A Appendix

A.1 A Counter Example of a Basis of the Null Space

We consider a special case of Eq. (9), where $a_i(\boldsymbol{x}) \equiv 0$, $b_i(\boldsymbol{x}) \equiv 1$, $g_i(\boldsymbol{x}) \equiv 0$, and the dimension is $d = 3$.

$$n_1(\boldsymbol{x})p_{j1}(\boldsymbol{x}) + n_2(\boldsymbol{x})p_{j2}(\boldsymbol{x}) + n_3(\boldsymbol{x})p_{j3}(\boldsymbol{x}) = 0, \qquad \boldsymbol{x} \in \gamma_i, \qquad \forall i = 1, \dots, m_j. \tag{A1}$$

And a counter example of $\boldsymbol{B}(\boldsymbol{x})$ is given by

$$\begin{aligned} \boldsymbol{B}(\boldsymbol{x}) &= [\boldsymbol{\beta}_1(\boldsymbol{x}), \boldsymbol{\beta}_2(\boldsymbol{x})], \\ \boldsymbol{\beta}_1(\boldsymbol{x}) &= [n_2(\boldsymbol{x}), -n_1(\boldsymbol{x}), 0]^\top, \\ \boldsymbol{\beta}_2(\boldsymbol{x}) &= [n_3(\boldsymbol{x}), 0, -n_1(\boldsymbol{x})]^\top. \end{aligned} \tag{A2}$$

One could verify that the above formula of $\boldsymbol{B}(\boldsymbol{x})$ is a basis of the null space, if $n_1(\boldsymbol{x}) \neq 0, \forall \boldsymbol{x} \in \gamma_i$. For a special case where γ_i is a plane parallel to the x-axis, however, we have $n_1(\boldsymbol{x}) \equiv 0, \forall \boldsymbol{x} \in \gamma_i$. In this case, $\boldsymbol{\beta}_1(\boldsymbol{x}), \boldsymbol{\beta}_2(\boldsymbol{x})$ are no longer linearly independent and cannot represent all possible solutions to $(p_{j1}(\boldsymbol{x}), p_{j2}(\boldsymbol{x}), p_{j3}(\boldsymbol{x}))$. Therefore, Eq. (A2) is not an admissible choice for $\boldsymbol{B}(\boldsymbol{x})$.

A.2 A Basis of the Null Space in Low Dimensions

Let $\tilde{\boldsymbol{n}} = (a_i, b_i\boldsymbol{n})/\sqrt{a_i^2 + b_i^2}$, $\tilde{g}_i = g_i/\sqrt{a_i^2 + b_i^2}$, and $\tilde{\boldsymbol{p}}_j = (u_j, \boldsymbol{p}_j)$. Eq. (9) is equivalent to

$$\tilde{\boldsymbol{n}}(\boldsymbol{x}) \cdot \tilde{\boldsymbol{p}}_j(\boldsymbol{x}) = \tilde{g}_i(\boldsymbol{x}), \qquad \boldsymbol{x} \in \gamma_i, \qquad \forall i = 1, \dots, m_j. \tag{A3}$$

For $d = 1$, we can rewrite Eq. (A3) as (the dimension of $\tilde{\boldsymbol{p}}_j$ is $d + 1$)

$$\tilde{n}_1(\boldsymbol{x})\tilde{p}_{j1}(\boldsymbol{x}) + \tilde{n}_2(\boldsymbol{x})\tilde{p}_{j2}(\boldsymbol{x}) = \tilde{g}_i(\boldsymbol{x}), \qquad \boldsymbol{x} \in \gamma_i, \qquad \forall i = 1, \dots, m_j. \tag{A4}$$

And we can find that the following basis is an acceptable one

$$\boldsymbol{B}(\boldsymbol{x}) = [\tilde{n}_2(\boldsymbol{x}), -\tilde{n}_1(\boldsymbol{x})]^\top, \tag{A5}$$

since $\boldsymbol{B}(\boldsymbol{x}) = 0 \Leftrightarrow \tilde{\boldsymbol{n}}(\boldsymbol{x}) = 0$, and the latter contradicts the fact that $\tilde{\boldsymbol{n}} \cdot \tilde{\boldsymbol{n}} = 1$. Then, we can use $\boldsymbol{B}$ to construct the general solution $\tilde{\boldsymbol{p}}_j^{\gamma_i}$ under $d = 1$.

And for $d = 2$, Eq. (A3) becomes

$$\tilde{n}_1(\boldsymbol{x})\tilde{p}_{j1}(\boldsymbol{x}) + \tilde{n}_2(\boldsymbol{x})\tilde{p}_{j2}(\boldsymbol{x}) + \tilde{n}_3(\boldsymbol{x})\tilde{p}_{j3}(\boldsymbol{x}) = \tilde{g}_i(\boldsymbol{x}), \qquad \boldsymbol{x} \in \gamma_i, \qquad \forall i = 1, \dots, m_j. \tag{A6}$$

An acceptable $\boldsymbol{B}(\boldsymbol{x})$ is given by

$$\begin{aligned} \boldsymbol{B}(\boldsymbol{x}) &= [\boldsymbol{\beta}_1(\boldsymbol{x}), \boldsymbol{\beta}_2(\boldsymbol{x}), \boldsymbol{\beta}_3(\boldsymbol{x})], \\ \boldsymbol{\beta}_1(\boldsymbol{x}) &= [0, \tilde{n}_3(\boldsymbol{x}), -\tilde{n}_2(\boldsymbol{x})]^\top, \\ \boldsymbol{\beta}_2(\boldsymbol{x}) &= [-\tilde{n}_3(\boldsymbol{x}), 0, \tilde{n}_1(\boldsymbol{x})]^\top, \\ \boldsymbol{\beta}_3(\boldsymbol{x}) &= [\tilde{n}_2(\boldsymbol{x}), -\tilde{n}_1(\boldsymbol{x}), 0]^\top. \end{aligned} \tag{A7}$$

We note that $\boldsymbol{\beta}_1(\boldsymbol{x}), \boldsymbol{\beta}_2(\boldsymbol{x}), \boldsymbol{\beta}_3(\boldsymbol{x})$ live in the null space and $\mathrm{rank}(\boldsymbol{B}(\boldsymbol{x})) = 2$. So $\boldsymbol{B}(\boldsymbol{x})$ contains a basis in the null space, which can be used to construct the general solution $\tilde{\boldsymbol{p}}_j^{\gamma_i}$ under $d = 2$.

A.3 Explanation for the General Solution

We first show how to find an admissible expression of $\boldsymbol{B}(\boldsymbol{x})$ in arbitrary dimensions with respect to Eq. (A3) which is equivalent to original formulation of the BC (see Eq. (9)). We perform a Gram–Schmidt orthogonalization of $\tilde{\boldsymbol{n}}$ (whose dimension is $d + 1$) on each vector in the standard basis to get

$$\boldsymbol{\beta}_k(\boldsymbol{x}) = \boldsymbol{e}_k - \frac{\boldsymbol{e}_k \cdot \tilde{\boldsymbol{n}}(\boldsymbol{x})}{\tilde{\boldsymbol{n}}(\boldsymbol{x}) \cdot \tilde{\boldsymbol{n}}(\boldsymbol{x})}\tilde{\boldsymbol{n}}(\boldsymbol{x}) = \boldsymbol{e}_k - \big(\boldsymbol{e}_k \cdot \tilde{\boldsymbol{n}}(\boldsymbol{x})\big)\tilde{\boldsymbol{n}}(\boldsymbol{x}), \qquad k = 1, \dots, d+1, \tag{A8}$$

where $[\boldsymbol{e}_1, \dots, \boldsymbol{e}_{d+1}] = \boldsymbol{I}_{d+1}$, and obviously all $\boldsymbol{\beta}_k(\boldsymbol{x}), k = 1, \dots, d + 1$, are in the $\mathrm{Null}(\tilde{\boldsymbol{n}}^\top)$. We set $\boldsymbol{B}(\boldsymbol{x}) = [\boldsymbol{\beta}_1(\boldsymbol{x}), \dots, \boldsymbol{\beta}_{d+1}(\boldsymbol{x})] = \boldsymbol{I}_{d+1} - \tilde{\boldsymbol{n}}(\boldsymbol{x})\tilde{\boldsymbol{n}}(\boldsymbol{x})^\top$. Furthermore, we can prove that $\mathrm{rank}(\boldsymbol{B}(\boldsymbol{x})) = d, \forall \boldsymbol{x} \in \gamma_i$ (see Lemma A.1). Therefore, for $\forall \boldsymbol{x} \in \gamma_i$, $\boldsymbol{B}(\boldsymbol{x})$ always contains a basis of $\mathrm{Null}(\tilde{\boldsymbol{n}}^\top)$, and we consider such a $\boldsymbol{B}(x)$ to be an ideal choice for the general solution $\tilde{\boldsymbol{p}}_j^{\gamma_i}$.

Lemma A.1. $\mathrm{rank}(\boldsymbol{B}(\boldsymbol{x})) = d$ *holds for all* $\boldsymbol{x} \in \gamma_i$, *where* $\boldsymbol{B}(\boldsymbol{x}) = \boldsymbol{I}_{d+1} - \tilde{\boldsymbol{n}}(\boldsymbol{x})\tilde{\boldsymbol{n}}(\boldsymbol{x})^\top$.

Proof. For all $\boldsymbol{x} \in \gamma_i$, we have known that $\boldsymbol{B} = \boldsymbol{I}_{d+1} - \tilde{\boldsymbol{n}}\tilde{\boldsymbol{n}}^\top$, where $\tilde{\boldsymbol{n}} \cdot \tilde{\boldsymbol{n}} = \tilde{\boldsymbol{n}}^\top\tilde{\boldsymbol{n}} = 1$, and

$$\boldsymbol{B}\tilde{\boldsymbol{n}} = (\boldsymbol{I}_{d+1} - \tilde{\boldsymbol{n}}\tilde{\boldsymbol{n}}^\top)\tilde{\boldsymbol{n}} = \tilde{\boldsymbol{n}} - \tilde{\boldsymbol{n}}\tilde{\boldsymbol{n}}^\top\tilde{\boldsymbol{n}} = \tilde{\boldsymbol{n}} - \tilde{\boldsymbol{n}} = \boldsymbol{0}. \tag{A9}$$

Hence, $\mathrm{rank}(\boldsymbol{B}) \leq d$. Besides, we notice that $\boldsymbol{H} = \boldsymbol{I}_{d+1} - 2\tilde{\boldsymbol{n}}\tilde{\boldsymbol{n}}^\top$ is a Householder matrix, which is an invertible matrix, since

$$\boldsymbol{H}^\top\boldsymbol{H} = (\boldsymbol{I}_{d+1} - 2\tilde{\boldsymbol{n}}\tilde{\boldsymbol{n}}^\top)^2 = \boldsymbol{I}_{d+1} - 4\tilde{\boldsymbol{n}}\tilde{\boldsymbol{n}}^\top + 4\tilde{\boldsymbol{n}}\tilde{\boldsymbol{n}}^\top\tilde{\boldsymbol{n}}\tilde{\boldsymbol{n}}^\top = \boldsymbol{I}_{d+1} - 4\tilde{\boldsymbol{n}}\tilde{\boldsymbol{n}}^\top + 4\tilde{\boldsymbol{n}}\tilde{\boldsymbol{n}}^\top = \boldsymbol{I}_{d+1}. \tag{A10}$$

So $\mathrm{rank}(\boldsymbol{H}) = d + 1$, and we have

$$d + 1 = \mathrm{rank}(\boldsymbol{I}_{d+1} - 2\tilde{\boldsymbol{n}}\tilde{\boldsymbol{n}}^\top) \leq \mathrm{rank}(\boldsymbol{I}_{d+1} - \tilde{\boldsymbol{n}}\tilde{\boldsymbol{n}}^\top) + \mathrm{rank}(\tilde{\boldsymbol{n}}\tilde{\boldsymbol{n}}^\top) = \mathrm{rank}(\boldsymbol{B}) + 1, \tag{A11}$$

which can deduce $d \leq \mathrm{rank}(\boldsymbol{B})$. Therefore, $\mathrm{rank}(\boldsymbol{B}) = d$. □

Finally, we show that the general solution in Eq. (10) satisfies the BC in Eq. (A3).

$$\begin{aligned}
\tilde{\boldsymbol{n}}(\boldsymbol{x}) \cdot \tilde{\boldsymbol{p}}_j^{\gamma_i}(\boldsymbol{x}) &= \tilde{\boldsymbol{n}}(\boldsymbol{x}) \cdot \boldsymbol{B}(\boldsymbol{x})\mathrm{NN}_j^{\gamma_i}(\boldsymbol{x}) + \tilde{\boldsymbol{n}}(\boldsymbol{x}) \cdot \tilde{\boldsymbol{n}}(\boldsymbol{x})\tilde{g}_i(\boldsymbol{x}) \\
&= \tilde{\boldsymbol{n}}(\boldsymbol{x}) \cdot \left(\boldsymbol{I}_{d+1} - \tilde{\boldsymbol{n}}(\boldsymbol{x})\tilde{\boldsymbol{n}}(\boldsymbol{x})^\top\right)\mathrm{NN}_j^{\gamma_i}(\boldsymbol{x}) + \tilde{g}_i(\boldsymbol{x}) \\
&= \tilde{g}_i(\boldsymbol{x}),
\end{aligned} \tag{A12}$$

where we omit the trainable parameters for simplicity. Besides, the discussion of $\boldsymbol{B}(\boldsymbol{x})$ in low-dimensional cases (i.e., $d = 1$ and $d = 2$, see Appendix A.3) is similar, and we will leave it to the reader.

A.4 Theoretical Guarantee of the Constructed Ansatz

In Appendix A.3, we have demonstrated that $\boldsymbol{B}(\boldsymbol{x})$ contains a basis of the null space of the BC for $\forall \boldsymbol{x} \in \gamma_i$ and the general solution in Eq. (10) satisfies the corresponding BC. In this subsection, we will show that our constructed ansatz in Eq. (11) is theoretically correct. We first prove that the ansatz in Eq. (11) satisfies all the BCs under the following assumptions.

Assumption A.2. The problem domain Ω is bounded.

Assumption A.3. The shortest distance between $\gamma_1, \ldots, \gamma_{m_j}$ is greater than zero for $j = 1, \ldots, n$.

Assumption A.4. All the extended distance functions $l^{\partial\Omega}, l^{\gamma_i}, i = 1, \ldots, m_j$ are continuous and satisfy that $\min_{\boldsymbol{x} \in \partial\Omega \setminus \gamma_i} l^{\gamma_i}(\boldsymbol{x}) \geq C_i$, $\forall \boldsymbol{x} \in \gamma_i$, $i = 1, \ldots, m_j$ for $j = 1, \ldots, n$, where C_i is a positive constant.

Theorem A.5. $\forall \epsilon > 0$, *there exists* $\beta_s^0 \in \mathbb{R}$, *such that*

$$|\tilde{\boldsymbol{n}}(\boldsymbol{x}) \cdot (\hat{u}_j, \hat{\boldsymbol{p}}_j) - \tilde{g}_i(\boldsymbol{x})| < \epsilon, \tag{A13}$$

holds for all $\beta_s > \beta_s^0$, $\boldsymbol{x} \in \gamma_i$, $i = 1, \ldots, m_j$, $j = 1, \ldots, n$, *where* $\tilde{\boldsymbol{n}} = (a_i, b_i\boldsymbol{n})/\sqrt{a_i^2 + b_i^2}$, $\tilde{g}_i = g_i/\sqrt{a_i^2 + b_i^2}$, *and* $\boldsymbol{B}(\boldsymbol{x}) = \boldsymbol{I}_{d+1} - \tilde{\boldsymbol{n}}(\boldsymbol{x})\tilde{\boldsymbol{n}}(\boldsymbol{x})^\top$.

Proof. For any $\boldsymbol{x} \in \gamma_i$, we have $l^{\partial\Omega}(\boldsymbol{x}) = 0$ according to the definition of the extended distance functions. Thus, $(\hat{u}_j, \hat{\boldsymbol{p}}_j)$ is now equal to

$$(\hat{u}_j, \hat{\boldsymbol{p}}_j) = \sum_{k=1}^{m_j} \exp\left[-\alpha_k l^{\gamma_k}(\boldsymbol{x})\right]\tilde{\boldsymbol{p}}_j^{\gamma_k}(\boldsymbol{x}), \tag{A14}$$

where we omit the trainable parameters. Then, according to Assumptions A.2 $\sim$ A.4, we can choose a sufficiently large β_s^i (see Eq. (12) for the relationship between α_i and β_s), such that

$$\begin{aligned}
|\tilde{\boldsymbol{n}}(\boldsymbol{x}) \cdot (\hat{u}_j, \hat{\boldsymbol{p}}_j) - \tilde{g}_i(\boldsymbol{x})| &= \left| \tilde{\boldsymbol{n}}(\boldsymbol{x}) \cdot \left(\sum_{k=1}^{m_j} \exp\big[-\alpha_k l^{\gamma_k}(\boldsymbol{x})\big] \tilde{\boldsymbol{p}}_j^{\gamma_k}(\boldsymbol{x}) \right) - \tilde{g}_i(\boldsymbol{x}) \right| \\
&< \left| \tilde{\boldsymbol{n}}(\boldsymbol{x}) \cdot \exp\big[-\alpha_i l^{\gamma_i}(\boldsymbol{x})\big] \tilde{\boldsymbol{p}}_j^{\gamma_i}(\boldsymbol{x}) - \tilde{g}_i(\boldsymbol{x}) \right| \\
&\quad + \left| \tilde{\boldsymbol{n}}(\boldsymbol{x}) \cdot \sum_{k \neq i} \exp\big[-\alpha_k l^{\gamma_k}(\boldsymbol{x})\big] \tilde{\boldsymbol{p}}_j^{\gamma_k}(\boldsymbol{x}) \right| \\
&\leq \left| \tilde{\boldsymbol{n}}(\boldsymbol{x}) \cdot \tilde{\boldsymbol{p}}_j^{\gamma_i}(\boldsymbol{x}) - \tilde{g}_i(\boldsymbol{x}) \right| + \left| \sum_{k \neq i} \exp\big[-\alpha_k l^{\gamma_k}(\boldsymbol{x})\big] \tilde{\boldsymbol{p}}_j^{\gamma_k}(\boldsymbol{x}) \right| \\
&= \left| \sum_{k \neq i} \exp\big[-\alpha_k l^{\gamma_k}(\boldsymbol{x})\big] \tilde{\boldsymbol{p}}_j^{\gamma_k}(\boldsymbol{x}) \right| \\
&< \epsilon,
\end{aligned} \tag{A15}$$

where we note that $l^{\gamma_i}(\boldsymbol{x}) = 0$ and $l^{\gamma_k}(\boldsymbol{x}) > 0, \forall k \neq i$ for all $\boldsymbol{x} \in \gamma_i$. Let $\beta_s^0 = \max\{\beta_s^i \mid i = 1, \ldots, m_j\}$, then according to the arbitrariness of j, we conclude that the theorem holds. □

Next, we will prove that our ansatz can approximate the solution to the PDEs with arbitrarily low errors under following assumptions in addition to Assumptions A.2 $\sim$ A.4.

Assumption A.6. The solution to the PDEs $\boldsymbol{u}(\boldsymbol{x})$ is unique, bounded, and at least first order continuous by element.

Assumption A.7. $a_i(\boldsymbol{x})$, $b_i(\boldsymbol{x})$, and $g_i(\boldsymbol{x})$ are continuous (hence $\tilde{g}_i(\boldsymbol{x})$ is continuous, too) in γ_i for $i = 1, \ldots, m_j$, $j = 1, \ldots, n$.

Assumption A.8. Since $\boldsymbol{B}(\boldsymbol{x}) = \boldsymbol{I}_{d+1} - \tilde{\boldsymbol{n}}(\boldsymbol{x})\tilde{\boldsymbol{n}}(\boldsymbol{x})^\top$ is a real symmetric matrix, we can perform an orthogonal diagonalization $\boldsymbol{B}(\boldsymbol{x}) = \boldsymbol{P}(\boldsymbol{x})^\top \Lambda(\boldsymbol{x}) \boldsymbol{P}(\boldsymbol{x})$, where $\Lambda(\boldsymbol{x}) = \mathrm{diag}(\lambda_1(\boldsymbol{x}), \ldots, \lambda_d(\boldsymbol{x}), 0)$, $\lambda_1(\boldsymbol{x}) > \cdots > \lambda_d(\boldsymbol{x}) > 0$. We assume that $\tilde{\boldsymbol{n}}(\boldsymbol{x})$, $\boldsymbol{P}(\boldsymbol{x})$, and $\Lambda(\boldsymbol{x})$ are piece-wise continuous by element in γ_i for $i = 1, \ldots, m_j$, $j = 1, \ldots, n$.

To begin with, we prove this lemma.

Lemma A.9. *$\forall \epsilon > 0$, there exists $\boldsymbol{\theta}_j^{\gamma_i} \in \Theta_j^{\gamma_i}$, such that*

$$\left\| \tilde{\boldsymbol{p}}_j^{\gamma_i}(\boldsymbol{x}; \boldsymbol{\theta}_j^{\gamma_i}) - \boldsymbol{q}(\boldsymbol{x}) \right\|_1 < \epsilon, \tag{A16}$$

holds for all $\boldsymbol{x} \in \gamma_i$ if $\boldsymbol{q}(\boldsymbol{x})$ is continuous in γ_i and satisfies the BC (i.e., $\tilde{\boldsymbol{n}}(\boldsymbol{x}) \cdot \boldsymbol{q}(\boldsymbol{x}) = \tilde{g}_i(\boldsymbol{x}), \forall \boldsymbol{x} \in \gamma_i$), where $\Theta_j^{\gamma_i}$ is the parameter space of the neural network $\mathrm{NN}_j^{\gamma_i}$, $\|\cdot\|_1$ is the 1-norm of matrices (operator norm), and $\tilde{\boldsymbol{p}}_j^{\gamma_i}$ as well as $\boldsymbol{q}$ are both of dimension $d+1$. The above conclusion holds for all $i = 1, \ldots, m_j$, $j = 1, \ldots, n$.

Proof. From Eq. (A8) and Lemma A.1, we know that $\boldsymbol{B}(\boldsymbol{x})$ contain a basis of $\mathrm{Null}(\tilde{\boldsymbol{n}}^\top)$. Since $\boldsymbol{q}(\boldsymbol{x})$ satifies the BC, we can represent it as

$$\boldsymbol{q}(\boldsymbol{x}) = \boldsymbol{B}(\boldsymbol{x})\boldsymbol{r}(\boldsymbol{x}) + \tilde{\boldsymbol{n}}(\boldsymbol{x})\tilde{g}_i(\boldsymbol{x}). \tag{A17}$$

Then we will show that there exists a piece-wise continuous choice of $\boldsymbol{r}(\boldsymbol{x})$. We rewrite Eq. (A17) as

$$\boldsymbol{q}(\boldsymbol{x}) = \boldsymbol{P}(\boldsymbol{x})^\top \Lambda(\boldsymbol{x}) \boldsymbol{P}(\boldsymbol{x}) \boldsymbol{r}(\boldsymbol{x}) + \tilde{\boldsymbol{n}}(\boldsymbol{x})\tilde{g}_i(\boldsymbol{x}). \tag{A18}$$

Since $\boldsymbol{B}(\boldsymbol{x})$ has $d+1$ column vectors, which is greater than the dimension of $\mathrm{Null}(\tilde{\boldsymbol{n}}^\top)$ (i.e., d), the choice of $\boldsymbol{q}(\boldsymbol{x})$ is not unique. We can choose a particular $\boldsymbol{q}(\boldsymbol{x})$ which satisfies that the last element of $\boldsymbol{P}(\boldsymbol{x})\boldsymbol{r}(\boldsymbol{x})$ is zero (i.e., $\boldsymbol{P}(\boldsymbol{x})\boldsymbol{r}(\boldsymbol{x}) = [\ldots, 0]^\top$). Next, we continue with the equivalent

transformation of Eq. (A18).

$$
\begin{aligned}
&\boldsymbol{P}(\boldsymbol{x})^\top \Lambda(\boldsymbol{x})\boldsymbol{P}(\boldsymbol{x})\boldsymbol{r}(\boldsymbol{x}) + \tilde{\boldsymbol{n}}(\boldsymbol{x})\tilde{g}_i(\boldsymbol{x}) = \boldsymbol{q}(\boldsymbol{x}),\\
\iff\quad & \boldsymbol{P}(\boldsymbol{x})^\top \Lambda(\boldsymbol{x})\boldsymbol{P}(\boldsymbol{x})\boldsymbol{r}(\boldsymbol{x}) = \boldsymbol{q}(\boldsymbol{x}) - \tilde{\boldsymbol{n}}(\boldsymbol{x})\tilde{g}_i(\boldsymbol{x}),\\
\iff\quad & \Lambda(\boldsymbol{x})\boldsymbol{P}(\boldsymbol{x})\boldsymbol{r}(\boldsymbol{x}) = \boldsymbol{P}(\boldsymbol{x})\left(\boldsymbol{q}(\boldsymbol{x}) - \tilde{\boldsymbol{n}}(\boldsymbol{x})\tilde{g}_i(\boldsymbol{x})\right),\\
\iff\quad & \operatorname{diag}(1,\dots,1,0)\boldsymbol{P}(\boldsymbol{x})\boldsymbol{r}(\boldsymbol{x}) = \Lambda'(\boldsymbol{x})\boldsymbol{P}(\boldsymbol{x})\left(\boldsymbol{q}(\boldsymbol{x}) - \tilde{\boldsymbol{n}}(\boldsymbol{x})\tilde{g}_i(\boldsymbol{x})\right),\\
\iff\quad & \boldsymbol{P}(\boldsymbol{x})\boldsymbol{r}(\boldsymbol{x}) = \Lambda'(\boldsymbol{x})\boldsymbol{P}(\boldsymbol{x})\left(\boldsymbol{q}(\boldsymbol{x}) - \tilde{\boldsymbol{n}}(\boldsymbol{x})\tilde{g}_i(\boldsymbol{x})\right),
\end{aligned}
\tag{A19}
$$

where $\Lambda'(\boldsymbol{x}) = \operatorname{diag}(1/\lambda_1(\boldsymbol{x}),\dots,1/\lambda_d(\boldsymbol{x}),0)$. The last equivalence holds because the last element of $\boldsymbol{P}(\boldsymbol{x})\boldsymbol{r}(\boldsymbol{x})$ is always zero. From Assumption A.7 and A.8, combining the above formula, we have that $\boldsymbol{P}(\boldsymbol{x})\boldsymbol{r}(\boldsymbol{x})$ is piece-wise continuous by element. Noticing that $\boldsymbol{r}(\boldsymbol{x}) = \boldsymbol{P}(\boldsymbol{x})^\top\boldsymbol{P}(\boldsymbol{x})\boldsymbol{r}(\boldsymbol{x})$, we know that the $\boldsymbol{r}(\boldsymbol{x})$ we chosen is also piece-wise continuous by element.

We notice that

$$
\begin{aligned}
\|\boldsymbol{B}(\boldsymbol{x})\|_1 &= \left\|\boldsymbol{I}_{d+1} - \tilde{\boldsymbol{n}}(\boldsymbol{x})\tilde{\boldsymbol{n}}(\boldsymbol{x})^\top\right\|_1\\
&\le \|\boldsymbol{I}_{d+1}\|_1 + \left\|\tilde{\boldsymbol{n}}(\boldsymbol{x})\tilde{\boldsymbol{n}}(\boldsymbol{x})^\top\right\|_1\\
&\le 1 + d + 1\\
&= d + 2.
\end{aligned}
\tag{A20}
$$

According to the Universal Approximation of neural networks [2], $\forall\epsilon > 0$, there exists $\boldsymbol{\theta}_j^{\gamma_i} \in \Theta_j^{\gamma_i}$, such that

$$
\left\|\mathrm{NN}_j^{\gamma_i}(\boldsymbol{x};\boldsymbol{\theta}_j^{\gamma_i}) - \boldsymbol{r}(\boldsymbol{x})\right\|_1 < \frac{\epsilon}{d+2},
\tag{A21}
$$

holds for all $\boldsymbol{x} \in \gamma_i$. Therefore,

$$
\begin{aligned}
\left\|\tilde{\boldsymbol{p}}_j^{\gamma_i}(\boldsymbol{x};\boldsymbol{\theta}_j^{\gamma_i}) - \boldsymbol{q}(\boldsymbol{x})\right\|_1 &= \left\|\boldsymbol{B}(\boldsymbol{x})\left(\mathrm{NN}_j^{\gamma_i}(\boldsymbol{x};\boldsymbol{\theta}_j^{\gamma_i}) - \boldsymbol{r}(\boldsymbol{x})\right)\right\|_1\\
&\le \|\boldsymbol{B}(\boldsymbol{x})\|_1 \left\|\mathrm{NN}_j^{\gamma_i}(\boldsymbol{x};\boldsymbol{\theta}_j^{\gamma_i}) - \boldsymbol{r}(\boldsymbol{x})\right\|_1\\
&< \epsilon.
\end{aligned}
\tag{A22}
$$

According to the arbitrariness of i and j, we conclude that the lemma holds. □

Finally, we state the following theorem.

Theorem A.10. *$\forall\epsilon > 0$, there exists $\beta_s \in \mathbb{R}$, $\boldsymbol{\theta}_{\mathrm{main}} \in \Theta_{\mathrm{main}}$, $\boldsymbol{\theta}_j^{\gamma_i} \in \Theta_j^{\gamma_i}, i = 1,\dots,m_j$, such that*

$$
\|(\hat{u}_j, \hat{\boldsymbol{p}}_j) - (u_j, \nabla u_j)\|_1 < \epsilon,
\tag{A23}
$$

holds for all $\boldsymbol{x} \in \Omega \cup \partial\Omega$, $j = 1,\dots,n$, where Θ_* is the parameter space of the corresponding neural network, $\|\cdot\|_1$ is the 1-norm. The ground truth solution to the PDEs is $\boldsymbol{u}(\boldsymbol{x}) = (u_1(\boldsymbol{x}),\dots,u_n(\boldsymbol{x}))$.

Proof. For $\boldsymbol{x} \in \gamma_i$, $i = 1,\dots,m_j$, $(u_j, \nabla u_j)$ is continuous (according to Assumption A.6) and satisfies the BC (which the solution needs to meet). From Lemma A.9, the definition of $(\hat{u}_j, \hat{\boldsymbol{p}}_j)$ in Eq. (11), and Assumptions A.2 $\sim$ A.4, we can find $\boldsymbol{\theta}_j^{\gamma_i} \in \Theta_j^{\gamma_i}$ and a large enough β_s^i such that Eq. (A23) holds for all $\boldsymbol{x} \in \gamma_i$.

Then we fix $\beta_s = \max\{\beta_s^i \mid i = 1,\dots,m_j\}$ and $\boldsymbol{\theta}_j^{\gamma_i} \in \Theta_j^{\gamma_i}, i = 1,\dots,m_j$ (which are what we determined for $\boldsymbol{x} \in \gamma_i, i = 1,\dots,m_j$). From Assumption A.2, we have $|l^{\partial\Omega}(\boldsymbol{x})| < C, \forall\boldsymbol{x} \in \Omega$, where C is a positive constant. For $\boldsymbol{x} \in \Omega$, according to the Universal Approximation Theorem of neural networks [1], there exists $\boldsymbol{\theta}_{\mathrm{main}} \in \Theta_{\mathrm{main}}$ satisfying that

$$
\left\|\frac{(u_j, \nabla u_j) - \sum_{i=1}^{m_j}\exp\left[-\alpha_i l^{\gamma_i}(\boldsymbol{x})\right]\tilde{\boldsymbol{p}}_j^{\gamma_i}(\boldsymbol{x};\boldsymbol{\theta}_j^{\gamma_i})}{l^{\partial\Omega}(\boldsymbol{x})} - \mathrm{NN}_{\mathrm{main}}(\boldsymbol{x};\boldsymbol{\theta}_{\mathrm{main}})\right\|_1 < \frac{\epsilon}{C}.
\tag{A24}
$$

Therefore,

$$
\begin{aligned}
&\|(\hat{u}_j, \hat{\boldsymbol{p}}_j) - (u_j, \nabla u_j)\|_1\\
=&\left\|(u_j, \nabla u_j) - \sum_{i=1}^{m_j}\exp\left[-\alpha_i l^{\gamma_i}(\boldsymbol{x})\right]\tilde{\boldsymbol{p}}_j^{\gamma_i}(\boldsymbol{x};\boldsymbol{\theta}_j^{\gamma_i}) - l^{\partial\Omega}(\boldsymbol{x})\mathrm{NN}_{\mathrm{main}}(\boldsymbol{x};\boldsymbol{\theta}_{\mathrm{main}})\right\|_1\\
=&\left\|\frac{(u_j, \nabla u_j) - \sum_{i=1}^{m_j}\exp\left[-\alpha_i l^{\gamma_i}(\boldsymbol{x})\right]\tilde{\boldsymbol{p}}_j^{\gamma_i}(\boldsymbol{x};\boldsymbol{\theta}_j^{\gamma_i})}{l^{\partial\Omega}(\boldsymbol{x})} - \mathrm{NN}_{\mathrm{main}}(\boldsymbol{x};\boldsymbol{\theta}_{\mathrm{main}})\right\|_1 \left|l^{\partial\Omega}(\boldsymbol{x})\right|\\
<&\frac{\epsilon}{C}\cdot C = \epsilon.
\end{aligned}
\tag{A25}
$$

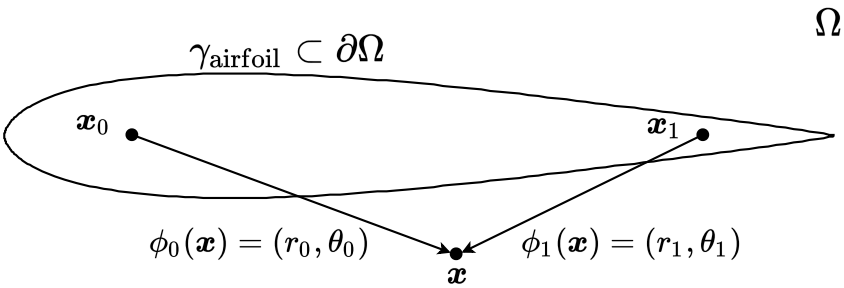


Figure A1: Illustration of the extension from γ_{airfoil} to $\Omega \cup \partial\Omega$.

According to the arbitrariness of j, we have proven this theorem. □

Besides, we note that it is easy to extend the above theorems to time-dependent cases (the ansatz is given in Appendix A.6), which will not be discussed separately here.

A.5 Extension of the Parameter Functions in the BCs

In Eq. (9), it is noted that a_i, b_i, $\boldsymbol{n}$ or g_i may be only defined at γ_i. But they are included in our ansatz (see Eq. (10) and Eq. (11)), which is defined in $\Omega \cup \partial\Omega$. So we need to extend their definition smoothly to $\Omega \cup \partial\Omega$, using interpolation or approximation via neural networks. We consider the airfoil boundary (i.e, γ_{airfoil}) in Section 5.3 as a motivating example.

Supposing $f(\boldsymbol{x})$ is only defined in γ_{airfoil}, our task is to extend its definition to $\Omega \cup \partial\Omega$. As shown in Figure A1, we first place two reference points (i.e., $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$) on the front and rear half of the airfoil. For any $\boldsymbol{x} \in \Omega \cup \partial\Omega$, it can be expressed as polar coordinates with respect to $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$, respectively. We concatenate the two polar coordinates to form a new space. We next perform interpolation and approximation under the new space. This is because in the new space we can better characterize the shape of the airfoil. It is true that there are many ways for coordinate transformations, not limited to the example here.

As for the interpolation, we can sample several points at the γ_{airfoil} to obtain the dataset $\{((\theta_0^{(i)}, \theta_1^{(i)}), f^{(i)})\}_{i=1}^N$. For any $\boldsymbol{x} \in \Omega \cup \partial\Omega$, we generate the corresponding extended $f(\boldsymbol{x})$ by interpolating in the dataset. The interpolation method used here depends on the smoothness requirements of the ansatz. In addition, the number of reference points can also be changed, and in experiments we found that only one reference point is enough.

Approximation via neural networks is a general method that does not require manual design. In this case, we can sample several points at the γ_{airfoil} to construct our dataset $\{((\phi_0(\boldsymbol{x}^{(i)}), \phi_1(\boldsymbol{x}^{(i)})), f^{(i)})\}_{i=1}^N$, followed by training a neural network on the dataset, i.e. $\text{NN}(\phi_0(\boldsymbol{x}^{(i)}), \phi_1(\boldsymbol{x}^{(i)})) \approx f^{(i)}$. For any $\boldsymbol{x} \in \Omega \cup \partial\Omega$, we take $\text{NN}(\phi_0(\boldsymbol{x}), \phi_1(\boldsymbol{x}))$ as the corresponding extended $f(\boldsymbol{x})$. We can also train the neural network in the original space. However, experimental results show that training on the new space can achieve better results. The reason may be that the complex geometry become smoother and easier to learn in the new space.

It is worth noting that, in addition to the cases mentioned above, the extended distance functions $l(\boldsymbol{x})$ (here we omit the superscript and see Eq. (4) for its definition) may also need to be handled similarly. Because for the complex geometry, the distance function can be very complex and we may want to replace it with a cheap surrogate model. The methods are similar, including approximating the distance function with a neural network, or constructing splines function [5].

A.6 The Hard-Constraint Framework for Time-dependent PDEs

In this section, we consider the following time-dependent PDEs

$$\mathcal{F}[\boldsymbol{u}(\boldsymbol{x}, t)] = \boldsymbol{0}, \qquad \boldsymbol{x} = (x_1, \ldots, x_d) \in \Omega, t \in (0, T], \tag{A26}$$

where t is the temporal coordinate, and the other notations are the same as those in Section 3.1. For each $u_j, j = 1, \ldots, n$, we pose suitable boundary conditions (BCs)

$$a_i(\boldsymbol{x}, t)u_j + b_i(\boldsymbol{x}, t)\big(\boldsymbol{n}(\boldsymbol{x}) \cdot \nabla u_j\big) = g_i(\boldsymbol{x}, t), \quad \boldsymbol{x} \in \gamma_i, t \in (0, T], \quad \forall i = 1, \ldots, m_j, \tag{A27}$$

and an initial condition (IC)

$$u_j(\boldsymbol{x}, 0) = f_j(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega. \tag{A28}$$

Following the pipeline described in Section 3.2, we can find the general solution $\tilde{\boldsymbol{p}}_j^{\gamma_i}(\boldsymbol{x}, t)$ as

$$\tilde{\boldsymbol{p}}_j^{\gamma_i} = \boldsymbol{B}(\boldsymbol{x}, t)\mathrm{NN}_j^{\gamma_i}(\boldsymbol{x}, t) + \tilde{\boldsymbol{n}}(\boldsymbol{x}, t)\tilde{g}_i(\boldsymbol{x}, t), \tag{A29}$$

where $\tilde{\boldsymbol{n}} = (a_i, b_i\boldsymbol{n})/\sqrt{a_i^2 + b_i^2}$, $\tilde{g}_i = g_i/\sqrt{a_i^2 + b_i^2}$, $\mathrm{NN}_j^{\gamma_i} : \mathbb{R}^{d+1} \to \mathbb{R}^{d+1}$ is a neural network, and $\boldsymbol{B}(\boldsymbol{x}, t) = \boldsymbol{I}_{d+1} - \tilde{\boldsymbol{n}}(\boldsymbol{x}, t)\tilde{\boldsymbol{n}}(\boldsymbol{x}, t)^\top$. And we omit the trainable parameters of neural networks for neatness.

Finally, we can construct our ansatz $(\hat{u}_j, \hat{\boldsymbol{p}}_j)$ as

$$(u_j^\dagger, \hat{\boldsymbol{p}}_j) = l^{\partial\Omega}(\boldsymbol{x})\mathrm{NN}_{\mathrm{main}}(\boldsymbol{x}, t) + \sum_{i=1}^{m_j} \exp\big[-\alpha_i l^{\gamma_i}(\boldsymbol{x})\big]\tilde{\boldsymbol{p}}_j^{\gamma_i}(\boldsymbol{x}, t), \quad \forall j = 1, \ldots, n, \tag{A30a}$$

$$\hat{u}_j = u_j^\dagger(\boldsymbol{x}, t)\big(1 - \exp\left[-\beta_t t\right]\big) + f_j(\boldsymbol{x})\exp\left[-\beta_t t\right], \quad \forall j = 1, \ldots, n, \tag{A30b}$$

where $u_j^\dagger$ is an intermediate variable that incorporates hard constraints in spatial dimensions, $\mathrm{NN}_{\mathrm{main}} : \mathbb{R}^{d+1} \to \mathbb{R}^{d+1}$ is the main neural network, $l^{\partial\Omega}, l^{\gamma_i}, i = 1, \ldots, m_j$ are extended distance functions (see Eq. (4)), α_i $(i = 1, \ldots, m_j)$ is determined in Eq. (12), and $\beta_t \in \mathbb{R}$ is a hyper-parameter of the "hardness" in the temporal domain.

A.7 Supplements to the Theoretical Analysis

In this section, we first give some supplements to the problem setting in Section 4. Then we present the proof of Theorem 4.1. Finally, we will characterize the mechanism described in Section 4 with another tool, the condition number.

A.7.1 Supplements to the Problem Setting

As mentioned in Section 4, we consider the following 1D Poisson's equation

$$\Delta u(x) = -a^2 \sin ax, \qquad x \in (0, 2\pi), \tag{A31a}$$

$$u(x) = 0, \qquad x = 0 \vee x = 2\pi, \tag{A31b}$$

where $a \in \mathbb{R}$ and u is the physical quantity of interest. Here we use a single-layer neural network of width K as our ansatz, i.e., $\hat{u} = \boldsymbol{c}^\top\sigma(\boldsymbol{w}x + \boldsymbol{b})$, where $\boldsymbol{c}, \boldsymbol{w}, \boldsymbol{b} \in \mathbb{R}^K$, σ is an element-wise activation function (for simplicity, we take σ as $\tanh$). To study the impact of the *extra fields* alone, we train $\hat{u}$ in a soft-constrained manner. For ease of discussion, we consider the loss function in continuous form

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_{\mathcal{F}}(\boldsymbol{\theta}) + \mathcal{L}_{\mathcal{B}}(\boldsymbol{\theta}) \approx \frac{1}{2\pi}\int_0^{2\pi} \big(\Delta\hat{u}(x) + a^2\sin(ax)\big)^2\,\mathrm{d}x + \big(\hat{u}(0)\big)^2 + \big(\hat{u}(2\pi)\big)^2, \tag{A32}$$

where $\boldsymbol{\theta} = (\boldsymbol{c}, \boldsymbol{w}, \boldsymbol{b})$ is a set of trainable parameters.

Let $p = \nabla u = \mathrm{d}u/\,\mathrm{d}x$. We reformulate Eq. (A31) via the *extra fields* to obtain

$$\nabla p(x) = -a^2 \sin ax, \qquad x \in (0, 2\pi), \tag{A33a}$$

$$p(x) = \nabla u(x), \qquad x \in (0, 2\pi), \tag{A33b}$$

$$u(x) = 0, \qquad x = 0 \vee x = 2\pi. \tag{A33c}$$

Our ansatz becomes $\hat{u} = \boldsymbol{c}^\top\sigma(\boldsymbol{w}x + \boldsymbol{b})$ and $\hat{p} = \boldsymbol{c}_p^\top\sigma(\boldsymbol{w}x + \boldsymbol{b})$, where $\boldsymbol{c}_p \in \mathbb{R}^K$ is a weight vector with respect to the output $\hat{p}$. We can see that the loss term of the BC does not change while that of the PDE becomes

$$\tilde{\mathcal{L}}_{\mathcal{F}}(\tilde{\boldsymbol{\theta}}) \approx \frac{1}{2\pi}\int_0^{2\pi} \Big[\big(\nabla\hat{p}(x) + a^2\sin(ax)\big)^2 + \big(\hat{p}(x) - \nabla\hat{u}(x)\big)^2\Big]\,\mathrm{d}x, \tag{A34}$$

where $\tilde{\boldsymbol{\theta}} = (\boldsymbol{c}, \boldsymbol{w}, \boldsymbol{b}, \boldsymbol{c}_p)$ is a set of trainable parameters.

A.7.2 Proof of Theorem 4.1

In this part, we provide detailed proof of Theorem 4.1.

We first derive the derivatives of the ansatz for the original PDEs (we recall that σ is $\tanh$ and we have $\sigma' = 1 - \sigma^2$)

$$\frac{\mathrm{d}\hat{u}}{\mathrm{d}x} = \boldsymbol{c}^\top \Big[\big(\mathbf{1} - \sigma^2(\boldsymbol{w}x + \boldsymbol{b})\big) \circ \boldsymbol{w} \Big], \tag{A35a}$$

$$\frac{\mathrm{d}^2\hat{u}}{\mathrm{d}x^2} = -2\boldsymbol{c}^\top \Big[\sigma(\boldsymbol{w}x + \boldsymbol{b}) \circ \big(\mathbf{1} - \sigma^2(\boldsymbol{w}x + \boldsymbol{b})\big) \circ \boldsymbol{w}^2 \Big]. \tag{A35b}$$

We will abbreviate $\sigma(\boldsymbol{w}x + \boldsymbol{b})$ as $\boldsymbol{\sigma}$, and then we have

$$\frac{\mathrm{d}\hat{u}}{\mathrm{d}x} = \boldsymbol{c}^\top \big[(\mathbf{1} - \boldsymbol{\sigma}^2) \circ \boldsymbol{w} \big], \tag{A36a}$$

$$\frac{\mathrm{d}^2\hat{u}}{\mathrm{d}x^2} = -2\boldsymbol{c}^\top \big[\boldsymbol{\sigma} \circ (\mathbf{1} - \boldsymbol{\sigma}^2) \circ \boldsymbol{w}^2 \big]. \tag{A36b}$$

Now, we can provide a bound for $(\partial \mathcal{L}_\mathcal{F}/\partial \boldsymbol{c})^\top$ as

$$\begin{aligned}
\left| \Big(\frac{\partial \mathcal{L}_\mathcal{F}}{\partial \boldsymbol{c}} \Big)^\top \right| &= \frac{1}{2\pi} \left| \int_0^{2\pi} 2\Big(\frac{\mathrm{d}^2\hat{u}}{\mathrm{d}x^2} + a^2 \sin(ax) \Big) \cdot \Big(\partial \Big(\frac{\mathrm{d}^2\hat{u}}{\mathrm{d}x^2} \Big) \Big/ \partial \boldsymbol{c} \Big) \, \mathrm{d}x \right| \\
&= \frac{2}{\pi} \left| \int_0^{2\pi} \Big(-2\boldsymbol{c}^\top \big[\boldsymbol{\sigma} \circ (\mathbf{1} - \boldsymbol{\sigma}^2) \circ \boldsymbol{w}^2 \big] + a^2 \sin(ax) \Big) \right. \\
&\qquad \left. \cdot \big[\boldsymbol{\sigma} \circ (\mathbf{1} - \boldsymbol{\sigma}^2) \circ \boldsymbol{w}^2 \big] \, \mathrm{d}x \right| \\
&\le \frac{2}{\pi} \left(\int_0^{2\pi} \Big(-2\boldsymbol{c}^\top \big[\boldsymbol{\sigma} \circ (\mathbf{1} - \boldsymbol{\sigma}^2) \circ \boldsymbol{w}^2 \big] + a^2 \sin(ax) \Big)^2 \mathrm{d}x \right)^{\frac{1}{2}} \\
&\qquad \cdot \left(\int_0^{2\pi} \big[\boldsymbol{\sigma} \circ (\mathbf{1} - \boldsymbol{\sigma}^2) \circ \boldsymbol{w}^2 \big]^2 \mathrm{d}x \right)^{\frac{1}{2}} \qquad (\mathrm{Cauchy - Schwarz}) \\
&\le \frac{2}{\pi} \left(\int_0^{2\pi} \big(2|\boldsymbol{c}|^\top \boldsymbol{w}^2 + a^2 \big)^2 \mathrm{d}x \right)^{\frac{1}{2}} \cdot \left(\int_0^{2\pi} \boldsymbol{w}^4 \, \mathrm{d}x \right)^{\frac{1}{2}} \\
&= 4\big(2|\boldsymbol{c}|^\top \boldsymbol{w}^2 + a^2 \big) \boldsymbol{w}^2 \\
&\le 8\big(|\boldsymbol{c}|^\top \boldsymbol{w}^2 + a^2 \big) \boldsymbol{w}^2,
\end{aligned} \tag{A37}$$

where $\le$ between two vectors is an element-wise comparison. Thus, $(\partial \mathcal{L}_\mathcal{F}/\partial \boldsymbol{c})^\top$ can be bounded by

$$\left| \Big(\frac{\partial \mathcal{L}_\mathcal{F}}{\partial \boldsymbol{c}} \Big)^\top \right| = \mathcal{O}\big(|\boldsymbol{c}|^\top \boldsymbol{w}^2 + a^2 \big) \cdot \boldsymbol{w}^2. \tag{A38}$$

Similarly, for $(\partial\mathcal{L}_{\mathcal{F}}/\partial\boldsymbol{w})^\top$, we have

$$\begin{aligned}
\left|\Big(\frac{\partial\mathcal{L}_{\mathcal{F}}}{\partial\boldsymbol{w}}\Big)^\top\right| &= \frac{1}{2\pi}\left|\int_0^{2\pi} 2\Big(\frac{\mathrm{d}^2\hat{u}}{\mathrm{d}x^2}+a^2\sin(ax)\Big)\cdot\Big(\partial\Big(\frac{\mathrm{d}^2\hat{u}}{\mathrm{d}x^2}\Big)\Big/\partial\boldsymbol{w}\Big)\,\mathrm{d}x\right| \\
&= \frac{2}{\pi}\left|\int_0^{2\pi}\Big(-2\boldsymbol{c}^\top\big[\boldsymbol{\sigma}\circ(\mathbf{1}-\boldsymbol{\sigma}^2)\circ\boldsymbol{w}^2\big]+a^2\sin(ax)\Big)\right. \\
&\qquad\left.\cdot\,\boldsymbol{c}\circ\big[2\boldsymbol{\sigma}\circ(\mathbf{1}-\boldsymbol{\sigma}^2)\circ\boldsymbol{w}+x(\mathbf{1}-3\boldsymbol{\sigma}^2)\circ(\mathbf{1}-\boldsymbol{\sigma}^2)\circ\boldsymbol{w}^2\big]\,\mathrm{d}x\right| \\
&\le \frac{2}{\pi}\left(\int_0^{2\pi}\Big(-2\boldsymbol{c}^\top\big[\boldsymbol{\sigma}\circ(\mathbf{1}-\boldsymbol{\sigma}^2)\circ\boldsymbol{w}^2\big]+a^2\sin(ax)\Big)^2\,\mathrm{d}x\right)^{\frac{1}{2}} \\
&\qquad\cdot\left(\int_0^{2\pi}\boldsymbol{c}^2\circ\big[2\boldsymbol{\sigma}\circ(\mathbf{1}-\boldsymbol{\sigma}^2)\circ\boldsymbol{w}\right. \\
&\qquad\qquad\left.+x(\mathbf{1}-3\boldsymbol{\sigma}^2)\circ(\mathbf{1}-\boldsymbol{\sigma}^2)\circ\boldsymbol{w}^2\big]^2\,\mathrm{d}x\right)^{\frac{1}{2}} \\
&\le \frac{2}{\pi}\left(\int_0^{2\pi}\big(2|\boldsymbol{c}|^\top\boldsymbol{w}^2+a^2\big)^2\,\mathrm{d}x\right)^{\frac{1}{2}}\cdot\left(\int_0^{2\pi}\boldsymbol{c}^2\circ(\boldsymbol{w}+2\pi\boldsymbol{w}^2)^2\,\mathrm{d}x\right)^{\frac{1}{2}} \\
&= 4\big(2|\boldsymbol{c}|^\top\boldsymbol{w}^2+a^2\big)\cdot|\boldsymbol{c}|\circ(|\boldsymbol{w}|+2\pi\boldsymbol{w}^2) \\
&\le 16\pi\big(|\boldsymbol{c}|^\top\boldsymbol{w}^2+a^2\big)\cdot|\boldsymbol{c}|\circ(|\boldsymbol{w}|+\boldsymbol{w}^2).
\end{aligned}\tag{A39}$$

Thus, the bound for $(\partial\mathcal{L}_{\mathcal{F}}/\partial\boldsymbol{w})^\top$ is given by

$$\left|\Big(\frac{\partial\mathcal{L}_{\mathcal{F}}}{\partial\boldsymbol{w}}\Big)^\top\right| = \mathcal{O}\big(|\boldsymbol{c}|^\top\boldsymbol{w}^2+a^2\big)\cdot\big(|\boldsymbol{c}|\circ|\boldsymbol{w}|\circ(|\boldsymbol{w}|+\mathbf{1})\big).\tag{A40}$$

And for $(\partial\mathcal{L}_{\mathcal{F}}/\partial\boldsymbol{b})^\top$, we have

$$\begin{aligned}
\left|\Big(\frac{\partial\mathcal{L}_{\mathcal{F}}}{\partial\boldsymbol{b}}\Big)^\top\right| &= \frac{1}{2\pi}\left|\int_0^{2\pi} 2\Big(\frac{\mathrm{d}^2\hat{u}}{\mathrm{d}x^2}+a^2\sin(ax)\Big)\cdot\Big(\partial\Big(\frac{\mathrm{d}^2\hat{u}}{\mathrm{d}x^2}\Big)\Big/\partial\boldsymbol{b}\Big)\,\mathrm{d}x\right| \\
&= \frac{2}{\pi}\left|\int_0^{2\pi}\Big(-2\boldsymbol{c}^\top\big[\boldsymbol{\sigma}\circ(\mathbf{1}-\boldsymbol{\sigma}^2)\circ\boldsymbol{w}^2\big]+a^2\sin(ax)\Big)\right. \\
&\qquad\left.\cdot\,\big[\boldsymbol{c}\circ(\mathbf{1}-3\boldsymbol{\sigma}^2)\circ(\mathbf{1}-\boldsymbol{\sigma}^2)\circ\boldsymbol{w}^2\big]\,\mathrm{d}x\right| \\
&\le \frac{2}{\pi}\left(\int_0^{2\pi}\Big(-2\boldsymbol{c}^\top\big[\boldsymbol{\sigma}\circ(\mathbf{1}-\boldsymbol{\sigma}^2)\circ\boldsymbol{w}^2\big]+a^2\sin(ax)\Big)^2\,\mathrm{d}x\right)^{\frac{1}{2}} \\
&\qquad\cdot\left(\int_0^{2\pi}\big[\boldsymbol{c}\circ(\mathbf{1}-3\boldsymbol{\sigma}^2)\circ(\mathbf{1}-\boldsymbol{\sigma}^2)\circ\boldsymbol{w}^2\big]^2\,\mathrm{d}x\right)^{\frac{1}{2}} \\
&\le \frac{2}{\pi}\left(\int_0^{2\pi}\big(2|\boldsymbol{c}|^\top\boldsymbol{w}^2+a^2\big)^2\,\mathrm{d}x\right)^{\frac{1}{2}}\cdot\left(\int_0^{2\pi}\boldsymbol{c}^2\circ\boldsymbol{w}^4\,\mathrm{d}x\right)^{\frac{1}{2}} \\
&= 4\big(2|\boldsymbol{c}|^\top\boldsymbol{w}^2+a^2\big)\cdot|\boldsymbol{c}|\circ\boldsymbol{w}^2 \\
&\le 8\big(|\boldsymbol{c}|^\top\boldsymbol{w}^2+a^2\big)\cdot|\boldsymbol{c}|\circ\boldsymbol{w}^2.
\end{aligned}\tag{A41}$$

Thus, the bound for $(\partial\mathcal{L}_{\mathcal{F}}/\partial\boldsymbol{b})^\top$ is given by

$$\left|\Big(\frac{\partial\mathcal{L}_{\mathcal{F}}}{\partial\boldsymbol{b}}\Big)^\top\right| = \mathcal{O}\big(|\boldsymbol{c}|^\top\boldsymbol{w}^2+a^2\big)\cdot\big(|\boldsymbol{c}|\circ\boldsymbol{w}^2\big).\tag{A42}$$

Recalling that $\boldsymbol{\theta} = (\boldsymbol{c}, \boldsymbol{w}, \boldsymbol{b})$, from Eq. (A38), Eq. (A40), and Eq. (A42), we have

$$\left|\left(\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathcal{F}}\right)^{\top}\right| = \left|\left(\frac{\partial \mathcal{L}_{\mathcal{F}}}{\partial \boldsymbol{\theta}}\right)^{\top}\right| = \mathcal{O}\big(|\boldsymbol{c}|^{\top}\boldsymbol{w}^2 + a^2\big) \cdot \big(\boldsymbol{w}^2, |\boldsymbol{c}| \circ |\boldsymbol{w}| \circ (|\boldsymbol{w}| + \mathbf{1}), |\boldsymbol{c}| \circ \boldsymbol{w}^2\big). \tag{A43}$$

In contrast, for the transformed PDE, we first derive the derivatives of the ansatz (i.e., $\hat{u} = \boldsymbol{c}^{\top}\sigma(\boldsymbol{w}x + \boldsymbol{b}), \hat{p} = \boldsymbol{c}_p^{\top}\sigma(\boldsymbol{w}x + \boldsymbol{b})$)

$$\frac{\mathrm{d}\hat{u}}{\mathrm{d}x} = \boldsymbol{c}^{\top}\big[(\mathbf{1} - \sigma^2(\boldsymbol{w}x + \boldsymbol{b})) \circ \boldsymbol{w}\big], \tag{A44a}$$

$$\frac{\mathrm{d}\hat{p}}{\mathrm{d}x} = \boldsymbol{c}_p^{\top}\big[(\mathbf{1} - \sigma^2(\boldsymbol{w}x + \boldsymbol{b})) \circ \boldsymbol{w}\big]. \tag{A44b}$$

We again abbreviate $\sigma(\boldsymbol{w}x + \boldsymbol{b})$ as $\boldsymbol{\sigma}$ to obtain

$$\frac{\mathrm{d}\hat{u}}{\mathrm{d}x} = \boldsymbol{c}^{\top}\big[(\mathbf{1} - \boldsymbol{\sigma}^2) \circ \boldsymbol{w}\big], \tag{A45a}$$

$$\frac{\mathrm{d}\hat{p}}{\mathrm{d}x} = \boldsymbol{c}_p^{\top}\big[(\mathbf{1} - \boldsymbol{\sigma}^2) \circ \boldsymbol{w}\big]. \tag{A45b}$$

We now compute a bound for $(\partial\tilde{\mathcal{L}}_{\mathcal{F}}/\partial\boldsymbol{c})^{\top}$

$$\begin{aligned}
\left|\left(\frac{\partial\tilde{\mathcal{L}}_{\mathcal{F}}}{\partial\boldsymbol{c}}\right)^{\top}\right| &= \frac{1}{2\pi}\left|\int_0^{2\pi} 2\Big(\hat{p} - \frac{\mathrm{d}\hat{u}}{\mathrm{d}x}\Big) \cdot \Big(\partial\Big(\frac{\mathrm{d}\hat{u}}{\mathrm{d}x}\Big)\Big/\partial\boldsymbol{c}\Big)\,\mathrm{d}x\right| \\
&= \frac{1}{\pi}\left|\int_0^{2\pi} \Big(\boldsymbol{c}_p^{\top}\boldsymbol{\sigma} - \boldsymbol{c}^{\top}\big[(\mathbf{1} - \boldsymbol{\sigma}^2) \circ \boldsymbol{w}\big]\Big) \cdot \big[(\mathbf{1} - \boldsymbol{\sigma}^2) \circ \boldsymbol{w}\big]\,\mathrm{d}x\right| \\
&\leq \frac{1}{\pi}\left(\int_0^{2\pi} \Big(\boldsymbol{c}_p^{\top}\boldsymbol{\sigma} - \boldsymbol{c}^{\top}\big[(\mathbf{1} - \boldsymbol{\sigma}^2) \circ \boldsymbol{w}\big]\Big)^2\,\mathrm{d}x\right)^{\frac{1}{2}} \cdot \left(\int_0^{2\pi} \big[(\mathbf{1} - \boldsymbol{\sigma}^2) \circ \boldsymbol{w}\big]^2\,\mathrm{d}x\right)^{\frac{1}{2}} \\
&\leq \frac{1}{\pi}\left(\int_0^{2\pi} \big(\|\boldsymbol{c}_p\|_1 + |\boldsymbol{c}|^{\top}|\boldsymbol{w}|\big)^2\,\mathrm{d}x\right)^{\frac{1}{2}} \cdot \left(\int_0^{2\pi} \boldsymbol{w}^2\,\mathrm{d}x\right)^{\frac{1}{2}} \\
&= 2\big(\|\boldsymbol{c}_p\|_1 + |\boldsymbol{c}|^{\top}|\boldsymbol{w}|\big)|\boldsymbol{w}|,
\end{aligned} \tag{A46}$$

Thus, the bound for $(\partial\tilde{\mathcal{L}}_{\mathcal{F}}/\partial\boldsymbol{c})^{\top}$ is given by

$$\left|\left(\frac{\partial\tilde{\mathcal{L}}_{\mathcal{F}}}{\partial\boldsymbol{c}}\right)^{\top}\right| = \mathcal{O}\big(\|\boldsymbol{c}_p\|_1 + |\boldsymbol{c}|^{\top}|\boldsymbol{w}|\big) \cdot |\boldsymbol{w}|. \tag{A47}$$

As for $(\partial\tilde{\mathcal{L}}_{\mathcal{F}}/\partial\boldsymbol{w})^\top$, we have

$$
\begin{aligned}
&\left|\Big(\frac{\partial\tilde{\mathcal{L}}_{\mathcal{F}}}{\partial\boldsymbol{w}}\Big)^\top\right| \\
=&\frac{1}{2\pi}\left|\int_0^{2\pi}2\Big(\frac{\mathrm{d}\hat{p}}{\mathrm{d}x}+a^2\sin(ax)\Big)\cdot\Big(\partial\Big(\frac{\mathrm{d}\hat{p}}{\mathrm{d}x}\Big)\Big/\partial\boldsymbol{w}\Big)\,\mathrm{d}x+\int_0^{2\pi}2\Big(\hat{p}-\frac{\mathrm{d}\hat{u}}{\mathrm{d}x}\Big)\cdot\Big(\frac{\partial\hat{p}}{\partial\boldsymbol{w}}-\partial\Big(\frac{\mathrm{d}\hat{u}}{\mathrm{d}x}\Big)\Big/\partial\boldsymbol{w}\Big)\,\mathrm{d}x\right| \\
=&\frac{1}{\pi}\left|\int_0^{2\pi}\Big(\boldsymbol{c}_p^\top\big[(\mathbf{1}-\boldsymbol{\sigma}^2)\circ\boldsymbol{w}\big]+a^2\sin(ax)\Big)\cdot\big[\boldsymbol{c}_p\circ(\mathbf{1}-\boldsymbol{\sigma}^2)\circ(\mathbf{1}-2x\boldsymbol{\sigma}\circ\boldsymbol{w})\big]\,\mathrm{d}x\right. \\
&\left.+\int_0^{2\pi}\Big(\boldsymbol{c}_p^\top\boldsymbol{\sigma}-\boldsymbol{c}^\top\big[(\mathbf{1}-\boldsymbol{\sigma}^2)\circ\boldsymbol{w}\big]\Big)\cdot\big[x\boldsymbol{c}_p\circ(\mathbf{1}-\boldsymbol{\sigma}^2)-\boldsymbol{c}\circ(\mathbf{1}-\boldsymbol{\sigma}^2)\circ(\mathbf{1}-2x\boldsymbol{\sigma}\circ\boldsymbol{w})\big]\,\mathrm{d}x\right| \\
\le&\frac{1}{\pi}\Bigg(\Big(\int_0^{2\pi}\Big(\boldsymbol{c}_p^\top\big[(\mathbf{1}-\boldsymbol{\sigma}^2)\circ\boldsymbol{w}\big]+a^2\sin(ax)\Big)^2\,\mathrm{d}x\Big)^{\frac{1}{2}} \\
&\qquad\cdot\Big(\int_0^{2\pi}\big[\boldsymbol{c}_p\circ(\mathbf{1}-\boldsymbol{\sigma}^2)\circ(\mathbf{1}-2x\boldsymbol{\sigma}\circ\boldsymbol{w})\big]^2\,\mathrm{d}x\Big)^{\frac{1}{2}} \\
&\quad+\Big(\int_0^{2\pi}\Big(\boldsymbol{c}_p^\top\boldsymbol{\sigma}-\boldsymbol{c}^\top\big[(\mathbf{1}-\boldsymbol{\sigma}^2)\circ\boldsymbol{w}\big]\Big)^2\,\mathrm{d}x\Big)^{\frac{1}{2}} \\
&\qquad\cdot\Big(\int_0^{2\pi}\big[x\boldsymbol{c}_p\circ(\mathbf{1}-\boldsymbol{\sigma}^2)-\boldsymbol{c}\circ(\mathbf{1}-\boldsymbol{\sigma}^2)\circ(\mathbf{1}-2x\boldsymbol{\sigma}\circ\boldsymbol{w})\big]^2\,\mathrm{d}x\Big)^{\frac{1}{2}}\Bigg) \\
\le&\,4\Bigg(\Big(\int_0^{2\pi}\big(|\boldsymbol{c}_p|^\top|\boldsymbol{w}|+a^2\big)^2\,\mathrm{d}x\Big)^{\frac{1}{2}}\cdot\Big(\int_0^{2\pi}\boldsymbol{c}_p^2\circ(\mathbf{1}+|\boldsymbol{w}|)^2\,\mathrm{d}x\Big)^{\frac{1}{2}} \\
&\quad+\Big(\int_0^{2\pi}\big(\|\boldsymbol{c}_p\|_1+|\boldsymbol{c}|^\top|\boldsymbol{w}|\big)^2\,\mathrm{d}x\Big)^{\frac{1}{2}}\cdot\Big(\int_0^{2\pi}\big[|\boldsymbol{c}_p|+|\boldsymbol{c}|\circ(\mathbf{1}+|\boldsymbol{w}|)\big]^2\,\mathrm{d}x\Big)^{\frac{1}{2}}\Bigg) \\
=&\,8\pi\Big(\big(|\boldsymbol{c}_p|^\top|\boldsymbol{w}|+a^2\big)\cdot\big[|\boldsymbol{c}_p|\circ(\mathbf{1}+|\boldsymbol{w}|)\big]+\big(\|\boldsymbol{c}_p\|_1+|\boldsymbol{c}|^\top|\boldsymbol{w}|\big)\cdot\big[|\boldsymbol{c}_p|+|\boldsymbol{c}|\circ(\mathbf{1}+|\boldsymbol{w}|)\big]\Big) \\
\le&\,40\pi\big(\|\boldsymbol{c}_p\|_1+\max(|\boldsymbol{c}|,|\boldsymbol{c}_p|)^\top|\boldsymbol{w}|+a^2\big)\cdot\big[\max(|\boldsymbol{c}|,|\boldsymbol{c}_p|)\circ\max(|\boldsymbol{w}|,\mathbf{1})\big].
\end{aligned}
\tag{A48}
$$

Thus, we can bound $(\partial\tilde{\mathcal{L}}_{\mathcal{F}}/\partial\boldsymbol{w})^\top$ by

$$
\begin{aligned}
\left|\Big(\frac{\partial\tilde{\mathcal{L}}_{\mathcal{F}}}{\partial\boldsymbol{w}}\Big)^\top\right|&\le 40\pi\big(\|\boldsymbol{c}_p\|_1+\max(|\boldsymbol{c}|,|\boldsymbol{c}_p|)^\top|\boldsymbol{w}|+a^2\big)\cdot\big[\max(|\boldsymbol{c}|,|\boldsymbol{c}_p|)\circ\max(|\boldsymbol{w}|,\mathbf{1})\big] \\
&=\mathcal{O}\big(\|\boldsymbol{c}_p\|_1+\max(|\boldsymbol{c}|,|\boldsymbol{c}_p|)^\top|\boldsymbol{w}|+a^2\big)\cdot\big[\max(|\boldsymbol{c}|,|\boldsymbol{c}_p|)\circ\max(|\boldsymbol{w}|,\mathbf{1})\big].
\end{aligned}
\tag{A49}
$$

And for $(\partial\tilde{\mathcal{L}}_{\mathcal{F}}/\partial\boldsymbol{b})^\top$, we have

$$
\begin{aligned}
&\left|\Big(\frac{\partial\tilde{\mathcal{L}}_{\mathcal{F}}}{\partial\boldsymbol{b}}\Big)^\top\right| \\
=&\frac{1}{2\pi}\bigg|\int_0^{2\pi}2\Big(\frac{\mathrm{d}\hat{p}}{\mathrm{d}x}+a^2\sin(ax)\Big)\cdot\Big(\partial\Big(\frac{\mathrm{d}\hat{p}}{\mathrm{d}x}\Big)\Big/\partial\boldsymbol{b}\Big)\,\mathrm{d}x \\
&\qquad+\int_0^{2\pi}2\Big(\hat{p}-\frac{\mathrm{d}\hat{u}}{\mathrm{d}x}\Big)\cdot\Big(\frac{\partial\hat{p}}{\partial\boldsymbol{b}}-\partial\Big(\frac{\mathrm{d}\hat{u}}{\mathrm{d}x}\Big)\Big/\partial\boldsymbol{b}\Big)\,\mathrm{d}x\bigg| \\
=&\frac{1}{\pi}\bigg|\int_0^{2\pi}\Big(\boldsymbol{c}_p^\top\big[(\mathbf{1}-\boldsymbol{\sigma}^2)\circ\boldsymbol{w}\big]+a^2\sin(ax)\Big)\cdot\big[-2\boldsymbol{c}_p\circ(\mathbf{1}-\boldsymbol{\sigma}^2)\circ\boldsymbol{\sigma}\circ\boldsymbol{w}\big]\,\mathrm{d}x \\
&+\int_0^{2\pi}\Big(\boldsymbol{c}_p^\top\boldsymbol{\sigma}-\boldsymbol{c}^\top\big[(\mathbf{1}-\boldsymbol{\sigma}^2)\circ\boldsymbol{w}\big]\Big)\cdot\big[\boldsymbol{c}_p\circ(\mathbf{1}-\boldsymbol{\sigma}^2)+2\boldsymbol{c}\circ(\mathbf{1}-\boldsymbol{\sigma}^2)\circ\boldsymbol{\sigma}\circ\boldsymbol{w}\big]\,\mathrm{d}x\bigg| \\
\le&\frac{1}{\pi}\Bigg(\bigg(\int_0^{2\pi}\Big(\boldsymbol{c}_p^\top\big[(\mathbf{1}-\boldsymbol{\sigma}^2)\circ\boldsymbol{w}\big]+a^2\sin(ax)\Big)^2\,\mathrm{d}x\bigg)^{\frac{1}{2}} \\
&\qquad\cdot\bigg(\int_0^{2\pi}\big[-2\boldsymbol{c}_p\circ(\mathbf{1}-\boldsymbol{\sigma}^2)\circ\boldsymbol{\sigma}\circ\boldsymbol{w}\big]^2\,\mathrm{d}x\bigg)^{\frac{1}{2}} \\
&+\bigg(\int_0^{2\pi}\Big(\boldsymbol{c}_p^\top\boldsymbol{\sigma}-\boldsymbol{c}^\top\big[(\mathbf{1}-\boldsymbol{\sigma}^2)\circ\boldsymbol{w}\big]\Big)^2\,\mathrm{d}x\bigg)^{\frac{1}{2}} \\
&\qquad\cdot\bigg(\int_0^{2\pi}\big[\boldsymbol{c}_p\circ(\mathbf{1}-\boldsymbol{\sigma}^2)+2\boldsymbol{c}\circ(\mathbf{1}-\boldsymbol{\sigma}^2)\circ\boldsymbol{\sigma}\circ\boldsymbol{w}\big]^2\,\mathrm{d}x\bigg)^{\frac{1}{2}}\Bigg) \\
\le&\frac{2}{\pi}\Bigg(\bigg(\int_0^{2\pi}\big(|\boldsymbol{c}_p|^\top|\boldsymbol{w}|+a^2\big)^2\,\mathrm{d}x\bigg)^{\frac{1}{2}}\cdot\bigg(\int_0^{2\pi}\boldsymbol{c}_p^2\circ\boldsymbol{w}^2\,\mathrm{d}x\bigg)^{\frac{1}{2}} \\
&+\bigg(\int_0^{2\pi}\big(\|\boldsymbol{c}_p\|_1+|\boldsymbol{c}|^\top|\boldsymbol{w}|\big)^2\,\mathrm{d}x\bigg)^{\frac{1}{2}}\cdot\bigg(\int_0^{2\pi}\big[|\boldsymbol{c}_p|+|\boldsymbol{c}|\circ|\boldsymbol{w}|\big]^2\,\mathrm{d}x\bigg)^{\frac{1}{2}}\Bigg) \\
=&4\Big(\big(|\boldsymbol{c}_p|^\top|\boldsymbol{w}|+a^2\big)\cdot\big[|\boldsymbol{c}_p|\circ|\boldsymbol{w}|\big]+\big(\|\boldsymbol{c}_p\|_1+|\boldsymbol{c}|^\top|\boldsymbol{w}|\big)\cdot\big[|\boldsymbol{c}_p|+|\boldsymbol{c}|\circ|\boldsymbol{w}|\big]\Big) \\
\le&12\big(\|\boldsymbol{c}_p\|_1+\max(|\boldsymbol{c}|,|\boldsymbol{c}_p|)^\top|\boldsymbol{w}|+a^2\big)\cdot\big[\max(|\boldsymbol{c}|,|\boldsymbol{c}_p|)\circ\max(|\boldsymbol{w}|,\mathbf{1})\big].
\end{aligned}
\tag{A50}
$$

Thus, we can bound $(\partial\tilde{\mathcal{L}}_{\mathcal{F}}/\partial\boldsymbol{b})^\top$ by

$$
\begin{aligned}
\left|\Big(\frac{\partial\tilde{\mathcal{L}}_{\mathcal{F}}}{\partial\boldsymbol{b}}\Big)^\top\right|\le&12\big(\|\boldsymbol{c}_p\|_1+\max(|\boldsymbol{c}|,|\boldsymbol{c}_p|)^\top|\boldsymbol{w}|+a^2\big)\cdot\big[\max(|\boldsymbol{c}|,|\boldsymbol{c}_p|)\circ\max(|\boldsymbol{w}|,\mathbf{1})\big] \\
=&\mathcal{O}\big(\|\boldsymbol{c}_p\|_1+\max(|\boldsymbol{c}|,|\boldsymbol{c}_p|)^\top|\boldsymbol{w}|+a^2\big)\cdot\big[\max(|\boldsymbol{c}|,|\boldsymbol{c}_p|)\circ\max(|\boldsymbol{w}|,\mathbf{1})\big].
\end{aligned}
\tag{A51}
$$

For $(\partial\tilde{\mathcal{L}}_{\mathcal{F}}/\partial\boldsymbol{c}_p)^\top$, we have

$$
\begin{aligned}
&\left|\Big(\frac{\partial\tilde{\mathcal{L}}_{\mathcal{F}}}{\partial\boldsymbol{c}_p}\Big)^\top\right| \\
=&\frac{1}{2\pi}\left|\int_0^{2\pi} 2\Big(\frac{\mathrm{d}\hat{p}}{\mathrm{d}x}+a^2\sin(ax)\Big)\cdot\Big(\partial\Big(\frac{\mathrm{d}\hat{p}}{\mathrm{d}x}\Big)\Big/\partial\boldsymbol{c}_p\Big)\,\mathrm{d}x+\int_0^{2\pi}2\Big(\hat{p}-\frac{\mathrm{d}\hat{u}}{\mathrm{d}x}\Big)\cdot\Big(\frac{\partial\hat{p}}{\partial\boldsymbol{c}_p}\Big)\,\mathrm{d}x\right| \\
=&\frac{1}{\pi}\left|\int_0^{2\pi}\Big(\boldsymbol{c}_p^\top\big[(\mathbf{1}-\boldsymbol{\sigma}^2)\circ\boldsymbol{w}\big]+a^2\sin(ax)\Big)\cdot\big[(\mathbf{1}-\boldsymbol{\sigma}^2)\circ\boldsymbol{w}\big]\,\mathrm{d}x\right. \\
&\qquad\left.+\int_0^{2\pi}\Big(\boldsymbol{c}_p^\top\boldsymbol{\sigma}-\boldsymbol{c}^\top\big[(\mathbf{1}-\boldsymbol{\sigma}^2)\circ\boldsymbol{w}\big]\Big)\cdot\boldsymbol{\sigma}\,\mathrm{d}x\right| \\
\le&\frac{1}{\pi}\Bigg(\Big(\int_0^{2\pi}\Big(\boldsymbol{c}_p^\top\big[(\mathbf{1}-\boldsymbol{\sigma}^2)\circ\boldsymbol{w}\big]+a^2\sin(ax)\Big)^2\mathrm{d}x\Big)^{\frac{1}{2}}\cdot\Big(\int_0^{2\pi}\big[(\mathbf{1}-\boldsymbol{\sigma}^2)\circ\boldsymbol{w}\big)\big]^2\,\mathrm{d}x\Big)^{\frac{1}{2}} \\
&\qquad+\Big(\int_0^{2\pi}\Big(\boldsymbol{c}_p^\top\boldsymbol{\sigma}-\boldsymbol{c}^\top\big[(\mathbf{1}-\boldsymbol{\sigma}^2)\circ\boldsymbol{w}\big]\Big)^2\mathrm{d}x\Big)^{\frac{1}{2}}\cdot\Big(\int_0^{2\pi}\boldsymbol{\sigma}^2\,\mathrm{d}x\Big)^{\frac{1}{2}}\Bigg) \\
\le&\frac{1}{\pi}\Bigg(\Big(\int_0^{2\pi}\big(|\boldsymbol{c}_p|^\top|\boldsymbol{w}|+a^2\big)^2\mathrm{d}x\Big)^{\frac{1}{2}}\cdot\Big(\int_0^{2\pi}\boldsymbol{w}^2\,\mathrm{d}x\Big)^{\frac{1}{2}} \\
&\qquad+\Big(\int_0^{2\pi}\big(\|\boldsymbol{c}_p\|_1+|\boldsymbol{c}|^\top|\boldsymbol{w}|\big)^2\mathrm{d}x\Big)^{\frac{1}{2}}\cdot\Big(\int_0^{2\pi}\mathbf{1}\,\mathrm{d}x\Big)^{\frac{1}{2}}\Bigg) \\
=&2\Big(\big(|\boldsymbol{c}_p|^\top|\boldsymbol{w}|+a^2\big)\cdot|\boldsymbol{w}|+\big(\|\boldsymbol{c}_p\|_1+|\boldsymbol{c}|^\top|\boldsymbol{w}|\big)\cdot\mathbf{1}\Big) \\
\le&4\big(\|\boldsymbol{c}_p\|_1+\max(|\boldsymbol{c}|,|\boldsymbol{c}_p|)^\top|\boldsymbol{w}|+a^2\big)\cdot\max(|\boldsymbol{w}|,\mathbf{1}).
\end{aligned}
\tag{A52}
$$

Thus, we can bound $(\partial\tilde{\mathcal{L}}_{\mathcal{F}}/\partial\boldsymbol{c}_p)^\top$ by

$$
\begin{aligned}
\left|\Big(\frac{\partial\tilde{\mathcal{L}}_{\mathcal{F}}}{\partial\boldsymbol{c}_p}\Big)^\top\right|\le&4\left(\|\boldsymbol{c}_p\|_1+\max(|\boldsymbol{c}|,|\boldsymbol{c}_p|)^\top|\boldsymbol{w}|+a^2\right)\cdot\max(|\boldsymbol{w}|,\mathbf{1}) \\
=&\mathcal{O}\big(\|\boldsymbol{c}_p\|_1+\max(|\boldsymbol{c}|,|\boldsymbol{c}_p|)^\top|\boldsymbol{w}|+a^2\big)\cdot\max(|\boldsymbol{w}|,\mathbf{1}).
\end{aligned}
\tag{A53}
$$

Recalling that $\tilde{\boldsymbol{\theta}}=(\boldsymbol{c},\boldsymbol{w},\boldsymbol{b},\boldsymbol{c}_p)$, from Eq. (A47), Eq. (A49), Eq. (A51), and Eq. (A53), noting that $\|\boldsymbol{c}_p\|_1+|\boldsymbol{c}|^\top|\boldsymbol{w}|\le\|\boldsymbol{c}_p\|_1+\max(|\boldsymbol{c}|,|\boldsymbol{c}_p|)^\top|\boldsymbol{w}|+a^2$, we have

$$
\begin{aligned}
\left|\big(\nabla_{\tilde{\boldsymbol{\theta}}}\tilde{\mathcal{L}}_{\mathcal{F}}\big)^\top\right|=\left|\Big(\frac{\partial\tilde{\mathcal{L}}_{\mathcal{F}}}{\partial\tilde{\boldsymbol{\theta}}}\Big)^\top\right|=&\mathcal{O}\big(\|\boldsymbol{c}_p\|_1+\max(|\boldsymbol{c}|,|\boldsymbol{c}_p|)^\top|\boldsymbol{w}|+a^2\big)\cdot\big(|\boldsymbol{w}|, \\
&\max(|\boldsymbol{c}|,|\boldsymbol{c}_p|)\circ\max(|\boldsymbol{w}|,\mathbf{1}),\max(|\boldsymbol{c}|,|\boldsymbol{c}_p|)\circ\max(|\boldsymbol{w}|,\mathbf{1}),\max(|\boldsymbol{w}|,\mathbf{1})\big).
\end{aligned}
\tag{A54}
$$

A.7.3 Analysis via the Condition Number

In addition to providing bounds for the gradients of $\mathcal{L}_{\mathcal{F}}$ and $\tilde{\mathcal{L}}_{\mathcal{F}}$, we can also use the condition number to characterize the sensitivity of their gradients with respect to the parameters of the neural network. Let $\boldsymbol{\theta}^{(t)}=(\boldsymbol{c}^{(t)},\boldsymbol{w}^{(t)},\boldsymbol{b}^{(t)})$ and $\tilde{\boldsymbol{\theta}}^{(t)}=(\boldsymbol{c}^{(t)},\boldsymbol{w}^{(t)},\boldsymbol{b}^{(t)},\boldsymbol{c}_p^{(t)})$ are the parameters of the neural network in the tth step (before and after the reformulation). For simplicity, we introduce the following notations

$$\Delta\boldsymbol{\theta}=\boldsymbol{\theta}^{(t+1)}-\boldsymbol{\theta}^{(t)},\tag{A55a}$$

$$\Delta\tilde{\boldsymbol{\theta}}=\tilde{\boldsymbol{\theta}}^{(t+1)}-\tilde{\boldsymbol{\theta}}^{(t)},\tag{A55b}$$

$$\Delta\mathcal{L}_{\mathcal{F}}=\mathcal{L}_{\mathcal{F}}(\boldsymbol{\theta}^{(t+1)})-\mathcal{L}_{\mathcal{F}}(\boldsymbol{\theta}^{(t)}),\tag{A55c}$$

$$\Delta\tilde{\mathcal{L}}_{\mathcal{F}}=\tilde{\mathcal{L}}_{\mathcal{F}}(\tilde{\boldsymbol{\theta}}^{(t+1)})-\tilde{\mathcal{L}}_{\mathcal{F}}(\tilde{\boldsymbol{\theta}}^{(t)}).\tag{A55d}$$

The condition numbers of $\mathcal{L}_{\mathcal{F}}$ and $\tilde{\mathcal{L}}_{\mathcal{F}}$ are defined as

$$\mathrm{cond} = \frac{|\Delta \mathcal{L}_{\mathcal{F}}|}{\|\Delta \boldsymbol{\theta}\|_2}, \quad \tilde{\mathrm{cond}} = \frac{|\Delta \tilde{\mathcal{L}}_{\mathcal{F}}|}{\|\Delta \tilde{\boldsymbol{\theta}}\|_2}. \tag{A56}$$

Next we derive the bounds for cond and $\tilde{\mathrm{cond}}$, respectively. We first consider cond

$$\begin{aligned}
\mathrm{cond} = \frac{|\Delta \mathcal{L}_{\mathcal{F}}|}{\|\Delta \boldsymbol{\theta}\|_2} &\approx \left| \left(\frac{\partial \mathcal{L}_{\mathcal{F}}}{\partial \boldsymbol{\theta}}\right)^{\top} \cdot \Delta \boldsymbol{\theta} \right| \Big/ \|\Delta \boldsymbol{\theta}\|_2 \leq \left\| \left(\frac{\partial \mathcal{L}_{\mathcal{F}}}{\partial \boldsymbol{\theta}}\right)^{\top} \right\|_2 \\
&= \mathcal{O}\Big(\left(|\boldsymbol{c}|^{\top} \boldsymbol{w}^2 + a^2\right) \cdot \left\| \boldsymbol{w}^2, |\boldsymbol{c}| \circ |\boldsymbol{w}| \circ (|\boldsymbol{w}| + \mathbf{1}), |\boldsymbol{c}| \circ \boldsymbol{w}^2 \right\|_2 \Big) \quad \text{(Eq. (A43))} \\
&= \mathcal{O}\Big(\left(|\boldsymbol{c}|^{\top} \boldsymbol{w}^2 + a^2\right) \cdot \left\| \boldsymbol{w}^2, |\boldsymbol{c}| \circ |\boldsymbol{w}| \circ (|\boldsymbol{w}| + \mathbf{1}), |\boldsymbol{c}| \circ \boldsymbol{w}^2 \right\|_1 \Big) \quad \text{(Equivalence of norms)} \\
&= \mathcal{O}\Big(\left(|\boldsymbol{c}|^{\top} \boldsymbol{w}^2 + a^2\right) \cdot \left(\|\boldsymbol{w}^2\|_1 + \||\boldsymbol{c}| \circ |\boldsymbol{w}| \circ (|\boldsymbol{w}| + \mathbf{1})\|_1 + \||\boldsymbol{c}| \circ \boldsymbol{w}^2\|_1 \right) \Big) \\
&= \mathcal{O}\Big(\left(|\boldsymbol{c}|^{\top} \boldsymbol{w}^2 + a^2\right) \cdot \left\| \max(|\boldsymbol{c}|, \mathbf{1}) \circ |\boldsymbol{w}| \circ \max(|\boldsymbol{w}|, \mathbf{1}) \right\|_1 \Big).
\end{aligned} \tag{A57}$$

Similarly, for $\tilde{\mathrm{cond}}$, we have

$$\begin{aligned}
\tilde{\mathrm{cond}} = \frac{|\Delta \tilde{\mathcal{L}}_{\mathcal{F}}|}{\|\Delta \tilde{\boldsymbol{\theta}}\|_2} &\approx \left| \left(\frac{\partial \tilde{\mathcal{L}}_{\mathcal{F}}}{\partial \tilde{\boldsymbol{\theta}}}\right)^{\top} \cdot \Delta \tilde{\boldsymbol{\theta}} \right| \Big/ \|\Delta \tilde{\boldsymbol{\theta}}\|_2 \leq \left\| \left(\frac{\partial \tilde{\mathcal{L}}_{\mathcal{F}}}{\partial \tilde{\boldsymbol{\theta}}}\right)^{\top} \right\|_2 \\
&= \mathcal{O}\Big(\left(\|\boldsymbol{c}_p\|_1 + \max(|\boldsymbol{c}|, |\boldsymbol{c}_p|)^{\top} |\boldsymbol{w}| + a^2\right) \cdot \Big\| |\boldsymbol{w}|, \max(|\boldsymbol{c}|, |\boldsymbol{c}_p|) \circ \max(|\boldsymbol{w}|, \mathbf{1}), \\
&\qquad \max(|\boldsymbol{c}|, |\boldsymbol{c}_p|) \circ \max(|\boldsymbol{w}|, \mathbf{1}), \max(|\boldsymbol{w}|, \mathbf{1}) \Big\|_2 \Big) \qquad \text{(Eq. (A54))} \\
&= \mathcal{O}\Big(\left(\|\boldsymbol{c}_p\|_1 + \max(|\boldsymbol{c}|, |\boldsymbol{c}_p|)^{\top} |\boldsymbol{w}| + a^2\right) \cdot \Big\| |\boldsymbol{w}|, \max(|\boldsymbol{c}|, |\boldsymbol{c}_p|) \circ \max(|\boldsymbol{w}|, \mathbf{1}), \\
&\qquad \max(|\boldsymbol{c}|, |\boldsymbol{c}_p|) \circ \max(|\boldsymbol{w}|, \mathbf{1}), \max(|\boldsymbol{w}|, \mathbf{1}) \Big\|_1 \Big) \qquad \text{(Equivalence of norms)} \\
&= \mathcal{O}\Big(\left(\|\boldsymbol{c}_p\|_1 + \max(|\boldsymbol{c}|, |\boldsymbol{c}_p|)^{\top} |\boldsymbol{w}| + a^2\right) \cdot \big(\|\boldsymbol{w}\|_1 \\
&\qquad + \|\max(|\boldsymbol{c}|, |\boldsymbol{c}_p|) \circ \max(|\boldsymbol{w}|, \mathbf{1})\|_1 + \|\max(|\boldsymbol{c}|, |\boldsymbol{c}_p|) \circ \max(|\boldsymbol{w}|, \mathbf{1})\|_1 \\
&\qquad + \|\max(|\boldsymbol{w}|, \mathbf{1})\|_1 \big) \Big) \\
&= \mathcal{O}\Big(\left(\|\boldsymbol{c}_p\|_1 + \max(|\boldsymbol{c}|, |\boldsymbol{c}_p|)^{\top} |\boldsymbol{w}| + a^2\right) \cdot \left\| \max(|\boldsymbol{c}|, |\boldsymbol{c}_p|, \mathbf{1}) \circ \max(|\boldsymbol{w}|, \mathbf{1}) \right\|_1 \Big).
\end{aligned} \tag{A58}$$

From Eq. (A57) and Eq. (A58), we find that for the original PDEs, the condition number has a higher order relationship with respect to $\boldsymbol{\theta}$. If $\boldsymbol{\theta}$ is large, the condition number can be very large, leading to oscillations in training. However, if $\boldsymbol{\theta}$ is small, the condition number will also be very small, resulting in smaller changes of $\boldsymbol{\theta}$ between adjacent iterations and therefore slower convergence. In contrast, after the reformulation, the condition number has a lower order relationship with respect to $\tilde{\boldsymbol{\theta}}$, which keeps the condition number more stable during training and alleviates this problem.

A.8 Experimental details

In the following, we will briefly introduce some essential details of our experiments, including the experimental environment, hyper-parameters, construction of the ansatz, and details of the governing PDEs in each experiment. We first introduce the experimental environment, while other details are put into the subsections corresponding to the experiments.

Experimental environment We use PyTorch [4] as our deep learning library. And our codes for the physics-informed learning are based on DeepXDE [3]. We train all the models except domain decomposition based baselines (i.e., xPINN, FBPINN, and PFNN-2) on one NVIDIA TITAN Xp 12GB GPU, while the other three are trained on eight NVIDIA GeForce RTX 3090 24GB GPUs (since domain decomposition based models consist of several sub-networks and require more memory to be stored). The operating system is Ubuntu 18.04.5 LTS. If the analytical solution is unavailable, the ground truth solutions to the PDEs (i.e., the testing data) will be generated by COMSOL Multiphysics, a FEM commercial software. And we have put the generated testing data into the zip file.

A.8.1 Simulation of a 2D battery pack (Heat Equation)

Governing PDEs The governing PDEs (along with boundary/initial conditions) are given by

$$\frac{\partial T}{\partial t} = k\Delta T(\boldsymbol{x}, t), \qquad \boldsymbol{x} \in \Omega, t \in (0, 1], \tag{A59a}$$

$$k\big(\boldsymbol{n}(\boldsymbol{x}) \cdot \nabla T(\boldsymbol{x}, t)\big) = h\big(T_a - T(\boldsymbol{x}, t)\big), \qquad \boldsymbol{x} \in \gamma_{\text{outer}}, t \in (0, 1], \tag{A59b}$$

$$k\big(\boldsymbol{n}(\boldsymbol{x}) \cdot \nabla T(\boldsymbol{x}, t)\big) = h\big(T_c - T(\boldsymbol{x}, t)\big), \qquad \boldsymbol{x} \in \gamma_{\text{cell},i}, t \in (0, 1], \quad i = 1, \dots, n_c, \tag{A59c}$$

$$k\big(\boldsymbol{n}(\boldsymbol{x}) \cdot \nabla T(\boldsymbol{x}, t)\big) = h\big(T_w - T(\boldsymbol{x}, t)\big), \qquad \boldsymbol{x} \in \gamma_{\text{pipe},i}, t \in (0, 1], \quad i = 1, \dots, n_w, \tag{A59d}$$

$$T(\boldsymbol{x}, 0) = T_0, \qquad \boldsymbol{x} \in \Omega, \tag{A59e}$$

where $\boldsymbol{x} = (x_1, x_2)$, t are the spatial and temporal coordinates, respectively, $T(\boldsymbol{x}, t)$ is the temperature over time, $k = 1$ is the thermal conductivity, $\Delta T = \partial^2 T/\partial x_1^2 + \partial^2 T/\partial x_2^2$, $h = 1$ is the heat transfer coefficient, $\nabla T = (\partial T/\partial x_1, \partial T/\partial x_2)$, $T_a = 0.1$, $T_c = 5$, $T_w = 1$ are, respectively, the temperature of the air, the cells ($n_c = 11$ cells of radius $r_c = 1$), the cooling pipes ($n_w = 6$ pipes of radius $r_w = 0.4$), $T_0 = 0.1$ is the initial temperature, and the geometry (i.e., Ω, γ_{outer}, etc) is shown in Figure 3(a).

And the reformulated PDEs are (which is used by the proposed model, HC)

$$\frac{\partial T}{\partial t} = k\big(\nabla \cdot \boldsymbol{p}(\boldsymbol{x}, t)\big), \qquad \boldsymbol{x} \in \Omega, t \in (0, 1], \tag{A60a}$$

$$\boldsymbol{p}(\boldsymbol{x}, t) = \nabla T, \qquad \boldsymbol{x} \in \Omega \cup \partial\Omega, t \in (0, 1], \tag{A60b}$$

$$k\big(\boldsymbol{n}(\boldsymbol{x}) \cdot \boldsymbol{p}(\boldsymbol{x}, t)\big) = h\big(T_a - T(\boldsymbol{x}, t)\big), \qquad \boldsymbol{x} \in \gamma_{\text{outer}}, t \in (0, 1], \tag{A60c}$$

$$k\big(\boldsymbol{n}(\boldsymbol{x}) \cdot \boldsymbol{p}(\boldsymbol{x}, t)\big) = h\big(T_c - T(\boldsymbol{x}, t)\big), \qquad \boldsymbol{x} \in \gamma_{\text{cell},i}, t \in (0, 1], \quad i = 1, \dots, n_c, \tag{A60d}$$

$$k\big(\boldsymbol{n}(\boldsymbol{x}) \cdot \boldsymbol{p}(\boldsymbol{x}, t)\big) = h\big(T_w - T(\boldsymbol{x}, t)\big), \qquad \boldsymbol{x} \in \gamma_{\text{pipe},i}, t \in (0, 1], \quad i = 1, \dots, n_w, \tag{A60e}$$

$$T(\boldsymbol{x}, 0) = T_0, \qquad \boldsymbol{x} \in \Omega, \tag{A60f}$$

where $\boldsymbol{p}(\boldsymbol{x}, t)$ is the introduced extra field.

Construction of the Ansatz Since the solution is a scalar function (i.e., $T(\boldsymbol{x}, t)$), we directly denote the solution by $u(\boldsymbol{x}, t) = T(\boldsymbol{x}, t)$. Let $\tilde{\boldsymbol{p}}$ denote $(u, \boldsymbol{p})$. We first derive the general solutions at $\gamma_{\text{outer}}, \gamma_{\text{cell},i}, \gamma_{\text{pipe},i}$, respectively.

For $\boldsymbol{x} \in \gamma_{\text{outer}}$, we have $a(\boldsymbol{x}) = h$, $b(\boldsymbol{x}) = k$, $g(\boldsymbol{x}) = hT_a$. According to Eq. (10), the general solution $\tilde{\boldsymbol{p}}^{\gamma_{\text{outer}}}$ is given by

$$\tilde{\boldsymbol{p}}^{\gamma_{\text{outer}}} = \boldsymbol{B}(\boldsymbol{x})\text{NN}^{\gamma_{\text{outer}}}(\boldsymbol{x}, t) + \frac{(h, k\boldsymbol{n})}{\sqrt{h^2 + k^2}} \frac{hT_a}{\sqrt{h^2 + k^2}}, \tag{A61}$$

where $\boldsymbol{B}(\boldsymbol{x})$ is computed in Eq. (A7) (with $\tilde{\boldsymbol{n}} = (\tilde{n}_1, \tilde{n}_2, \tilde{n}_3) = (h, k\boldsymbol{n})/\sqrt{h^2 + k^2}$). And for $\boldsymbol{x} \in \gamma_{\text{cell},i}$ and $\gamma_{\text{pipe},i}$, the derivation is similar, where we only need to change T_a to T_c and T_w, respectively.

Then, we gather all the general solutions computed to form our ansatz $(\hat{u}, \hat{\boldsymbol{p}})$ according to Eq. (A30), where $\{\gamma_{\text{outer}}, \gamma_{\text{cell},1}, \dots, \gamma_{\text{cell},n_c}, \gamma_{\text{pipe},1}, \dots, \gamma_{\text{pipe},n_w}\}$ are reordered as $\{\gamma_i\}_{i=1}^{1+n_c+n_w}$ and $f(\boldsymbol{x}) = T_0$.

Choices of Extended Distance Functions For $\gamma_{\text{cell},i}$ and $\gamma_{\text{pipe},i}$, since they are 2D circles, we can directly choose the extended distance functions $l^{\gamma_{\text{cell},i}}(\boldsymbol{x})$ and $l^{\gamma_{\text{pipe},i}}(\boldsymbol{x})$ as the distance between $\boldsymbol{x}$ and the center minus the radius. For the rectangular γ_{outer}, supposing that it is given by $[a_1, a_2] \times [b_1, b_2]$, we construct the extended distance function $l^{\gamma_{\text{outer}}}$ as follows

$$l^{\gamma_{\text{outer}}}(\boldsymbol{x}) = \text{SoftMin}(x_1 - a_1, a_2 - x_1, x_2 - b_1, b_2 - x_2), \tag{A62}$$

where $\boldsymbol{x} = (x_1, x_2)$, SoftMin is a differentiable version of min function which is implemented by `LogSumExp` in PyTorch, i.e., $\text{SoftMin}(\boldsymbol{y}) = \text{LogSumExp}(-\beta\boldsymbol{y})/(-\beta)$, $\beta = 4$. And the extended function $l^{\partial\Omega}(\boldsymbol{x})$ is computed by taking the SoftMin of the distances to all the boundaries

$$l^{\partial\Omega}(\boldsymbol{x}) = \text{SoftMin}(l^{\gamma_{\text{outer}}}(\boldsymbol{x}), l^{\gamma_{\text{cell},1}}(\boldsymbol{x}), \dots, l^{\gamma_{\text{cell},n_c}}(\boldsymbol{x}), l^{\gamma_{\text{pipe},1}}(\boldsymbol{x}), \dots, l^{\gamma_{\text{pipe},n_w}}(\boldsymbol{x})). \tag{A63}$$

Implementation All the models are trained for 5000 Adam iterations (with a learning rate scheduler of `ReduceLROnPlateau` from PyTorch and an initial learning rate of 0.01), followed by a L-BFGS optimization until convergence. Unless otherwise specified, the mean squared error (MSE) is used for the loss function and $\tanh$ is used for the activation function. And the hyper-parameters of each model are listed as follow

- **HC**: The main neural network is a multilayer perceptron (MLP) of size $[3] + 4 \times [50] + [3]$ (which means 3 inputs, 4 hidden layers of width 50, and 3 outputs). The sub-networks (corresponding to Eq. (A60c), Eq. (A60d), and Eq. (A60e)) are all MLPs of size $[3] + 3 \times [20] + [3]$. And the hyper-parameters of "hardness" are $\beta_s = 5$ and $\beta_t = 10$.
- **PINN**: The ansatz is an MLP of size $[3] + 4 \times [50] + [1]$.
- **PINN-LA**: The weights of the loss terms corresponding to the BCs are approximated by $\hat{\lambda}_i = \max_{\theta_n}\{|\nabla_\theta \mathcal{L}_r(\theta_n)|\} \Big/ \overline{|\nabla_\theta \lambda_i \mathcal{L}_i(\theta_n)|}$. And the parameter of the moving average is $\alpha = 0.1$, which is recommended by the paper [7]. Besides, the parameters of the PINN are the same as above.
- **PINN-LA-2**: In our modified version, we approximate the weights of the loss terms as $\hat{\lambda}_i = \overline{|\nabla_\theta \mathcal{L}_r(\theta_n)|} \Big/ \overline{|\nabla_\theta \lambda_i \mathcal{L}_i(\theta_n)|}$. And the parameter of the moving average is also $\alpha = 0.1$. Here we replace the maximum with the mean to make the weights of the loss terms more stable during the training process.
- **FBPINN**: The domain of the problem is divided into $4 \times 6 = 24$ subdomains by a regular grid. The size of the sub-network, an MLP, corresponding to each subdomain is $[3] + 3 \times [30] + [1]$. And the scale factor is $\sigma = 0.4$, chosen so that the window function is close to zero outside the overlapping region of the subdomains.
- **xPINN**: The domain of the problem is divided into $4 \times 6 = 24$ subdomains by a regular grid. The size of the sub-network corresponding to each subdomain is $[3] + 3 \times [30] + [1]$. And the loss terms of the interface condition include average solution as well as residual continuity conditions.
- **PFNN**: The PFNN considers the variational formulation of the Eq. (A59) (i.e., Ritz formulation), and embed the initial condition (IC) into its ansatz (similar to Eq. (5)). And the size of the neural network (an MLP) is $[3] + 4 \times [50] + [1]$.
- **PFNN-2**: The PFNN-2 replaces a single neural network with a domain decomposition based neural network on the basis of PFNN. In the original literature [6], the domain is decomposed in a hard way (like xPINN). However, in our experiments (see Table 1), we find that the performance of hard decomposition is relatively poor, which is because new loss terms are needed to maintain the continuity of the ansatz at the interfaces between the sub-domains, which further aggravates the unbalanced competition. To overcome this, we instead employ a soft domain decomposition, as in FBPINN. See the parts of PFNN and FBPINN for the values of hyper-parameters.

A.8.2 Simulation of an Airfoil (Navier-Stokes Equations)

Governing PDEs The governing PDEs are given by

$$\begin{aligned}
\boldsymbol{u}(\boldsymbol{x}) \cdot \nabla \boldsymbol{u}(\boldsymbol{x}) &= -\nabla p(\boldsymbol{x}) + v \nabla^2 \boldsymbol{u}(\boldsymbol{x}), && \boldsymbol{x} \in \Omega, && \text{(A64a)}\\
\nabla \cdot \boldsymbol{u}(\boldsymbol{x}) &= 0, && \boldsymbol{x} \in \Omega, && \text{(A64b)}\\
\boldsymbol{u}(\boldsymbol{x}) &= \boldsymbol{u}_0(\boldsymbol{x}), && \boldsymbol{x} \in \gamma_{\text{inlet}} \cup \gamma_{\text{top}} \cup \gamma_{\text{bottom}}, && \text{(A64c)}\\
p(\boldsymbol{x}) &= 1, && \boldsymbol{x} \in \gamma_{\text{outlet}}, && \text{(A64d)}\\
\boldsymbol{n}(\boldsymbol{x}) \cdot \boldsymbol{u}(\boldsymbol{x}) &= 0, && \boldsymbol{x} \in \gamma_{\text{airfoil}}, && \text{(A64e)}
\end{aligned}$$

where $\boldsymbol{u}(\boldsymbol{x}) = (u_1(\boldsymbol{x}), u_2(\boldsymbol{x}))$, $p(\boldsymbol{x})$, $v = 1/50$ are the velocity, pressure, and viscosity of the fluid, respectively, $\boldsymbol{u}_0(\boldsymbol{x}) = (1, 0)$, and the geometry of the problem (i.e., Ω, γ_{inlet}, etc) is shown in Figure 3(b).

And the reformulated PDEs are (which is used by the proposed model, HC)

$$\big(\boldsymbol{u}(\boldsymbol{x}) - v\nabla\big) \cdot \big(\boldsymbol{p}_1(\boldsymbol{x}), \boldsymbol{p}_2(\boldsymbol{x})\big) = -\nabla p(\boldsymbol{x}), \qquad \boldsymbol{x} \in \Omega, \tag{A65a}$$

$$\nabla \cdot \boldsymbol{u}(\boldsymbol{x}) = 0, \qquad \boldsymbol{x} \in \Omega, \tag{A65b}$$

$$\boldsymbol{p}_1(\boldsymbol{x}) = \nabla u_1, \qquad \boldsymbol{x} \in \Omega \cup \partial\Omega, \tag{A65c}$$

$$\boldsymbol{p}_2(\boldsymbol{x}) = \nabla u_2, \qquad \boldsymbol{x} \in \Omega \cup \partial\Omega, \tag{A65d}$$

$$\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{u}_0(\boldsymbol{x}), \qquad \boldsymbol{x} \in \gamma_{\text{inlet}} \cup \gamma_{\text{top}} \cup \gamma_{\text{bottom}}, \tag{A65e}$$

$$p(\boldsymbol{x}) = 1, \qquad \boldsymbol{x} \in \gamma_{\text{outlet}}, \tag{A65f}$$

$$\boldsymbol{n}(\boldsymbol{x}) \cdot \boldsymbol{u}(\boldsymbol{x}) = 0, \qquad \boldsymbol{x} \in \gamma_{\text{airfoil}}, \tag{A65g}$$

where $\boldsymbol{p}_1(\boldsymbol{x})$ and $\boldsymbol{p}_2(\boldsymbol{x})$ are the introduced extra fields.

Construction of the Ansatz Here, the solution is $(\boldsymbol{u}(\boldsymbol{x}), p(\boldsymbol{x}))$. For $p(\boldsymbol{x})$, the general solution in γ_{outlet} is exactly $p^{\gamma_{\text{outlet}}}(\boldsymbol{x}) = 1$. And the ansatz for p is given by

$$\hat{p} = p^{\gamma_{\text{outlet}}}(\boldsymbol{x}) + l^{\gamma_{\text{outlet}}}(\boldsymbol{x})\text{NN}_{\text{main}}(\boldsymbol{x})[3], \tag{A66}$$

where $[3]$ means taking the third elements of the output of $\text{NN}_{\text{main}}(\boldsymbol{x})$.

For $\boldsymbol{u}(\boldsymbol{x})$, the general solution in $\gamma_{\text{inlet}} \cup \gamma_{\text{top}} \cup \gamma_{\text{bottom}}$ is exactly $\boldsymbol{u}^{\gamma_*}(\boldsymbol{x}) = \boldsymbol{u}_0(\boldsymbol{x})$, where we define an alias γ_* for $\gamma_{\text{inlet}} \cup \gamma_{\text{top}} \cup \gamma_{\text{bottom}}$. In γ_{airfoil}, the general solution is given by

$$\boldsymbol{u}^{\gamma_{\text{airfoil}}} = \boldsymbol{B}(\boldsymbol{x})\text{NN}^{\gamma_*}(\boldsymbol{x}), \tag{A67}$$

where $\boldsymbol{B}(\boldsymbol{x}) = [n_2(\boldsymbol{x}), -n_1(\boldsymbol{x})]^\top$ according to Eq. (A5) and the output of $\text{NN}^{\gamma_*}(\boldsymbol{x})$ is a scalar. Gathering $\boldsymbol{u}^{\gamma_*}$ and $\boldsymbol{u}^{\gamma_{\text{airfoil}}}$, we then follow Eq. (11) to obtain the ansatz for $\hat{\boldsymbol{u}}$

$$\hat{\boldsymbol{u}} = l^{\partial\Omega}(\boldsymbol{x})\text{NN}_{\text{main}}(\boldsymbol{x})[1:2] + \exp\big[-\alpha_{\gamma_*} l^{\gamma_*}(\boldsymbol{x})\big]\boldsymbol{u}^{\gamma_*}(\boldsymbol{x}) + \exp\big[-\alpha_{\gamma_{\text{airfoil}}} l^{\gamma_{\text{airfoil}}}(\boldsymbol{x})\big]\boldsymbol{u}^{\gamma_{\text{airfoil}}}(\boldsymbol{x}), \tag{A68}$$

where $[1:2]$ means taking the first two elements of the output of $\text{NN}_{\text{main}}(\boldsymbol{x})$ and α_{γ_*} as well as $\alpha_{\gamma_{\text{airfoil}}}$ are similarly defined as in Eq. (12).

Choices of Extended Distance Functions For $l^{\gamma_{\text{airfoil}}}(\boldsymbol{x})$, a direct way is to calculate the distance between $\boldsymbol{x}$ and the airfoil γ_{airfoil}. However, it may be very time consuming since the γ_{airfoil} is highly complicated. So we prefer to approximate the true distance with an MLP with 3 hidden layers of width 30. We train the neural network before training our main model with 1024×6 points sampled in Ω ($5/6$ of them are sampled in the bounding box of the airfoil, and the rest are sampled in Ω) along with their truth distances (which are computed by using the formula of the distance to a polygon) for $10,000$ Adam epochs (with a learning rate scheduler of `ReduceLROnPlateau` from PyTorch and an initial learning rate of 0.001). The loss function is a ℓ_1 loss. A polar coordinate transformation trick is utilized as in Appendix A.5.

And for γ_{outlet}, since it is a vertical line, for example, $x_1 = a$, we can compute the extended distance function as $l^{\gamma_{\text{outlet}}}(\boldsymbol{x}) = a - x_1$, where $\boldsymbol{x} = (x_1, x_2)$. And γ_* is an open rectangle, so we can compute $l^{\gamma_*}(\boldsymbol{x})$ similarly to the case of the rectangle (see Eq. (A62)) while ignoring the right side. Besides, $l^{\partial\Omega}(\boldsymbol{x})$ is still computed by taking the SoftMin of the distances to all the boundaries.

Implementation All the models are trained for 5000 Adam iterations (with a learning rate scheduler of `ReduceLROnPlateau` from PyTorch and an initial learning rate of 0.001), followed by a L-BFGS optimization until convergence. And the hyper-parameters of each model are listed as follow

- **HC**: The main neural network is an MLP of size $[2] + 6 \times [50] + [7]$. The sub-network (corresponding to Eq. (A65g)) is an MLP of size $[2]+4\times[40]+[1]$. And the hyper-parameters of "hardness" is $\beta_s = 5$.
- **PINN**: The ansatz is an MLP of size $[2] + 6 \times [50] + [3]$.
- **PINN-LA**: The parameter of the moving average is $\alpha = 0.1$.
- **PINN-LA-2**: The parameter of the moving average is $\alpha = 0.1$.
- **FBPINN**: The domain of the problem is divided into $3\times6 = 18$ subdomains by a regular grid. The size of the sub-network, an MLP, corresponding to each subdomain is $[2]+4\times[30]+[3]$. And the scale factor is $\sigma = 0.2$, chosen so that the window function is close to zero outside the overlapping region of the subdomains.

- **xPINN**: The domain of the problem is divided into $3 \times 6 = 18$ subdomains by a regular grid. The size of the sub-network corresponding to each subdomain is $[2] + 4 \times [30] + [3]$. And the loss terms of the interface condition include average solution as well as residual continuity conditions.

A.8.3 High-dimensional Heat Equation

Governing PDEs The governing PDEs are given by

$$\frac{\partial u}{\partial t} = k\Delta u(\boldsymbol{x}, t) + f(\boldsymbol{x}, t), \qquad \boldsymbol{x} \in \Omega \subset \mathbb{R}^d, t \in (0, 1], \tag{A69a}$$

$$\boldsymbol{n}(\boldsymbol{x}) \cdot \nabla u(\boldsymbol{x}, t) = g(\boldsymbol{x}, t), \qquad \boldsymbol{x} \in \partial\Omega, t \in (0, 1], \tag{A69b}$$

$$u(\boldsymbol{x}, 0) = g(\boldsymbol{x}, 0), \qquad \boldsymbol{x} \in \Omega, \tag{A69c}$$

where u is the quantity of interest, $k = 1/d$, $f(\boldsymbol{x}, t) = -k|\boldsymbol{x}|^2 \exp(0.5|\boldsymbol{x}|^2 + t)$, $d = 10$, Ω is a unit ball (i.e., $\Omega = \{|\boldsymbol{x}| \leq 1\}$), and $g(\boldsymbol{x}, t) = \exp(0.5|\boldsymbol{x}|^2 + t)$ which is also the analytical solution to above PDEs.

And the reformulated PDEs are (which is used by the proposed model, HC)

$$\frac{\partial u}{\partial t} = k\big(\nabla \cdot \boldsymbol{p}(\boldsymbol{x}, t)\big) + f(\boldsymbol{x}, t), \qquad \boldsymbol{x} \in \Omega \subset \mathbb{R}^d, t \in (0, 1], \tag{A70a}$$

$$\boldsymbol{p}(\boldsymbol{x}, t) = \nabla u, \qquad \boldsymbol{x} \in \Omega, t \in (0, 1], \tag{A70b}$$

$$\boldsymbol{n}(\boldsymbol{x}) \cdot \boldsymbol{p}(\boldsymbol{x}, t) = g(\boldsymbol{x}, t), \qquad \boldsymbol{x} \in \partial\Omega, t \in (0, 1], \tag{A70c}$$

$$u(\boldsymbol{x}, 0) = g(\boldsymbol{x}, 0), \qquad \boldsymbol{x} \in \Omega, \tag{A70d}$$

where $\boldsymbol{p}(\boldsymbol{x}, t)$ is the introduced extra field.

Construction of the Ansatz The solution to the PDEs is a scalar function u and there is only one boundary $\partial\Omega = \{|\boldsymbol{x}| = 1\}$. We now derive the general solution $\boldsymbol{p}^{\partial\Omega}$ with respect to Eq. (A70c)

$$\boldsymbol{p}^{\partial\Omega} = \boldsymbol{B}(\boldsymbol{x})\mathrm{NN}^{\partial\Omega} + \boldsymbol{n}(\boldsymbol{x})g(\boldsymbol{x}, t), \tag{A71}$$

where $\boldsymbol{B}(\boldsymbol{x}) = \boldsymbol{I}_d - \boldsymbol{n}(\boldsymbol{x})\boldsymbol{n}(\boldsymbol{x})^\top$. And the ansatz $(\hat{u}, \hat{\boldsymbol{p}})$ is given by

$$\hat{\boldsymbol{p}} = l^{\partial\Omega}(\boldsymbol{x})\mathrm{NN}_{\mathrm{main}}(\boldsymbol{x}, t)[1:d] + \boldsymbol{p}^{\partial\Omega}(\boldsymbol{x}, t), \tag{A72a}$$

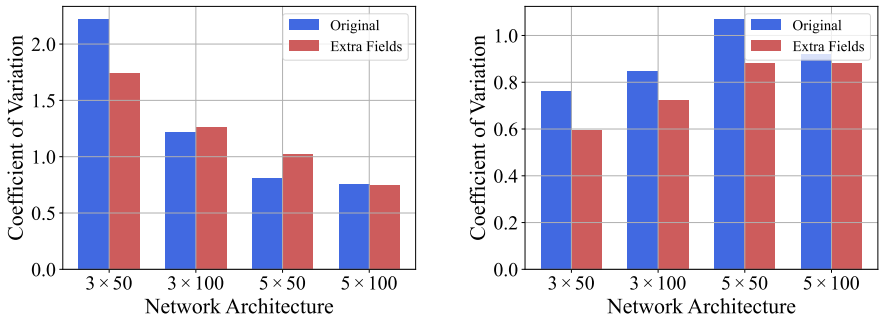
$$\hat{u} = \mathrm{NN}_{\mathrm{main}}(\boldsymbol{x}, t)[d+1]\big(1 - \exp[-\beta_t t]\big) + g(\boldsymbol{x}, 0)\exp[-\beta_t t], \tag{A72b}$$

where $[1:d]$ means taking the first d elements of the output of $\mathrm{NN}_{\mathrm{main}}(\boldsymbol{x}, t)$ while $[d+1]$ means the last element.

Choices of Extended Distance Functions Since $\partial\Omega$ is a ND sphere, we can compute $l^{\partial\Omega}(\boldsymbol{x})$ by subtracting the distance between $\boldsymbol{x}$ and the center from the radius (the symbol is different from the previous 2D circles, since $\partial\Omega$ is the outer boundary).

Implementation All the models are trained for 5000 Adam iterations (with a learning rate scheduler of `ReduceLROnPlateau` from PyTorch and an initial learning rate of 0.01), followed by a L-BFGS optimization until convergence. And the hyper-parameters of each model are listed as follow

- **HC**: The main neural network is an MLP of size $[11] + 4 \times [50] + [11]$. The sub-network (corresponding to Eq. (A70c)) is an MLP of size $[11] + 3 \times [20] + [10]$. And the hyper-parameters of "hardness" is $\beta_t = 10$ (here we only have one boundary, so we can construct our ansatz (in the spatial domain) as in Eq. (3) instead of Eq. (A30a) and β_s is no longer needed).
- **PINN**: The ansatz is an MLP of size $[11] + 4 \times [50] + [1]$.
- **PINN-LA**: The parameter of the moving average is $\alpha = 0.1$.
- **PINN-LA-2**: The parameter of the moving average is $\alpha = 0.1$.
- **PFNN**: The size of the neural network (an MLP) is $[11] + 4 \times [50] + [1]$.



(a) Poisson's equation (b) Schrödinger equation

Figure A2: The CV of all the values of $\overline{|\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathcal{F}}|}$ and $\overline{|\nabla_{\tilde{\boldsymbol{\theta}}} \tilde{\mathcal{L}}_{\mathcal{F}}|}$ during training.

A.8.4 Ablation Study: Extra fields

Here, our experiment is divided into two parts where we consider the Poisson's equation and the nonlinear Schrödinger equation, respectively. The relevant details are as follows

Poisson's Equation The governing PDEs are described as

$$\Delta u(x) = -a^2 \sin ax, \qquad x \in (0, 2\pi), \tag{A73a}$$
$$u(x) = 0, \qquad x = 0 \vee x = 2\pi, \tag{A73b}$$

where $a = 2$ and $u(x)$ is the physical quantity of interest.

And the reformulated PDEs are (corresponding to the *extra fields*)

$$\nabla p(x) = -a^2 \sin ax, \qquad x \in (0, 2\pi), \tag{A74a}$$
$$p(x) = \nabla u(x), \qquad x \in (0, 2\pi), \tag{A74b}$$
$$u(x) = 0, \qquad x = 0 \vee x = 2\pi. \tag{A74c}$$

where $p(x)$ is the introduced extra field.

The two models (PINNs with and without the *extra fields*) are trained with $N_f = 128$ collocation points and $N_b = 2$ boundary points for 10,000 Adam iterations (with a learning rate of 0.001). And we have tested different network architectures, including $[1] + 3 \times [50] + [\cdot]$, $[1] + 3 \times [100] + [\cdot]$, $[1] + 5 \times [50] + [\cdot]$, $[1] + 5 \times [100] + [\cdot]$, where for the PINN without the *extra fields*, the number of outputs is 1, and for the PINN with the *extra fields*, the number of outputs is 2.

Schrödinger Equation The governing PDEs are described as

$$i\frac{\partial h}{\partial t} + \frac{1}{2}\frac{\partial^2 h}{\partial x^2} + |h(x,t)|^2 h(x,t) = 0, \qquad x \in (-5, 5), t \in (0, \pi/2], \tag{A75a}$$
$$h(t, -5) = h(t, 5), \qquad t \in (0, \pi/2], \tag{A75b}$$
$$\frac{\partial h}{\partial x}(t, -5) = \frac{\partial h}{\partial x}(t, 5), \qquad t \in (0, \pi/2], \tag{A75c}$$
$$h(0, x) = 2\,\mathrm{sech}(x), \qquad x \in (-5, 5), \tag{A75d}$$

where $h(x, t)$ is the physical quantity of interest.

And the reformulated PDEs are (corresponding to the *extra fields*)

$$i\frac{\partial h}{\partial t} + \frac{1}{2}\frac{\partial p}{\partial x} + |h(x,t)|^2 h(x,t) = 0, \qquad x \in (-5, 5), t \in (0, \pi/2], \tag{A76a}$$
$$p(x,t) = \frac{\partial h}{\partial x}, \qquad x \in [-5, 5], t \in (0, \pi/2], \tag{A76b}$$
$$h(t, -5) = h(t, 5), \qquad t \in (0, \pi/2], \tag{A76c}$$
$$\frac{\partial h}{\partial x}(t, -5) = \frac{\partial h}{\partial x}(t, 5), \qquad t \in (0, \pi/2], \tag{A76d}$$
$$h(0, x) = 2\,\mathrm{sech}(x), \qquad x \in (-5, 5), \tag{A76e}$$

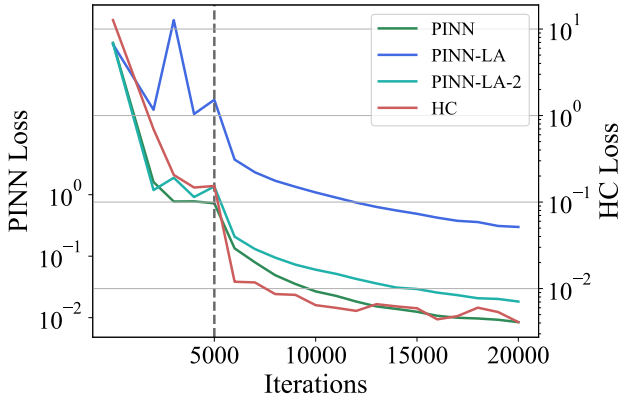


Figure A3: The convergence history of the simulation of a 2D battery pack.

where $p(x)$ is the introduced extra field.

The two models (PINNs with and without the *extra fields*) are trained with $N_f = 1000$ collocation points, $N_b = 20$ boundary points, and $N_i = 200$ initial points for 10,000 Adam iterations (with a learning rate of 0.001). And we have tested different network architectures, including $[1]+3\times[50]+[\cdot]$, $[1]+3\times[100]+[\cdot]$, $[1]+5\times[50]+[\cdot]$, $[1]+5\times[100]+[\cdot]$, where for the PINN without the *extra fields*, the number of outputs is 2, and for the PINN with the *extra fields*, the number of outputs is 4.

Experimental Results We report the ratio of the the moving variance (MovVar) of $\overline{|\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathcal{F}}|}$ to that of $\overline{|\nabla_{\tilde{\boldsymbol{\theta}}}\tilde{\mathcal{L}}_{\mathcal{F}}|}$ at each iteration during training, where the window size of the MovVar is 500 and after the MovVar, a moving average filter with a window size of 500 is applied. The results are shown in Figure 4. Besides, we also calculate the coefficient of variation (CV) of all the values of $\overline{|\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathcal{F}}|}$ and $\overline{|\nabla_{\tilde{\boldsymbol{\theta}}}\tilde{\mathcal{L}}_{\mathcal{F}}|}$, respectively. And we give the results in Figure A2. Using the CV as a criterion, we also find that the *extra fields* significantly reduces the gradient oscillations during training, especially for the complex nonlinear PDEs.

A.9 Empirical Analysis of Convergence

In this subsection, we will empirically analyze the convergence of our method as well as some representative baselines in the context of the simulation of a 2D battery pack (see Section 5.2). We now report the training history with respect to iterations in Figure A3. The left axis shows the loss of PINNs (including PINN, PINN-LA, and PINN-LA-2) while the right axis shows the loss of our method, HC. The PINN loss is computed by adding up all the loss terms (including the losses of PDEs and BCs), where the loss weights are ignored for PINN-LA and PINN-LA-2. The first 5000 iterations are trained with Adam (separated by the gray dotted line), and the last 15000 are trained with L-BFGS.

From the results in Figure A3, we can see that the loss functions of all models drop significantly after switching to L-BFGS. This shows that L-BFGS can further promote convergence through utilizing

Table 1: Parallel experimental results of the simulation of a 2D battery pack (MAE of T)

	$t = 0$	$t = 0.5$	$t = 1$	average
PINN	0.1232 ± 0.0219	0.0417 ± 0.0141	0.0263 ± 0.0078	0.0499 ± 0.0135
PINN-LA	0.1083 ± 0.0266	0.0927 ± 0.0372	0.1168 ± 0.0739	0.0969 ± 0.0385
PINN-LA-2	0.1065 ± 0.0059	0.0322 ± 0.0031	$\mathbf{0.0200} \pm 0.0020$	0.0400 ± 0.0031
FBPINN	0.0763 ± 0.0071	0.0258 ± 0.0037	0.0205 ± 0.0041	0.0318 ± 0.0027
xPINN	0.2085 ± 0.0252	0.1144 ± 0.0194	0.1352 ± 0.0241	0.1310 ± 0.0194
PFNN	$\mathbf{0.0000} \pm 0.0000$	0.3769 ± 0.0974	0.6012 ± 0.2274	0.3522 ± 0.1019
PFNN-2	$\mathbf{0.0000} \pm 0.0000$	0.3814 ± 0.0381	0.5247 ± 0.0394	0.3365 ± 0.0236
HC	$\mathbf{0.0000} \pm 0.0000$	$\mathbf{0.0244} \pm 0.0010$	0.0226 ± 0.0012	$\mathbf{0.0219} \pm 0.0007$

Table 2: Parallel experimental results of the simulation of a 2D battery pack (MAPE of T)

	$t=0$	$t=0.5$	$t=1$	average
PINN	$123.16 \pm 21.91\%$	$10.97 \pm 3.57\%$	$4.52 \pm 0.98\%$	$23.58 \pm 5.61\%$
PINN-LA	$108.14 \pm 26.57\%$	$24.15 \pm 8.07\%$	$17.98 \pm 8.97\%$	$33.64 \pm 8.89\%$
PINN-LA-2	$106.39 \pm 5.86\%$	$8.70 \pm 0.57\%$	**3.86** $\pm 0.36\%$	$19.56 \pm 1.00\%$
FBPINN	$76.25 \pm 7.06\%$	$7.69 \pm 0.68\%$	$5.26 \pm 0.71\%$	$15.07 \pm 0.57\%$
xPINN	$208.36 \pm 25.21\%$	$26.25 \pm 4.99\%$	$18.15 \pm 3.05\%$	$49.60 \pm 7.37\%$
PFNN	**0.02** $\pm 0.00\%$	$94.63 \pm 17.68\%$	$105.38 \pm 27.16\%$	$80.92 \pm 15.08\%$
PFNN-2	**0.02** $\pm 0.00\%$	$71.39 \pm 6.98\%$	$82.04 \pm 8.90\%$	$61.65 \pm 4.64\%$
HC	**0.02** $\pm 0.00\%$	**5.29** $\pm 0.16\%$	$3.87 \pm 0.14\%$	**5.22** $\pm 0.17\%$

Table 3: Parallel experimental results of the simulation of an airfoil (MAE)

	u_1	u_2	p
PINN	0.4234 ± 0.0809	0.0681 ± 0.0162	0.3204 ± 0.1404
PINN-LA	0.4467 ± 0.0450	0.0630 ± 0.0061	0.3028 ± 0.0480
PINN-LA-2	0.4542 ± 0.0875	0.0679 ± 0.0111	0.3230 ± 0.1115
FBPINN	0.3975 ± 0.0221	0.0544 ± 0.0030	0.2650 ± 0.0059
xPINN	0.6942 ± 0.0432	0.0581 ± 0.0013	1.1587 ± 0.1251
HC	**0.2824** ± 0.0215	**0.0435** ± 0.0024	**0.2144** ± 0.0114

the information of the second derivatives of the loss function. However, we may not start with L-BFGS because it can easily lead to divergence. We consider Adam+L-BFGS to be a practical choice. Furthermore, we find that the convergence of PINNs is negatively affected by the tricks of learning rate annealing algorithm, especially the PINN-LA without our modification. The HC has the fastest convergence rate among all models. This means that the hard-constraint method or extra fields may be helpful in accelerating convergence.

A.10 Parallel Experiments

In this subsection, we revisit the three experiments in Section 5.2$\sim$5.4 and perform parallel tests in 5 runs to assess the significance of the results. We report the testing results (along with the 95% confidence intervals) in Table 1$\sim$6. From the results, we can see that our method, HC still outperforms all the other baselines. Besides, HC has the least variation, which shows that the hard-constraint methods can improve the stability of training.

A.11 Ethics Statement

PDEs have important applications in many fields, including applied physics, automobile manufacturing, economics, and the aerospace industry. Solving PDE via neural networks has attracted much attention in recent years, and it may be applied to the above fields in the future. Our method also belongs to this kind. However, the method for solving PDE based on neural networks has no theoretical explanation and safety guarantee for the time being. Applying such methods to security-sensitive domains may lead to unexpected incidents and the cause of the accident may be hard to diagnose. Possible solutions include developing alternatives with theoretical interpretability or using safeguards.

Table 4: Parallel experimental results of the simulation of an airfoil (WMAPE)

	u_1	u_2	p
PINN	0.5358 ± 0.1024	1.1709 ± 0.2778	0.2921 ± 0.1279
PINN-LA	0.5653 ± 0.0570	1.0819 ± 0.1048	0.2760 ± 0.0437
PINN-LA-2	0.5747 ± 0.1106	1.1670 ± 0.1920	0.2944 ± 0.1016
FBPINN	0.5030 ± 0.0279	0.9347 ± 0.0517	0.2416 ± 0.0054
xPINN	0.8784 ± 0.0546	0.9986 ± 0.0225	1.0562 ± 0.1140
HC	**0.3573** ± 0.0272	**0.7472** ± 0.0418	**0.1954** ± 0.0104

Table 5: Parallel experimental results of the high-dimensional heat equation (MAE of u)

	$t=0$	$t=0.5$	$t=1$	average
PINN	0.0204 ± 0.0148	0.0357 ± 0.0104	0.1600 ± 0.0600	0.0525 ± 0.0173
PINN-LA	0.0430 ± 0.0751	0.3039 ± 0.6691	0.8011 ± 1.7228	0.3464 ± 0.7531
PINN-LA-2	0.0287 ± 0.0670	0.2071 ± 0.6524	0.5933 ± 1.7225	0.2455 ± 0.7433
PFNN	$\mathbf{0.0000} \pm 0.0000$	0.0895 ± 0.0727	0.2130 ± 0.1790	0.0963 ± 0.0788
HC	$\mathbf{0.0000} \pm 0.0000$	$\mathbf{0.0028} \pm 0.0006$	$\mathbf{0.0046} \pm 0.0008$	$\mathbf{0.0027} \pm 0.0006$

Table 6: Parallel experimental results of the high-dimensional heat equation (MAPE of u)

	$t=0$	$t=0.5$	$t=1$	average
PINN	$1.15 \pm 0.51\%$	$1.41 \pm 0.41\%$	$3.83 \pm 1.45\%$	$1.75 \pm 0.58\%$
PINN-LA	$2.86 \pm 5.07\%$	$12.06 \pm 26.54\%$	$19.35 \pm 41.65\%$	$11.46 \pm 24.75\%$
PINN-LA-2	$1.92 \pm 4.55\%$	$8.19 \pm 25.80\%$	$14.29 \pm 41.54\%$	$8.00 \pm 24.25\%$
PFNN	$\mathbf{0.00} \pm 0.00\%$	$3.59 \pm 2.93\%$	$5.19 \pm 4.36\%$	$3.20 \pm 2.60\%$
HC	$\mathbf{0.00} \pm 0.00\%$	$\mathbf{0.11} \pm 0.03\%$	$\mathbf{0.11} \pm 0.02\%$	$\mathbf{0.10} \pm 0.02\%$

References

[1] Tianping Chen and Hong Chen. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Transactions on Neural Networks*, 6(4):911–917, 1995.

[2] Bernardo Llanas, Sagrario Lantarón, and Francisco J Sáinz. Constructive approximation of discontinuous functions by neural networks. *Neural Processing Letters*, 27(3):209–226, 2008.

[3] Lu Lu, Xuhui Meng, Zhiping Mao, and George Em Karniadakis. Deepxde: A deep learning library for solving differential equations. *SIAM Review*, 63(1):208–228, 2021.

[4] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[5] Hailong Sheng and Chao Yang. Pfnn: A penalty-free neural network method for solving a class of second-order boundary-value problems on complex geometries. *Journal of Computational Physics*, 428:110085, 2021.

[6] Hailong Sheng and Chao Yang. Pfnn-2: A domain decomposed penalty-free neural network method for solving partial differential equations. *arXiv preprint arXiv:2205.00593*, 2022.

[7] Sifan Wang, Yujun Teng, and Paris Perdikaris. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing*, 43(5):A3055–A3081, 2021.