Project proposal for using transformer ICL to solve parametric PDE and inverse problems

Janis Aiad and Xiayimei Han

May 2026

1 Problem statement

In this project, we want to generalize the results in [1]. More specifically, we study the linear elliptic PDEs that are second-order, strongly elliptic on a bounded Lipschitz domain $\Omega \subset \mathbb{R}^{d_0}$:

$$\begin{cases} \mathcal{L}_{a,V} u(x) := -\nabla \cdot (a(x)\nabla u(x)) + V(x)u(x) = f(x) & x \in \Omega, \\ u(x) = 0 & x \in \partial\Omega. \end{cases} \tag{1}$$

where $a \in L^\infty(\Omega)$ is strictly positive, $V \in L^\infty(\Omega)$ is non-negative and $f \in \chi_f \subset L^2(\Omega)$. By standard well-posedness of the elliptic PDE, the solution $u \in \chi_u \subset H_0^1(\Omega)$.

At the training stage, we are given a training dataset comprising N length-n prompts of source-solution pairs $\{(f_i^j, u_i^j)_{i=1}^n\}_{j=1}^N$, where $f_i^j \overset{\text{i.i.d.}}{\sim} P_f$ and parameters $(a_j, V_j) \overset{\text{i.i.d.}}{\sim} P_a \times P_V$. After pre-training, the ICL model is asked to predict the solution for a new source term $f_\star$ conditioned on a new prompt $\{(f_i, u_i)_{i=1}^m\}$ from a new task. The prompt length m may differ from n.

A practical ICL model can only operate on finite-dimensional data. Following [1], we use a Galerkin discretization. Let $\{\phi_k\}_{k=1}^\infty$ be a basis and let $\Phi(x) = (\phi_1(x), \dots, \phi_d(x))$. We write

$$u(x) \approx \langle \mathbf{u}, \Phi(x) \rangle, \qquad f(x) \approx \langle \mathbf{f}, \Phi(x) \rangle.$$

The coefficients solve

$$A\mathbf{u} = \mathbf{f}, \qquad A_{ij} = \langle \phi_j, \mathcal{L}_{a,V}\phi_i \rangle, \qquad \mathbf{f}_i = \langle f, \phi_i \rangle. \tag{2}$$

The forward Galerkin task is therefore to apply A^{-1} to $\mathbf{f}$.

Our additional question is inverse or semi-inverse. We assume that the unknown objects live in low-dimensional coefficient spaces. There are three related cases.

First, in a forward/source-coefficient case, the unknown input source is written

$$f_z = f_0 + \sum_{k=1}^{K_f} z_k^f \varphi_k.$$

The prompt is used to estimate the source coefficients z^f, after which one solves the Galerkin forward problem.

Second, in an inverse/operator case, the coefficient or the Galerkin matrix is written in a low-dimensional family

$$A(z^A) = A_0 + \sum_{k=1}^{K_A} z_k^A A_k.$$

The prompt is used to estimate the operator coefficients z^A, after which one reconstructs $A(z^A)$.

Third, in the most general formulation, we do not assume that a stable inverse A^{-1} is directly available. Instead, both the source and the operator are estimated from the prompt, and the final prediction is obtained by solving the finite-dimensional least-squares problem

$$\widehat{\mathbf{u}}_\star = \arg\min_{\mathbf{u}} \left\| A(\widehat{z}^A)\mathbf{u} - \mathbf{f}_{\widehat{z}^f,\star} \right\|^2 + \gamma \|\mathbf{u}\|^2.$$

This covers non-invertible, ill-conditioned, or overdetermined discretizations. It also explains why the model must be parametric in the prompt: the transformer weights are fixed at test time, so adaptation to a new coefficient a or a new source family must occur through the inferred coefficients z^A, z^f and the recurrent least-squares computation, not by changing the weights.

2 Proposed method

The common finite-dimensional problem is a least-squares inference problem for a task-dependent coefficient vector. Depending on the case, this vector is either z^f, z^A, or the joint vector

$$z = (z^f, z^A).$$

We write the abstract system as

$$G_m z \simeq b_m, \qquad z_\star = (G_m^\top G_m + \lambda I)^{-1} G_m^\top b_m. \tag{3}$$

What is λ? Here the rows $g_i^\top$ of G_m are weak-form equations extracted from the prompt. In the operator-inverse case, for instance,

$$(G_m)_{(i,r),k} = \int_\Omega \psi_k \nabla u_i \cdot \nabla v_r \, dx,$$

What do you mean by $(G_m)_{(i,r),k}$? It seems confusing to me that a matrix has 3 subscript coordinates. where ψ_k is a coefficient basis and v_r is a test function.And u_i is...? In the source-coefficient case, G_m is the analogous matrix measuring how the source basis functions φ_k are observed through the prompt. What do you mean by observed through? In the joint case, the same recurrent least-squares mechanism estimates both source and operator coefficients and then solves the final Galerkin least-squares system for $\mathbf{u}_\star$.

The transformer block is viewed as a recurrent solver for (3). If

$$r_i^\ell = b_i - g_i^\top z^\ell,$$

then the ideal preconditioned Richardson step is

$$z^{\ell+1} = z^\ell + B_\Theta \left[G_m^\top (b_m - G_m z^\ell) - \lambda z^\ell \right]. \tag{4}$$

The feed-forward network of the transformer represents the learned preconditioner B_Θ.

The role of attention is to choose or weight equations that are informative for spectral directions of the coefficient space. A head direction $p_h \in \mathbb{R}^K$ is a direction in the basis of z. If z denotes source coefficients, then

$$\varphi_{p_h} = \sum_k (p_h)_k \varphi_k.$$

If z denotes operator coefficients, then

$$A_{p_h} = \sum_k (p_h)_k A_k.$$

The useful choice is spectral: $p_h \simeq u_h$, where u_h is an eigenvector, or an approximate invariant direction, of $G_m^\top G_m$. Why $G_m^\top G_m$? With keys $k_i \simeq g_i$ and queries $q_h \simeq p_h$, the score

$$q_h^\top k_i \simeq p_h^\top g_i$$

measures how much weak equation i probes the spectral direction p_h.

In the simplest signed softmax version, one uses two queries p_h and $-p_h$, scalar affine values $v_i^\ell = r_i^\ell$, and a feed-forward block to combine the positive and negative attention outputs. The resulting recurrence is a softmax-attention implementation of the Richardson update:

$$z^{\ell+1} = z^\ell + \underbrace{\mathcal{P}_\Theta}_{\text{FFN preconditioner}} \left[\underbrace{\sum_{h=1}^{H} W_h^O \sum_i \underbrace{\operatorname{softmax}_i(q_h^\top k_i)}_{\text{Q/K routing}} \underbrace{v_i^\ell}_{\text{value}} - \lambda z^\ell}_{\text{routed residual information}} \right]. \tag{5}$$

Thus Q/K choose informative equations, values carry residual information, and the feed-forward block implements the preconditioned update. Should the routed residual information include $-\lambda z^l$? Is H the number of heads?

Kernel viewpoint

The query-key scores can also be interpreted as a learned kernel between spectral directions and weak equations. In the linear case, $q_h^\top k_i \simeq p_h^\top g_i$. More generally, the model may learn feature maps

$$q_h = \varphi_Q(p_h), \qquad k_i = \varphi_K(g_i),$$

so that

$$q_h^\top k_i \simeq \kappa(p_h, g_i).$$

This is the point where nonlinear features enter the architecture. If the coefficient vector z is sparse or simple in the right feature space, the softmax kernel can route attention to the weak equations with large leverage for the relevant spectral modes. The recurrent layers then perform a Richardson/KRR correction in this feature-adapted coordinate system.

3 Division of work and methodology

In this project, Hancya is in charge of the encoder. More specifically, she will formulate how to encode elliptic PDEs using the same Galerkin encoder as in [1], and then extend it to weak-form inverse features (G_m, b_m). The forward/source version estimates coefficients of the source f_{z^f} in a chosen basis. The inverse/operator version estimates coefficients of $A(z^A)$, or of the PDE coefficient that induces $A(z^A)$. The joint version estimates both z^f and z^A and then solves the final least-squares Galerkin system for the predicted solution. This joint formulation is the most flexible one, because it does not require assuming that the discretized operator is exactly invertible.

The main objective of our project is to prove a better generalization error bound than the one in [1]. In greater detail, in our study, we want to prove an asymptotic decay of test error in terms of N, m and n that is faster than the result in [1], which is

$$\frac{1}{m} + \frac{1}{n^2} + \frac{1}{\sqrt{N}},$$

where N denotes the number of tasks covered in training, m is the length, i.e. the number of source-solution pairs in each prompt.

In addition, Hancya will study the proof of in-distribution generalization error for ICL in Appendix B of [1] and adapt it to the least-squares system (3). The transformer part of the project studies whether a recurrent transformer with shared W_Q, W_K, W_V and shared feed-forward weights can approximate (4). The target is a bound of the form

$$\mathbb{E}\|\widehat{u}_\star - u_\star\|^2 \lesssim \varepsilon_{\mathrm{Gal}}(d) + C\left(\frac{K_{\mathrm{eff}}}{m} + \rho^{2L} + \frac{\mathrm{Comp}(\Theta)}{\sqrt{N}} + \varepsilon_{\mathrm{arch}}\right),$$

where L is the number of recurrent layers, ρ^{2L} is the Richardson solver error, and K_{eff} is the effective dimension of the source, operator, or joint source–operator family. Why does ρ have superscript $2L$? Also, do you want to explicitly write down the tasks that you wish to tackle?

A Weak-form derivation of the inverse least-squares system

We describe the operator-inverse case. Suppose

$$a_z(x) = a_0(x) + \sum_{k=1}^{K} z_k \psi_k(x).$$

For each prompt pair (f_i, u_i) and test function v_r, the weak form gives

$$\int_\Omega a_z \nabla u_i \cdot \nabla v_r \, dx = \int_\Omega f_i v_r \, dx.$$

Substituting the expansion of a_z gives

$$\sum_{k=1}^{K} z_k \int_\Omega \psi_k \nabla u_i \cdot \nabla v_r \, dx = \int_\Omega f_i v_r \, dx - \int_\Omega a_0 \nabla u_i \cdot \nabla v_r \, dx.$$

Thus

$$G_{(i,r),k} = \int_\Omega \psi_k \nabla u_i \cdot \nabla v_r \, dx, \qquad b_{(i,r)} = \int_\Omega f_i v_r \, dx - \int_\Omega a_0 \nabla u_i \cdot \nabla v_r \, dx.$$

Stacking the equations gives

$$G_m z^A = b_m.$$

The source-coefficient case is analogous, except that the unknown vector is z^f in

$$f_{z^f} = f_0 + \sum_k z_k^f \varphi_k.$$

The joint case combines both sets of coefficients. One first estimates

$$z = (z^f, z^A)$$

from the prompt by a least-squares system of the form (3). Then the prediction for a new source is obtained from the finite-dimensional least-squares solve

$$\widehat{\mathbf{u}}_\star = \arg\min_{\mathbf{u}} \left\| A(\widehat{z}^A)\mathbf{u} - \mathbf{f}_{\widehat{z}^f,\star} \right\|^2 + \gamma \|\mathbf{u}\|^2.$$

This joint formulation includes the forward/source case, the inverse/operator case, and the case where the final Galerkin system is rectangular, singular, or ill-conditioned.

B Richardson iteration and preconditioning

Let

$$H_m = G_m^\top G_m + \lambda I, \qquad c_m = G_m^\top b_m.$$

The ridge solution solves

$$H_m z_\star = c_m.$$

A Richardson iteration is

$$z^{\ell+1} = z^\ell + B(c_m - H_m z^\ell).$$

For $B = \eta I$, convergence requires

$$\rho(I - \eta H_m) < 1.$$

If B approximates H_m^{-1}, the iteration is preconditioned. In our transformer interpretation, this preconditioning is performed by the feed-forward network after attention.

C Attention implementation

Let

$$r_i^\ell = b_i - g_i^\top z^\ell.$$

For a head direction p_h, use

$$q_{h,+} = p_h, \qquad q_{h,-} = -p_h, \qquad k_i = g_i.$$

The scores are

$$s_{h,+,i} = p_h^\top g_i, \qquad s_{h,-,i} = -p_h^\top g_i.$$

The softmax weights are

$$a^\ell_{h,+,i} = \frac{\exp(s_{h,+,i}/\tau)}{\sum_j \exp(s_{h,+,j}/\tau)}, \qquad a^\ell_{h,-,i} = \frac{\exp(s_{h,-,i}/\tau)}{\sum_j \exp(s_{h,-,j}/\tau)}.$$

With scalar values $v_i^\ell = r_i^\ell$, the messages are

$$m^\ell_{h,+} = \sum_i a^\ell_{h,+,i} r_i^\ell, \qquad m^\ell_{h,-} = \sum_i a^\ell_{h,-,i} r_i^\ell.$$

The feed-forward block combines these messages to approximate the signed directional moment

$$p_h^\top G_m^\top r^\ell = \sum_i (p_h^\top g_i) r_i^\ell.$$

The final update is

$$z^{\ell+1} = z^\ell + F_\Theta(z^\ell, \{m^\ell_{h,+}, m^\ell_{h,-}\}_{h=1}^H).$$

The target is

$$F_\Theta \approx B_\Theta \left[G_m^\top (b_m - G_m z^\ell) - \lambda z^\ell \right].$$

Do we have any guarantee on the target?

D Experimental checks

The experiments so far test the abstract least-squares system

$$b_i = g_i^\top z + \xi_i$$

before using a full PDE encoder. They are meant to verify the algorithmic mechanism separately from Galerkin discretization errors.

First, a constructive recurrent Richardson solver converges geometrically to the ridge solution. In the clean setting $K = 16$, $m = 128$, and full attention capacity, the error to the ridge posterior mean decreases from approximately

$$1.18 \times 10^{-1}$$

at depth $L = 1$, to

$$2.78 \times 10^{-5}$$

at $L = 8$, to about

$$10^{-8}$$

at $L = 16$, and reaches numerical precision around $L = 32$. This confirms the interpretation of depth as the number of Richardson iterations.

Second, linear attention computes the signed residual moment exactly when the attention capacity covers the coefficient space. In the constructive model, a head with projection P_h computes

$$P_h^\top P_h G_m^\top (b_m - G_m z^\ell).$$

Hence, when

$$\sum_h P_h^\top P_h = I_K,$$

the linear attention block recovers the full moment $G_m^\top (b_m - G_m z^\ell)$. Experimentally, this corresponds to the threshold

$$H d_h \geq K.$$

Below this threshold, linear attention only updates a subspace and fails to recover all coefficients.

Third, softmax attention with scalar values does not directly compute the signed Richardson moment. This is expected because softmax gives positive normalized weights, whereas $G_m^\top r^\ell$ is a signed, unnormalized sum.

Fourth, softmax with full vector values works even below the query-key rank threshold, because the value stream already carries all coefficient directions. In this case the value contains the signed weak evidence $g_i r_i^\ell$. For example, in a regime with $K = 8$ and $Hd_h = 4 < K$, the full-value softmax version reaches an error around 10^{-5}, while the corresponding low-rank linear attention remains around 10^{-1} to 1.

Fifth, when the values are projected to the same subspace as the query/key heads, this below-capacity advantage disappears. In the same $K = 8$, $Hd_h = 4$ regime, projected values give an error around 0.56, essentially matching the low-rank linear attention. This confirms that the improvement came from the value channel, not from a hidden ability of low-rank Q/K to reconstruct all directions.

Finally, preconditioning is essential for ill-conditioned systems. With scalar Richardson, the contraction factor approaches one when the condition number of $G_m^\top G_m$ is large. Jacobi preconditioning substantially improves stability. In the synthetic tests, softmax attention with full vector values and Jacobi preconditioning remains accurate even at high condition number, while projected values and low-rank linear attention fail below the capacity threshold.

These experiments support the proposed decomposition:

Q/K route weak equations, V carries residual information, the FFN learns the preconditioned Richardson update.

References

[1] Frank Cole, Yulong Lu, Riley O'Neill, and Tianhao Zhang. Provable in-context learning of linear systems and linear elliptic PDEs with transformers. *arXiv preprint arXiv:2409.12293*, 2024.

[2] G. Kang et al. Transformers can learn posterior predictive distributions in-context. *arXiv preprint arXiv:2605.26713*, 2026.

[3] Matthew Smart, Soumya Ganguly, Nilava Metya, Alexandre V. Morozov, and Anirvan M. Sengupta. Attention as in-context empirical Bayes: A two-stage view via particle dynamics. *arXiv preprint arXiv:2605.29351*, 2026.