, the corresponding relationship between the electric field distribution in space and the spatial coordinates is:

$$\mathbf{E}(r) = -\frac{dV}{dr}\hat{\mathbf{r}} = -\frac{d}{dr}\left(\frac{q}{4\pi\epsilon_0 r}\right)\hat{\mathbf{r}} = \frac{q}{4\pi\epsilon_0 r^2}\hat{\mathbf{r}}, \tag{10}$$

where $r = \sqrt{(x-x_s)^2 + (y-y_s)^2 + (z-z_s)^2}$, $\hat{\mathbf{r}} = (\frac{x-x_s}{r}, \frac{y-y_s}{r}, \frac{z-z_s}{r})$ is the unit vector from the spatial position (x, y, z) to the electric field source position (x_s, y_s, z_s). Equation (10) is the Coulomb law, which determines the electric field at a specific point based on spatial coordinates. However, inverting this equation using traditional numerical methods to derive spatial coordinates from electric field values is challenging. In this paper, we propose a neural network that takes electric field signals as input and outputs the corresponding spatial coordinates. The network's loss is calculated using Coulomb's law and iteratively optimized for high-precision localization. Specific details will be discussed in the next section.

III. ALGORITHM DESIGN

In this section, we first introduce the design of the algorithm's loss function, including how it is integrated with Coulomb law. Based on this, the principles and implementation details of the EFILN algorithm are presented, along with the corresponding pseudocode.

A. Loss Function Design

Assume that N electrode systems are deployed in the studied underwater area, with their coordinates given by $(x_s^i, y_s^i, z_s^i), i = 1, \ldots, N$. According to Equation (10), the electric field components in the three directions at the point (x, y, z) can be calculated as follows:

$$E_x = \sum_{i=1}^{N}\left(\frac{Q_i(x - x_s^i)}{4\pi\epsilon_0((x - x_s^i)^2 + (y - y_s^i)^2 + (z - z_s^i)^2)^{\frac{3}{2}}}\right), \tag{11}$$

$$E_y = \sum_{i=1}^{N}\left(\frac{Q_i(y - y_s^i)}{4\pi\epsilon_0((x - x_s^i)^2 + (y - y_s^i)^2 + (z - z_s^i)^2)^{\frac{3}{2}}}\right), \tag{12}$$

$$E_z = \sum_{i=1}^{N}\left(\frac{Q_i(z - z_s^i)}{4\pi\epsilon_0((x - x_s^i)^2 + (y - y_s^i)^2 + (z - z_s^i)^2)^{\frac{3}{2}}}\right), \tag{13}$$

where $Q_i(i = 1, \ldots, N)$ represents the charge of the ith electrode system.

Let $u(E_x, E_y, E_z, \theta)$ represents the neural network (NN) in the EFILN algorithm system. This network can take electric field signals as input and output spatial coordinates, thereby achieving electric field inversion localization. Assuming the dataset size is M, the loss in each of the three directions can be defined as follows:

$$Loss_x = \frac{1}{M}\sum_{j=1}^{M}(u_x(E_x^j, E_y^j, E_z^j, \theta) - x_j)^2, \tag{14}$$

$$Loss_y = \frac{1}{M}\sum_{j=1}^{M}(u_y(E_x^j, E_y^j, E_z^j, \theta) - y_j)^2, \tag{15}$$

$$Loss_z = \frac{1}{M}\sum_{j=1}^{M}(u_z(E_x^j, E_y^j, E_z^j, \theta) - z_j)^2, \tag{16}$$

where u_x, u_y, and u_z represent the x, y and z coordinates of the position vector output by the NN, respectively. x_j, y_j, z_j, E_x^j, E_y^j and E_z^j correspond to the spatial coordinates and electric field information of the jth point in the dataset, respectively. θ is the neural network's parameter.

Based on the above analysis, our loss function is designed as follows:

$$Loss = \alpha Loss_x + \beta Loss_y + \gamma Loss_z, \quad (17)$$

where α, β and γ are the weighting coefficients for the three components of the loss function.

B. The EFILN Algorithm

The framework of the EFILN algorithm is illustrated in Fig. 2. The challenge inherent in the localization algorithm resides in establishing a precise correspondence between the electric field signals and the spatial coordinates of the underwater target, culminating in the accurate derivation of said coordinates from the physical quantities. In this study, we utilize the NN to solve this challenge. Gaussian noise of a certain intensity is added to the dataset generated by the forward simulation using Coulomb law to better simulate real underwater environments. The noisy dataset is then normalized as follows:

$$\hat{\mathbf{X}} = \left(\frac{x - x_{min}}{x_{max} - x_{min}}, \frac{y - y_{min}}{y_{max} - y_{min}}, \frac{z - z_{min}}{z_{max} - z_{min}}\right), \quad (18)$$

$$\hat{\mathbf{E}} = \left(\frac{E_x - E_{xmin}}{E_{xmax} - E_{xmin}}, \frac{E_y - E_{ymin}}{E_{ymax} - E_{ymin}}, \frac{E_z - E_{zmin}}{E_{zmax} - E_{zmin}}\right), \quad (19)$$

where x_{min}, y_{min}, z_{min}, x_{max}, y_{max} and z_{max} represent the minimum and maximum values of x, y and z, respectively. The same applies to the parameters related to the electric field.

Then, the normalized input data undergoes a linear transformation through multiplication with weights and addition of a bias term b_l, yielding the layer's output. This process is mathematically represented as follows:

$$v_k = \sum_k \sigma_{kl} h_k + b_l, \ k = 1, \cdots, n; l = 1, \cdots, m, \quad (20)$$

where h_k denotes the hidden layer node ($0 < k \leq n$), v_k is the visible layer node ($0 < k \leq m$), and σ_{kl} represents the connection weight matrix between the visible layer and the hidden layer. The hidden layer output can be calculated as follows:

$$\ddagger_l = \mathcal{F}_a(\sum_k \sigma_{kl} h_k + b_l), \quad (21)$$

where $\mathcal{F}_a(\cdot)$ is the activation function. In this paper, the activation function is chosen to be tanh.

Given its fast convergence properties, we first utilize the Adam, an optimizer combining Momentum and RMSProp benefits. By sampling a minibatch of $\mathcal{J}$ examples from the training set $\{\mathcal{D}^{(1)}, \ldots, \mathcal{D}^{(\mathcal{J})}\}$ with corresponding targets $\mathcal{I}^{(\mathcal{V})}$, the gradient can be computed as

$$\mathcal{G} \leftarrow \frac{1}{C} \nabla_\theta \sum_{\mathcal{V}=1}^{\mathcal{J}} Loss(u(\mathcal{D}^{(\mathcal{V})}, \theta), \mathcal{I}^{(\mathcal{V})}), \quad (22)$$

Algorithm 1: The EFILN Algorithm

1 Initialize the neural network's parameter θ.

2 Sampling points at specified intervals $x_{step}, y_{step}, z_{step}$ and applying Coulomb law to generate the position-electric field dataset. Then normalized this dataset.

3 for *each epoch* n **do**

4   Input the electric field information from the dataset into the neural network and output the predicted position coordinates $(x_{pred}, y_{pred}, z_{pred})$.

5   Calculate the loss by Coulomb law:

$$Loss = \alpha Loss_x + \beta Loss_y + \gamma Loss_z. \quad (26)$$

6   Updated parameters by Adam algorithm.

7 end

8 Update parameters by L-BFGS until the gradient tolerance or change tolerance falls below the specified threshold.

9 Evaluate the effects of EFILN.

---

where $\mathcal{G}$ is the gradient and C is the number of samples in a batch.

Then, we update biased first and second moment estimate

$$\mathcal{A} \leftarrow \mathcal{B}_1 \mathcal{A} + (1 - \mathcal{B}_1 \mathcal{G}), \quad (23)$$

$$\varsigma \leftarrow \mathcal{B}_2 \varsigma + (1 - \mathcal{B}_2 \mathcal{G})\mathcal{G}, \quad (24)$$

where $\mathcal{B}_1$ and $\mathcal{B}_2$ are the exponential decay rates for moment estimates, and $\mathcal{A}$ and ς are the first and second moment variables, respectively. Hence, it is possible to rectify bias in both the first and second moments through the computation of the update. The process of achieving the gradient update entails the repetition of the aforementioned cycle.

After initial optimization with the Adam optimizer, we switched to the L-BFGS optimizer for fine-tuning. L-BFGS is a quasi-Newton method that updates parameters based on an approximation of the second-order derivative (Hessian matrix). It excels at making fine adjustments in the later training stages, helping the model reach local optima, and is also memory efficient. The update formula is as follows:

$$H_{k+1} = \left(I - \frac{s_k g_k^T}{g_k^T s_k}\right) H_k \left(I - \frac{g_k s_k^T}{g_k^T s_k}\right) + \frac{s_k s_k^T}{g_k^T s_k}, \quad (25)$$

where H_k and H_{k+1} represent the Hessian matrices from the kth and $(k+1)^{th}$ iterations, respectively. I is the identity matrix, $s_k = \theta_{k+1} - \theta_k$ denotes the change in network's parameter, and $g_k = \nabla u(\theta_{k+1}) - \nabla u(\theta_k)$ represents the change in gradients. L-BFGS limits memory usage by storing only the updates from the most recent iterations to construct an approximation of the Hessian matrix. Our optimization strategy, which first uses Adam followed by L-BFGS, combines the strengths of both: Adam's efficient exploration and L-BFGS's precise optimization. The pseudocode for EFILN is presented in Algorithm 1.
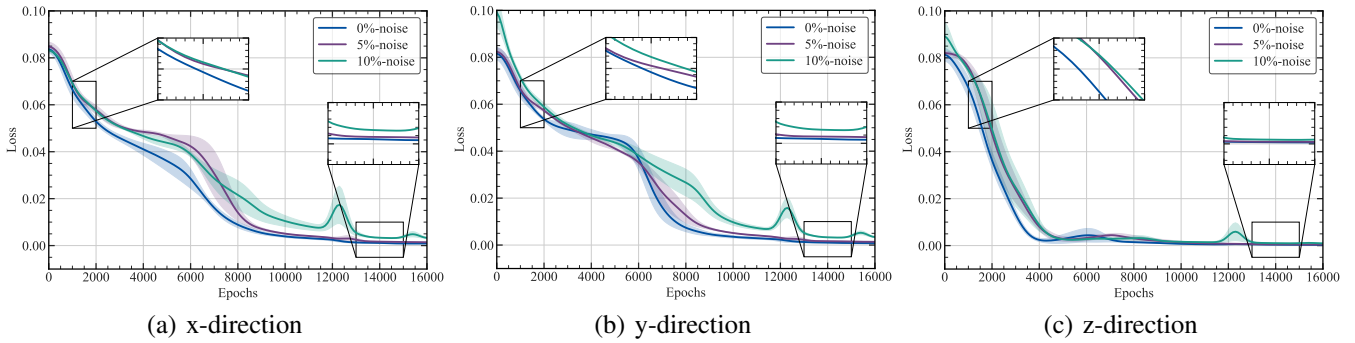
Fig. 3. Comparison of three-directional training loss under different noise levels: (a) x-direction's training loss (b) y-direction's training loss (c) z-direction's training loss.

TABLE II
PARAMETERS OF THE ENVIRONMENT AND FUSION FRAMEWORK.

Parameters	Values
Permittivity of vacuum ϵ_0	$8.854\times10^{-12}\ F/m$
x, y, z-direction underwater range	$[10m, 110m]$
Step size	$0.5\ m$
The number of electrode systems	2
The amount of charge on the electrode	$\pm1\ C$
The positions of the electrode systems	(0,0,0), (0,0,100)
The number of neurons in hidden layers	16
The number of hidden layers	8
The learning rate of Adam	0.0001
The learning rate of L-BFGS	10
The history size of L-BFGS	50
The gradient tolerance of the L-BFGS	1×10^{-12}
The maximum number of epochs	50000

TABLE III
COMPARISON OF THREE-DIRECTIONAL LOCALIZATION ACCURACY UNDER
DIFFERENT NOISE INTENSITIES.

	0%	5%	10%
x	$0.814m$	$1.013m$	$1.274m$
y	$0.908m$	$1.266m$	$0.970m$
z	$0.863m$	$0.730m$	$0.803m$

TABLE IV
LOCALIZATION ACCURACY COMPARISON ACROSS VARYING SAMPLING
INTERVALS.

	$0.5m$	$0.75m$	$1.0m$
x	$0.814m$	$1.102m$	$1.328m$
y	$0.908m$	$1.345m$	$1.651m$
z	$0.863m$	$1.182m$	$0.948m$

TABLE V
COMPARISON OF DIFFERENT LOCALIZATION ALGORITHMS.

	EFILN	RL- glmu	CSPLLS
location error	0.9%	2.0%	10.0%

IV. EXPERIMENTS AND RESULTS

In this section, we first present the experiment settings, including both environmental and algorithmic parameters. Subsequently, simulations under various conditions are conducted to validate the superior performance of EFILN.

A. Experiment Settings

The experimental code in this study was executed on a personal computer equipped with 13th Gen Intel® Core™ i7-13650HX processor and NVIDIA GeForce RTX 4060 Laptop GPU. The parameters can be divided into two parts: environment parameters and algorithm parameters, which are listed in Table II for summary.

B. Experiment Results

To better simulate the complexity of underwater environments and verify robustness, Gaussian noise with intensities of 5% and 10% was added to the dataset. The comparison of the three-directional loss under different noise conditions and the localization accuracy are shown in Fig. 3 and Table III, respectively. As noise increases, the dataset becomes more disrupted, leading to higher training loss. However, even in the presence of noise, the loss of EFILN still converges to a very low value, and it achieves high localization accuracy, confirming the robustness of EFILN.

Subsequently, small-sample tests were conducted to evaluate the SSL capability of EFILN. Starting with a sampling interval of $0.5m$, the step size was gradually increased by $0.25m$ during the training process, reaching a final value of $1.0m$. The experimental results are shown in Table IV. It can be observed that as the sampling interval increases, the positioning accuracy of EFILN decreases slightly. This is because the increase in step size leads to a significant reduction in the dataset size, thus weakening the neural network's fitting ability. However, even with a limited number of samples, EFILN still achieves high positioning accuracy, demonstrating the strong SSL capability of the algorithm, which meets the positioning accuracy requirements to some extent even with insufficient samples.

To visualize EFILN's high-precision localization capability, its performance was validated using spiral, circular and randomly selected spatial trajectories. To ensure rigorous validation, the selected points for the three types of trajectories do not overlap with any points in the dataset. The results are

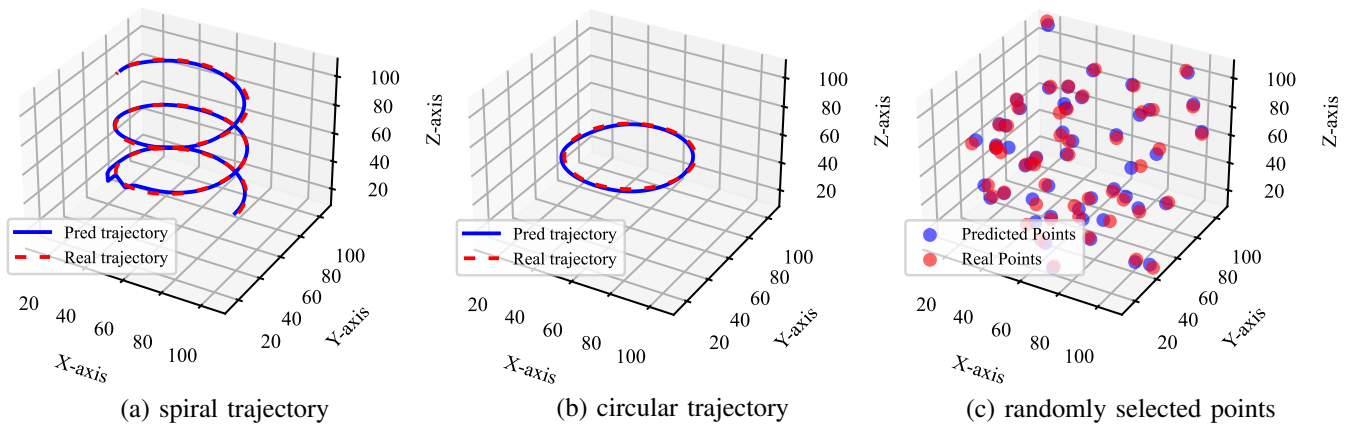(a) spiral trajectory (b) circular trajectory (c) randomly selected points

Fig. 4. Comparison of real and predicted trajectories under three different simulation paths: (a) spiral trajectory (b) circular trajectory (c) randomly selected points.

shown in Fig. 4. The predicted trajectories closely overlap with the real trajectories in all three cases, demonstrating the practical utility of EFILN.

Finally, under the conditions of the simulation experiments in this paper, the localization error of EFILN was compared with two other localization algorithms, RL-glmu [18] and CSPLLS [19]. The results, shown in Table V, indicate that EFILN's localization error is less than 50% and 10% of the other two methods, respectively, further demonstrating the superior performance of EFILN.

V. CONCLUSION

In this paper, we developed a deep neural network, EFILN, based on electric field signals for high-precision underwater localization. This network leverages the relationship between electric field signals generated by underwater electrode systems and spatial positions to invert the electric field data and accurately determine position coordinates. By employing a combination of the Adam and L-BFGS optimizers, the network's training performance is further enhanced. Extensive simulation experiments demonstrate that the network achieves high localization accuracy, exhibits strong robustness, and is well-suited for small-sample learning. Considering the complexity of underwater environments, future work will focus on validating real-world underwater field data.

REFERENCES

[1] R. A. Khalil, N. Saeed, M. I. Babar, and T. Jan, "Toward the internet of underwater things: Recent developments and future challenges," *IEEE Consumer Electronics Magazine*, vol. 10, no. 6, pp. 32–37, 2020.

[2] X. Pan, Y. Shen, and J. Zhang, "Iout based underwater target localization in the presence of time synchronization attacks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 6, pp. 3958–3973, 2021.

[3] R. Su, Z. Gong, C. Li, and X. Shen, "Algorithm design and performance analysis of target localization using mobile underwater acoustic array networks," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 2, pp. 2395–2406, 2022.

[4] H. Lee, H.-K. Jung, S.-H. Cho, Y. Kim, H. Rim, and S. K. Lee, "Real-time localization for underwater moving object using precalculated dc electric field template," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 10, pp. 5813–5823, 2018.

[5] J. Dou, S. Chao-long, W. Xiang-jun, and W. Zi-xia, "Research on electric field localization algorithm based on kalman filter," in *2015 Chinese Automation Congress (CAC)*, 2015, pp. 1485–1488.

[6] P. Yu, J. Cheng, and J. Zhang, "Ship target tracking using underwater electric field," *Progress In Electromagnetics Research M*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:214465529

[7] W. Liu, Z. Xi, H. Wang, and R. Zhang, "Two-dimensional deep learning inversion of magnetotelluric sounding data," *Journal of Geophysics and Engineering*, vol. 18, no. 5, pp. 627–641, 2021.

[8] X. Shi and J. Wang, "One dimensional magnetotelluric sounding inversion using simulated annealing," *Earth Science-Journal of China University of Geosciences (in Chinese)*, vol. 23, no. 5, pp. 542–546, 1998.

[9] S. P. Sharma, "Vfsares–a very fast simulated annealing fortran program for interpretation of 1-d dc resistivity sounding data from various electrode arrays," *Computers & Geosciences*, vol. 42, pp. 177–188, 2012.

[10] R. Shaw and S. Srivastava, "Particle swarm optimization: A new tool to invert geophysical data," *Geophysics*, vol. 72, no. 2, pp. F75–F83, 2007.

[11] X.-M. SHI, M. XIAO, J.-K. FAN, G.-S. YANG, and X.-H. ZHANG, "The damped pso algorithm and its application for magnetotelluric sounding data inversion," *Chinese Journal of Geophysics*, vol. 52, no. 4, pp. 1114–1120, 2009.

[12] C. Schwarzbach, R.-U. Börner, and K. Spitzer, "Two-dimensional inversion of direct current resistivity data using a parallel, multi-objective genetic algorithm," *Geophysical Journal International*, vol. 162, no. 3, pp. 685–695, 2005.

[13] J. Xu, Y. Ding, Z. Zhang, G. Xie, Z. Wang, Y. Zeng, and G. Li, "Multi-auv assisted seamless underwater target tracking relying on deep learning and reinforcement learning," in *2024 International Joint Conference on Neural Networks (IJCNN)*, 2024, pp. 1–9.

[14] Y. Ding, J. Xu, G. Xie, G. Li, J. Wang, and Y. Ren, "Advanced framework for underwater node repair via multi-auv based on multi-agent offline reinforcement learning," 09 2024.

[15] V. Puzyrev, "Deep learning electromagnetic inversion with convolutional neural networks," *Geophysical Journal International*, vol. 218, no. 2, pp. 817–832, 2019.

[16] Q. Di, G. Xue, C. Yin, and X. Li, "New methods of controlled-source electromagnetic detection in china," *Science China Earth Sciences*, vol. 63, pp. 1268–1277, 2020.

[17] B. Liu, Q. Guo, S. Li, B. Liu, Y. Ren, Y. Pang, X. Guo, L. Liu, and P. Jiang, "Deep learning inversion of electrical resistivity data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 8, pp. 5715–5728, 2020.

[18] Y. Yue, W. Su, and Y. Zhao, "Underwater acoustic aided grid localization of multi-uuvs with reinforcement learning," in *2023 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, 2023, pp. 1–6.

[19] G. Qiao, C. Zhao, F. Zhou, and N. Ahmed, "Distributed localization based on signal propagation loss for underwater sensor networks," *IEEE Access*, vol. 7, pp. 112 985–112 995, 2019.