Emergence, Retention and Mitigation of Ill-conditioning due to Basis Lifting in KANs

Mazlum Ferhat Arslan[1,2,*] **Weihong Guo**[1] **Shuo Li**[2,3]
[1]Dept. of Mathematics, Applied Mathematics and Statistics
[2]Dept. of Biomedical Engineering
[3]Dept. of Computer and Data Sciences
Case Western Reserve University
mferhata@gmail.com wx49@case.edu shuo.li11@case.edu
[*]Corresponding author

Abstract

Kolmogorov–Arnold Networks (KANs) expand each input coordinate through a fixed basis family, yet the optimization role of this basis lifting remains poorly understood. We identify basis lifting as a structural source of ill-conditioning in KANs and provide a mechanism-level analysis linking lifted feature geometry to convergence, accuracy, and seed sensitivity. Our analysis studies the evolution of lifted empirical Gram matrices under stochastic gradient descent (SGD) and shows that fixed basis lifting induces and preserves near-rank deficiency through eigenvalue and determinant drift. It further concentrates feature energy into a smaller number of directions, amplifying dominant eigenvalues and worsening condition numbers. These results show that basis selection is not merely an expressivity choice, but also an optimization and conditioning choice. Motivated by this mechanism, we introduce T-KAN, a learnable basis-coordinate transformation that aligns basis directions before the learned linear map. It reshapes lifted Gram geometries while preserving the expressivity and introducing negligible parameter overhead. In controlled small-to-moderate KAN regimes, T-KAN often improves conditioning and optimization with B-spline, Chebyshev, and RBF KANs.

1 Introduction

Kolmogorov Arnold Networks (KANs) expand each coordinate through a fixed basis family, and this basis lifting can make training slower, less accurate, and more seed-sensitive by ill-conditioning lifted feature geometry. This geometry directly shapes optimization since every learned linear map acts after it. Figure 1 shows this connection through lifted Gramian spectra and training curves. Standard KANs exhibit poorly spread spectra and large lifted Gramian condition numbers, attaining slower convergence, lower accuracy, and higher seed variability. The central obstruction is ill-conditioned geometry created due to the fixed basis coordinates.

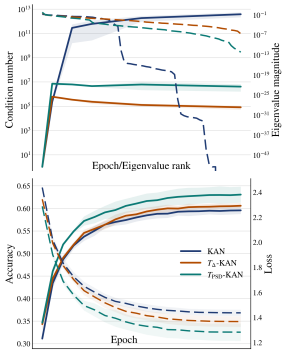


Figure 1: Ill-conditioning of KANs (top) slows convergence and task performance (bottom); T-KAN mitigates them. Details in Sec. 6.1.

If the lifted basis coordinates were already aligned with task-relevant directions, basis lifting would not introduce a geometric bottleneck. The lifted representation would match the task, and weak directions would reflect task geometry, not basis mismatch. Yet the right basis for a task is not known a priori. As empirical work demonstrates, different bases favor different regimes. B-splines suit symbolic regression [Liu et al., 2024]; polynomials suit Helmholtz PDEs [Shukla et al.,

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

2024]; RBFs suit convection-diffusion [Li, 2024]. As training discovers task directions, the fixed orientation for the basis can become misaligned leading to worsened conditioning.

In this work, we identify sufficient stochastic-geometric conditions under which basis lifting creates and retains near-rank deficiency, and we empirically show that these signatures appear in trained KANs. Our analysis shows that basis lifting changes the stochastic geometry of training, causing weak lifted directions to collapse and feature energy to concentrate. This view isolates two sources of spectral imbalance: *i)* basis derivatives shape stochastic motion along weak lifted directions; *ii)* basis curvature bends lifted features toward compressed configurations. Together, they create and retain near-rank deficiency, which leads to amplified ill-conditioning due to basis lifting. This gives a concrete mechanism behind the spectra and training behavior in Fig. 1.

Motivated by our analysis, we introduce T-KAN, which inserts a small learnable transformation across the basis channels of each KAN layer before the learned linear map. Rather than changing the scalar basis functions or increasing the lifted span, this transformation reparameterizes the same lifted features in a coordinate system better aligned with the empirical training geometry. For invertible transformations, the represented function class is preserved, while the lifted Gramian is reshaped through an induced basis-side preconditioner. Thus, T-KAN targets the optimization geometry created by basis lifting directly by learning how the basis functions should be coordinated for the task.

Contributions. This paper provides a novel mechanism-level account of KAN ill-conditioning and proposes a principled preconditioning method. Our main contributions include:

1. **Novel ill-conditioning mechanisms for KANs.** We identify previously uncharacterized mechanisms arising from KAN's fixed basis lifting: derivative-noise amplification and curvature compression. We give sufficient conditions under which these mechanisms create and retain near-rank deficiency in lifted empirical Gramians, then show that it leads to amplified ill-conditioning.
2. **Principled learned basis preconditioning.** We introduce T-KAN, a learned basis transformation for KANs with negligible parameter overhead. Unlike changing the basis family or adding a wider lifted representation, T-KAN preserves the lifted feature span, but changes the Gram geometry seen by optimization. It is learned jointly with the model, targets the empirical lifted geometry and applies across basis families.
3. **Cross-basis empirical validation.** We demonstrate that the assumed conditions can be met with simple datasets producing ill-conditioned KAN behavior and verify that T-KAN improves B-spline, Chebyshev, and RBF KANs. Across five image classification datasets, it improves lifted spectra and condition numbers; yielding higher accuracy and faster convergence across tasks.

2 Related work

Ill-conditioning of Gram matrices is a training stability bottleneck. When feature or Jacobian Gram matrices have spread-out spectra, updates become primarily governed by the worst-conditioned directions [Du et al., 2019, Nguyen et al., 2021] and can become too aggressive in sharp directions [Xiao et al., 2020, Liu et al.]. This appears through convergence rates and admissible step sizes controlled by Gramian eigenvalues [Du et al., 2019, Nguyen et al., 2021, Zhao et al., 2024]. It is also a dominant concern at initialization, where distorted spectra make training sensitive to depth, architecture and hyperparameters [Pennington et al., 2017, Xiao et al., 2018, Schoenholz et al., 2016].

KAN training instability is empirically established. KAN–MLP comparisons [Yu et al., 2024, Shukla et al., 2024, Sohail, 2024] report convergence and test performance vary across basis orders, initialization, optimizers, and learning rates. Kiyani et al. [2025] show large method-dependent gaps; initialization studies show dependence on parameter scaling [Rigas et al., 2025]; and [Rigas et al., 2024, Zheng et al., 2025] report grid-extension loss spikes and oscillatory activations.

From basis-specific effects to basis lifting analysis. Prior KAN work studies basis regularization [Arslan et al., 2025], grid refinement [Southworth et al., 2026], solver choices [Actor et al., 2025, Kiyani et al., 2025, Sohail, 2024, Dudek and Rodak, 2025, Poluektov and Polar, 2023], and model compression [Arslan et al., 2026]. While fixed spline analyses tie geometry to prescribed coordinate maps or knot refinement [Southworth et al., 2026] they do not explain why lifted representations become unstable during SGD. Our analysis identifies fixed basis misalignment as a concrete optimization bottleneck arising from basis lifting in KANs.

Fixed spline-coordinate changes vs. learned basis-coordinate preconditioning. Southworth et al. [2026] establish an analytic linear change of basis between spline KAN layers and multichannel power-ReLU MLP layers, and show that gradient descent in one coordinate system corresponds to preconditioned gradient descent in the other. Their basis change matrix is fixed by the spline knots and order, and their multilevel method uses fixed interpolation operators between nested spline grids. In comparison, our work does not rely on a prescribed transform or splines in general. Instead, it learns a layerwise basis-coordinate transform T_ℓ jointly with the model, targeting the empirical lifted-feature Gramian and applying the same mechanism to B-spline, Chebyshev, and RBF bases.

3 Preliminaries

Basis lifting in KAN layers. Let $B \geq 1$, and let $\psi_1, \ldots, \psi_B : \mathbb{R} \to \mathbb{R}$ be fixed univariate basis functions. For layer ℓ, let d_ℓ and $d_{\ell+1}$ denote the input and output dimensions, and let $u = (u_1, \ldots, u_{d_\ell}) \in \mathbb{R}^{d_\ell}$. The lifted feature matrix is

$$\Psi_\ell(u) := \big(\psi_b(u_j)\big)_{1\leq j\leq d_\ell,\ 1\leq b\leq B} \in \mathbb{R}^{d_\ell \times B}. \tag{1}$$

We call $\mathrm{vec}(\Psi_\ell(u))$ the **basis-lifted representation**. A KAN layer is linear in this lifted vector:

$$\Phi_\ell(u) = W_\ell\ \mathrm{vec}\big(\Psi_\ell(u)\big), \qquad W_\ell \in \mathbb{R}^{d_{\ell+1}\times d_\ell B} \quad \text{(learnable parameters)}. \tag{2}$$

Lifted Gramian, rank deficiency and conditioning. For layer-ℓ, the **empirical lifted Gramian** is given by

$$G_\ell := \frac{1}{N}\sum_{n=1}^{N} \mathrm{vec}\big(\Psi_\ell(u_\ell^{(n)})\big)\ \mathrm{vec}\big(\Psi_\ell(u_\ell^{(n)})\big)^\top \in \mathbb{R}^{d_\ell B\times d_\ell B}, \tag{3}$$

Table 1: Notation table

B	basis size
d_ℓ	layer-ℓ input dim.
$\Psi_\ell(u)$	lifted feature matrix
$\Phi_\ell(u)$	KAN layer map
$u_\ell^{(n)}$	nth sample at layer ℓ
G_ℓ	empirical feature Gram
$G_{\ell,ab}$	(a,b) block of G_ℓ

where $u_\ell^{(n)} \in \mathbb{R}^{d_\ell}$, $n = 1, \ldots, N$. Under row-major vectorization, we write $G_\ell = [G_{\ell,ab}]_{a,b=1}^{d_\ell}$ with $G_{\ell,ab} \in \mathbb{R}^{B\times B}$, so the B channels of each coordinate are contiguous. Since G_ℓ is a Gramian, it is positive semidefinite. Writing its eigenvalues as $\lambda_{max} := \lambda_1(G_\ell) \geq \cdots \geq \lambda_{min} := \lambda_{d_\ell B}(G_\ell) \geq 0$, we say the lifted representation is rank deficient if $\mathrm{rank}(G_\ell) < m$, equivalently $\lambda_m(G_\ell) = 0$. It is near rank deficient if its smallest eigenvalues are nonzero but negligible, i.e. $\lambda_{min} \leq \varepsilon\,\lambda_{max}$. When G_ℓ is positive definite, its conditioning is measured by $\kappa(G_\ell) = \lambda_{max}/\lambda_{min}$; large values indicate that the matrix is ill-conditioned. Exact rank deficiency gives $\kappa(G_\ell) = \infty$, while near rank deficiency corresponds to very large but finite $\kappa(G_\ell)$. Since G_ℓ is formed from N lifted sample vectors in $\mathbb{R}^{d_\ell B}$, $\mathrm{rank}(G_\ell) \leq \min\{N, d_\ell B\}$. Throughout, we assume $N \geq d_\ell B$ to avoid sample size induced rank deficiency.

Following block Gauss–Newton view [Botev et al., 2017], the corresponding block curvature is

$$\mathcal{H}_{W_\ell} = \frac{1}{N}\sum_{n=1}^{N} \mathcal{C}_\ell^{(n)} \otimes \mathrm{vec}\big(\Psi_\ell(u_\ell^{(n)})\big)\,\mathrm{vec}\big(\Psi_\ell(u_\ell^{(n)})\big)^\top \overset{\text{K-FAC}}{\approx} \frac{1}{N}\sum_{n=1}^{N} \mathcal{C}_\ell^{(n)} \otimes G_\ell, \tag{4}$$

where $\mathcal{C}_\ell^{(n)}$ is the output-curvature matrix for sample n, and the last approximation holds in the isotropic-output case, or under K-FAC [Martens and Grosse, 2015] factorization. Isotropic output curvature is proportional to the identity. K-FAC factorizes the averaged output- and feature-side terms. Thus, G_ℓ determines the conditioning contributed by the lifted representation. Consequently, small eigenvalues of G_ℓ create weak layer-wise update directions even when the output-curvature factor is well conditioned.

4 Basis Lifting Amplifies Ill-conditioning in KANs

This section formalizes sufficient conditions under which nonlinear basis lifting can amplify ill-conditioning by shrinking residual basis directions outside the pre-lift feature span. At layer ℓ, $u \in \mathbb{R}^{d_\ell}$ is one input vector and $u_\ell^{(n)}$ is the input for sample n. The rows of $U_t \in \mathbb{R}^{N \times d_\ell}$ are these sample inputs at training time t. We fix the layer and write $d = d_\ell$. The identity basis gives the block U_t; adding $q = \psi_2$ gives $H_t = q(U_t)$, entrywise. The residual covariance S_t is the Gram matrix of the component of H_t outside $\mathrm{col}(U_t)$:

$$H_{\perp,t} = (I - P_{U_t})H_t, \qquad S_t = N^{-1} H_{\perp,t}^\top H_{\perp,t}, \tag{5}$$

where $P_U = UU^\dagger$ is the orthogonal projector onto $\mathrm{col}(U)$.

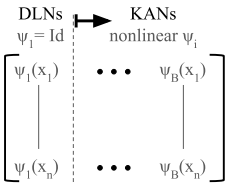


Figure 2: Our analysis extend results from deep linear networks to KANs by introducing nonlinear ψ_j, $j > 1$.

The analysis assumes that U_t remains in a fixed-rank neighborhood and $S_t \succ 0$. Let ζ be the first time either condition fails. For $t < \zeta$, P_{U_t}, $U_t^\dagger$, and $\log\det S_t$ are well-defined and smooth, so Itô's formula applies.

The linear-network result of Varre et al. is the starting point of our analysis when no extra basis function is present.

Theorem 1 (Linear networks; Varre et al. [2024])**.** *For two-layer linear networks trained with square loss, label-noise SGD induces rank deficiency through determinant and eigenvalue dynamics. This corresponds to $B = 1$ with $\psi_1(x) = x$, for which the residual block in Eq.* (5) *is absent.*

The rank of S_t equals the additional independent rank contributed by the added basis block. A zero mode means that an added direction is redundant, while a small mode means that it is weak or nearly dependent.

In the following, Theorems 2 and 4 establish emergence: under the stated stochastic-dominance condition, residual basis volume contracts and can cross a near-degeneracy threshold in finite time. Theorem 6 establishes retention: once residual volume is small, the same negative-drift dynamics makes it difficult to escape from that region.

4.1 Emergence of Near Rank Deficiency due to Basis Misalignment

Write $S_t = S(U_t)$, where $S(U)$ is the residual covariance in (5) evaluated at a feature matrix U. Let $f(U) = \log\det S(U)$ and $A_S(U) = \nabla_U f(U)$. For a direction of feature change V, $Df(U)[V] = \langle A_S(U), V\rangle_F$; thus A_S measures how residual volume responds to changes in the features. Suppose that locally the feature process has Itô form

$$dU_t = a_t\,dt + \sum_{m=1}^{p} V_{m,t}\,dB_t^m, \tag{6}$$

where a_t is mean feature motion from task optimization and $V_{m,t}$ are stochastic minibatch-noise directions. For this process define the noise compression and the residual curvature term by

$$\mathcal{K}_S(t) = \frac{1}{2}\sum_m \mathrm{tr}\big(S_t^{-1} DS_t[V_{m,t}] S_t^{-1} DS_t[V_{m,t}]\big)\,, \tag{7}$$

$$\mathcal{E}_S(t) = \frac{1}{2}\sum_m \mathrm{tr}\big(S_t^{-1} D^2 S_t[V_{m,t}, V_{m,t}]\big)\,. \tag{8}$$

$\mathcal{K}_S$ is nonnegative and is the log-determinant compression produced by stochastic motion. $\mathcal{E}_S$ has mixed sign and contains basis curvature, projection curvature, and other second-order effects.

Assumption 1 (Stochastic compression dominates residual-volume expansion)**.** *For some fixed $\rho > 0$ and all $t < \zeta$,*

$$\mathcal{K}_S(t) \geq \langle A_S(U_t), a_t\rangle_{F,+} + (\mathcal{E}_S(t))_+ + \rho \tag{9}$$

Assumption 1 requires stochastic compression to exceed the total expansion from mean feature motion and second-order geometry by a margin ρ. The two positive-part terms retain only contributions that increase residual volume. Theorem 2 derives the drift of $\log\det S_t$ and shows that it is negative under this assumption.

Theorem 2 (Residual collapse from dynamic misalignment)**.** *Assume the feature dynamics* (6) *for* $0 \le t < \zeta$*, where* U_t *has fixed rank and* $S_t \succ 0$*. Under Assumption 1, there is a continuous local martingale* M_t *such that*

$$d \log\det S_t \le -\rho\, dt + dM_t. \tag{10}$$

For the single added basis function, the residual determinant and the smallest residual mode have the explicit velocity formulas in Proposition 3. In this calculation, U, H, S denote U_t, H_t, S_t at a fixed time.

Proposition 3 (Near rank deficiency for one added basis function)**.** *Let* $H = q(U)$*,* $H_\perp = (I - P_U)H$*, and* $S = N^{-1}H_\perp^\top H_\perp$*, with* $S \succ 0$*. The determinant drift is linear in the feature velocity:* $d(\log\det S)/dt = \langle A_{\det}, \dot U\rangle_F$*, where the determinant sensitivity is*

$$A_{\det} = \frac{2}{N}\Big[q'(U) \odot (H_\perp S^{-1}) - H_\perp S^{-1}(U^\dagger H)^\top\Big]. \tag{11}$$

If b *is a unit eigenvector for a simple smallest eigenvalue of* S*, then* $d\lambda_{\min}(S)/dt = \langle A_b, \dot U\rangle_F$*, where the weakest mode sensitivity is*

$$A_b = \frac{2}{N}\Big[q'(U) \odot \big((H_\perp b)b^\top\big) - (H_\perp b)(U^\dagger H b)^\top\Big]. \tag{12}$$

Now suppose $\dot U = M + \Xi$*, where* M *is the conditional mean velocity and* Ξ *is conditionally mean-zero with* $\langle A, \Xi\rangle_F$ *sub-Gaussian with proxy* $\sigma^2\|A\|_F^2$ *for every fixed matrix* A*. If* $\langle A_{\det}, M\rangle_F \le -\gamma$*, then*

$$\mathbb{P}\left(\frac{d}{dt}\log\det S \le -\frac{\gamma}{2}\right) \ge 1 - \exp\left(-\frac{\gamma^2}{8\sigma^2\|A_{\det}\|_F^2}\right). \tag{13}$$

If $\langle A_b, M\rangle_F \le -\gamma$*, the same bound holds with* $\frac{d\lambda_{\min}(S)}{dt}$ *and* A_b *in place of* $d(\log\det S)/dt$ *and* $A_{\det}$*.*

Proposition 3 states that once the mean velocity points in a shrinking direction, random velocity fluctuations preserve the shrinkage with high probability. The basis enters explicitly through q', while the strength of the effect is amplified by the conditioning of S.

Choose $0 < \delta < \det S_0$ as a small residual-volume threshold. Theorem 4 bounds the probability that $\det S_t$ reaches δ by time T, before ζ. Since S_t has r_S eigenvalues, $\det S_t \le \delta$ implies that at least one is no larger than δ^{1/r_S}. Setting $\delta = \alpha^{r_S}$ therefore gives $\lambda_{\min}(S_t) \le \alpha$.

Theorem 4 (Emergence of residual rank deficiency)**.** *Let* r_S *be the number of columns of* $H_\perp$*, and let* $\tau_\delta = \inf\{t : \det S_t \le \delta\}$*. Suppose that, up to* $\tau_\delta \wedge \zeta$*, the process satisfies* $d\log\det S_t \le -\rho\, dt + dM_t$ *and* $d\langle M\rangle_t \le \Lambda\, dt$*, where* ζ *is the time at which the analysis conditions cease to hold. If* $\det S_0 > \delta$*, then for every* $T > 0$*,*

$$\mathbb{P}(\tau_\delta \le T \wedge \zeta) \ge 1 - \exp\left(-\frac{(\rho T - \log(\det S_0/\delta))_+^2}{2\Lambda T}\right) - \mathbb{P}(\zeta \le T). \tag{14}$$

On the event $\tau_\delta \le T \wedge \zeta$*,* $\lambda_{\min}(S_{\tau_\delta}) \le \delta^{1/r_S}$*. In particular, choosing* $\delta = \alpha^{r_S}$ *gives* $\lambda_{\min}(S_{\tau_\delta}) \le \alpha$*.*

Together, Theorems 2 and 4 establish that dynamic misalignment under noise dominance creates near rank deficiency for the residual basis, and Lemma 1 translates this into redundancy of the added basis inside $[U, H]$; its statement and proof are in Appendix C.

4.2 General Basis Mechanisms, Retention, and Ill Conditioning

The residual formulation extends from one added basis function to a full basis dictionary. For $B \ge 2$, define the residual dictionary block by

$$H_B = [\psi_2(U), \dots, \psi_B(U)], \qquad H_{B,\perp} = (I - P_U)H_B, \qquad S_B = N^{-1}H_{B,\perp}^\top H_{B,\perp}. \tag{15}$$

Along training, write $S_{B,t} = S_B(U_t)$. For this extension, ζ is the first exit from the fixed-rank neighborhood on which $S_{B,t} \succ 0$. The case $B = 2$ gives exactly the object S_t in (5). The case $B = 1$ has no residual dictionary block, so the residual theory is vacuous and the linear behavior is governed by Theorem 1.

The following result on stochastic motion due to basis derivatives is the direct generalization of Theorem 2. We use the same definitions as (7) and (8), with S replaced by S_B, and write $A_B^{\det} = \nabla_U \log\det S_B$ for the residual dictionary sensitivity.

Let $\mathcal{K}_B$ be the noise compression in (7) with $S \mapsto S_B$, and let $\mathcal{E}_B$ be the corresponding term in (8).

Assumption 1' (Dictionary stochastic-compression dominance)**.** *For some fixed $\rho_{\psi'} > 0$ and all $t < \zeta$,*

$$\mathcal{K}_B(t) \geq \langle A_B^{\det}(U_t), a_t\rangle_{F,+} + (\mathcal{E}_B(t))_+ + \rho_{\psi'} \tag{16}$$

Equation (16) applies Assumption 1 to the full residual dictionary S_B, replacing S by S_B, ρ by $\rho_{\psi'}$, $\mathcal{K}_S$ by $\mathcal{K}_B$, $\mathcal{E}_S$ by $\mathcal{E}_B$, and A_S by $A_B^{\det}$. For $B = 2$, it recovers the single-residual setting; for $B > 2$, it controls the residual volume of the entire non-identity dictionary.

Theorem 5 (Stochastic motion due to basis derivatives)**.** *Assume the local feature dynamics* (6)*, the residual dictionary* (15)*, and Assumption 1'. Then there is a continuous local martingale M_t such that $d\log\det S_{B,t} \leq -\rho_{\psi'}\,dt + dM_t$.*

For $B = 2$, this is exactly Theorem 2. For $B > 2$, it says that the same residual-volume mechanism acts on the entire non-identity dictionary. The role of the basis functions is now carried by the Jacobian of $[\psi_2, \ldots, \psi_B]$, rather than by the single derivative q'.

The curvature-margin refinement is given in Lemma 2 in Appendix C. The same negative drift also controls escape from a region of small residual volume.

Theorem 6 (Retention of residual rank deficiency)**.** *Let $B \geq 2$, let $F_t = \det S_{B,t}$, and suppose that $0 < F_0 \leq \alpha < \beta$. Let $\sigma_\beta = \inf\{t : F_t \geq \beta\}$. If, up to $\sigma_\beta \wedge \zeta$, $d\log F_t \leq -\rho\,dt + dM_t$ and $d\langle M\rangle_t \leq \Lambda\,dt$, then*

$$\mathbb{P}(\sigma_\beta \leq \zeta) \leq \exp\left(-\frac{2\rho\log(\beta/F_0)}{\Lambda}\right). \tag{17}$$

For $B = 2$, this recovers the single-residual setting S_t.

Near residual rank deficiency, the inverse covariance S^{-1} appears in both the determinant sensitivity and the negative log-determinant curvature term. Thus poor conditioning makes the system more sensitive to shrinking motions and can strengthen stochastic compression.

Proposition 7 (Ill conditioning amplifies residual drift)**.** *For the single added basis function, the determinant sensitivity in* (11) *satisfies*

$$\|A_{\det}\|_F \leq \frac{2\|H_\perp\|_F}{N\lambda_{\min}(S)}\left(\|q'\|_\infty + \|U^\dagger H\|_{\mathrm{op}}\right). \tag{18}$$

Moreover, if b is a unit eigenvector for $\lambda_{\min}(S)$, then the noise compression in (7) *obeys*

$$\mathcal{K}_S \geq \frac{1}{2}\sum_m \frac{\left(b^\top DS[V_m]b\right)^2}{\lambda_{\min}(S)^2}. \tag{19}$$

In particular, if some noise direction moves the weakest residual mode with $|b^\top DS[V_m]b| \geq c$, then $\mathcal{K}_S \geq c^2/(2\lambda_{\min}(S)^2)$.

Proposition 7 explains why emergence and ill conditioning reinforce one another. As $\lambda_{\min}(S)$ decreases, residual volume becomes more ill-conditioned, while ill conditioning increases the sensitivity of the log determinant and can enlarge the noise-compression term that drives further collapse.

5 T-KAN: Learned Basis Transformations via $W\mathrm{vec}(\Psi(u)\mathbf{T})$

This section introduces T-KAN, a direct intervention on the optimization geometry motivated by the analysis in Sections 4 which showed that basis lifting can cause ill-conditioned lifted feature Gramians and poor optimization geometry. Instead of replacing the fixed scalar basis functions $\{\psi_j\}_{j=1}^B$, we learn a transformation of the basis coordinates induced by these functions. Specifically, T-KAN is defined by the transformed lifted feature map

$$u \mapsto \mathrm{vec}(\Psi_T(u)) := \mathrm{vec}(\Psi(u)T), \qquad T \in \mathbb{R}^{B\times B} \quad \text{(learnable parameters)} \tag{20}$$

The role of $T_\ell \in \mathbb{R}^{B\times B}$ is to learn a basis-coordinate system in which gradients and conditioning is well behaved. Since $W\,\mathrm{vec}(\Psi_\ell(u)T_\ell) = W(T_\ell^\top \otimes I_{d_\ell})\,\mathrm{vec}(\Psi_\ell(u))$, T_ℓ can be absorbed into an effective coefficient matrix $\widetilde{W} := W(T_\ell^\top \otimes I_{d_\ell})$. Thus, its contribution is optimization geometry rather than expressivity: the transformed Gramian is $G_{\ell,T} = (T_\ell^\top \otimes I_{d_\ell})\,G_\ell\,(T_\ell \otimes I_{d_\ell})$. The KAN property we seek to retain is its explicit scalar edge-function representation. The learned transformation changes the training parameterization. After training, every T can be folded into the adjacent KAN coefficients, resulting in an ordinary KAN in the original basis span: every edge remains a scalar univariate function that can be plotted and inspected exactly as in the original KAN, and no additional transformation layer remains at inference.

Additionally, T introduces only B^2 parameters per layer, typically negligible compared with $|W| = mnB$.

Let $\mathcal{L}$ denote the empirical training loss. For *fixed* transformations, following the preconditioning view of Southworth et al. [2026], gradient descent in W induces a basis preconditioner $P = TT^\top$, expressed in the effective coordinates $\widetilde{W}$:

$$\widetilde{W}^+ = \widetilde{W} - \eta\nabla_{\widetilde{W}}\mathcal{L}(\widetilde{W})(TT^\top \otimes I_d). \tag{21}$$

Unlike the fixed preconditioning, learned T produces an *adaptive* preconditioner. Indeed, the preconditioner evolves in response to the loss geometry

$$P^+ = (T - \eta\nabla_T\mathcal{L})(T - \eta\nabla_T\mathcal{L})^\top = P - \eta\left(\nabla_T\mathcal{L}\,T^\top + T(\nabla_T\mathcal{L})^\top\right) + \mathrm{O}(\eta^2), \tag{22}$$

contributing the additional first-order loss decrease $-\eta\|\nabla_T\mathcal{L}\|_F^2$, which is absent when T is fixed. This can accelerate optimization by aligning the basis coordinates.

The following theorem shows that T-KAN can improve the condition number. The formal statement under basis-coordinate separability, together with the optimal improvement, is given in Appendix D.

Theorem 8 (Informal)**.** *Suppose the dominant ill-conditioning of the lifted feature Gramian separates into basis and coordinate factors, $G_\ell \approx C_\ell \otimes S_\ell$, where $C_\ell \in \mathbb{R}^{B\times B}$ captures basis-side correlations and $S_\ell \in \mathbb{R}^{d_\ell\times d_\ell}$ captures coordinate-side correlations. Then T-KAN can reduce the condition number from approximately $\kappa(C_\ell)\kappa(S_\ell)$ to approximately $\kappa(S_\ell)$.*

We propose two parameterizations of T. Additive parameterization T_Δ is an unconstrained transformation centered around the identity, $T_\Delta := I + \Delta$ with the induced preconditioner

$$P_\Delta = T_\Delta T_\Delta^\top = I + \Delta + \Delta^\top + \Delta\Delta^\top. \tag{23}$$

Symmetric positive definite T_{PSD} is the constrained transformation $T_{\mathrm{PSD}} := LL^\top + \epsilon I$[1]. Hence,

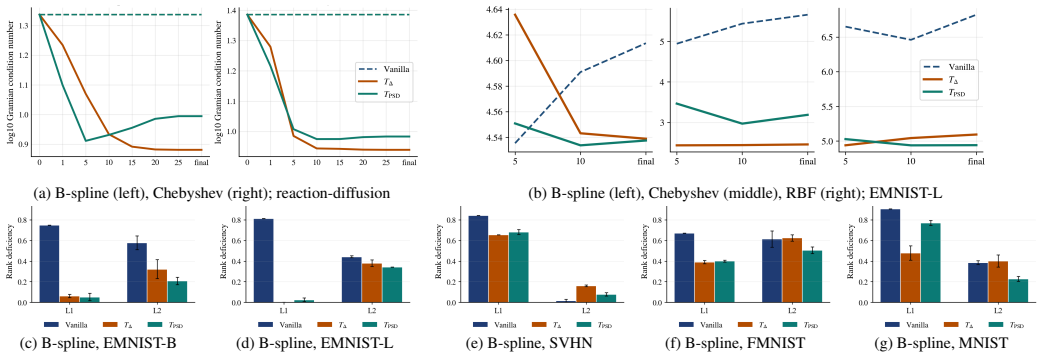
$$P_{\mathrm{PSD}} = T_{\mathrm{PSD}}T_{\mathrm{PSD}}^\top = T_{\mathrm{PSD}}^2. \tag{24}$$

T_{PSD} parameterizes a stable positive-definite transformation of the lifted basis. As a result, it can represent stretching, scaling along orthogonal directions. Unlike T_Δ, it cannot learn basis rotations.

Lastly, since W and T are learned jointly, there is a scale non-identifiability: scaling T can be offset by inversely scaling W. In practice, we control this by *i)* initializing the transformations near the identity and *ii)* regularizing them toward the identity via $\|T - I\|_F^2$.

6 Results

Experimental setup. Models are trained under the same setting, detailed in Appendix A. **Methods.** B-spline KANs, and optimizations targeting it, HKAN [Dudek and Rodak, 2025] and Multilevel-KAN [Southworth et al., 2026]; Chebyshev KANs [Sidharth et al., 2024]; and RBF KANs [Li, 2024] are used as baselines. **Datasets.** In order to isolate ill-conditioning of KANs, we evaluate our method on small scale datasets. We use EMNIST Balanced/Letters and SVHN as higher-complexity alternatives to MNIST and FMNIST. While EMNIST increases class cardinality and character-level ambiguity (e.g., `l` vs `1` vs `I`), SVHN introduces real-world image variability. This heterogeneity is expected to increase gradient diversity and the variance of minibatch gradient estimates. Thus, we treat these datasets as natural testbeds for higher effective SGD noise relative to MNIST.



(a) B-spline (left), Chebyshev (right); reaction-diffusion

(b) B-spline (left), Chebyshev (middle), RBF (right); EMNIST-L

(c) B-spline, EMNIST-B (d) B-spline, EMNIST-L (e) B-spline, SVHN (f) B-spline, FMNIST (g) B-spline, MNIST

Figure 3: **T-KANs mitigate lifted-feature Gramian ill-conditioning when vanilla KANs develop it.** Top: aggregate lifted-feature Gramian conditioning. In (b), the EMNIST-L curves use the post-warmup average over first two layers, where vanilla conditioning increases while both transformed variants remain lower. Bottom: near-rank deficiency of B-spline lifted Gramians at L_1 and L_2, computed as $1 - \#\{\lambda_i \geq \tau\lambda_{\max}\}/n$ with $\tau = 10^{-2}$ for L_1 and $\tau = 10^{-3}$ for L_2; bars show mean over seeds with standard deviation.

6.1 KANs can develop Gramian ill-conditioning; T-KAN mitigates it

In Fig. 1 we report the averaged Gramian conditioning on EMNIST-Balanced. The plotted thresholded condition number is $\kappa_{10^{-10}}(G_3^{\mathrm{read}}) = \frac{\lambda_1}{\max\{10^{-10}, \lambda_{n-m+1}\}}$, where $m = \lceil 0.3n \rceil$. That is, we compare the top eigenvalue to the largest eigenvalue in the bottom 30% of the spectrum, with a numerical floor. The lower panel overlays validation accuracy and loss for the same runs. This shows that the vanilla KAN develops a much more ill-conditioned geometry, while the transformed models keep this geometry better controlled and train faster.

Figure 3 isolates the conditioning failure mode that motivates the transformation layers. In the reaction–diffusion diagnostic, vanilla KANs maintain substantially larger lifted-feature Gramian condition numbers, whereas both T_Δ and T_{PSD} rapidly reduce the aggregate condition number. The same phenomenon appears in classification when we focus on the post-warmup feature-forming blocks especially the first two layers: on EMNIST-L, vanilla B-spline, Chebyshev, and RBF KANs develop worse conditioning, while the transformed models end with lower $\log_{10}\kappa$. The lower row shows the complementary rank-collapse view. Across EMNIST-B, EMNIST-L, and FMNIST, vanilla B-spline KANs generally produce more near-rank-deficient lifted Gramians, especially at the first layer, while transformed bases preserve more active directions. When the data has cleaner gradient directions, as in MNIST and FMNIST, learned rotations help less; which is visible in Figures 3(f,g).

Figure 4 shows that vanilla B-spline KANs produce faster-decaying lifted-feature Gramian spectra, indicating that feature energy concentrates into fewer directions, making the representation more nearly rank deficient. In contrast, T_Δ and T_{PSD} preserve larger trailing eigenvalues, suggesting that learned transformations maintain a richer lifted feature geometry.

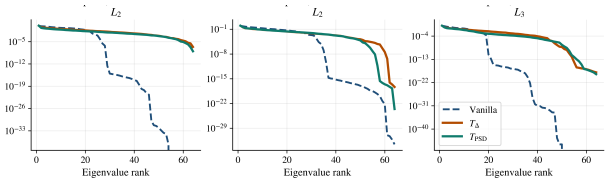


Figure 4: Vanilla KAN eigenvalues decay faster across layers.

6.2 T-KAN improves accuracy across basis families

Table 2 shows that learned basis transformations improve mean test accuracy in most matched comparisons. The gains are largest when the vanilla basis is weakest: RBF-KAN improves by 12.1 points on SVHN and 12.2 points on EMNIST-B with T_Δ, and by 5–6 points on EMNIST-L and FMNIST. Chebyshev and B-spline models start from stronger baselines, so the gains are smaller but still broadly positive, especially on EMNIST-B and EMNIST-L. Multilevel B-spline results show that basis transformations are complementary to spline refinement: except on SVHN, adding T_Δ

[1]For B-spline KANs, T_{PSD} excludes boundary basis functions which are sensitive to the outliers; $\epsilon = 10^{-3}$.

improves both refinement schedules. Both T_Δ and T_{PSD} improve test accuracy in a vast majority of the settings.

A second pattern is that T_Δ and T_{PSD} play different roles. The unconstrained T_Δ variant gives the largest mean gains on the harder or more mismatched settings, especially SVHN and EMNIST-B, where basis rotations appear useful. In contrast, T_{PSD} is often competitive on MNIST and FMNIST, where a structured rescaling is sufficient. This suggests that the useful transformation depends on how much basis-coordinate misalignment the task induces.

Table 2: T-KANs provide consistent test accuracy increases with overall lower standard deviations. Within each basis family and dataset, the best mean accuracy is bolded, including ties; standard deviations are not bolded. Two effects stand out: on SVHN, T_Δ-RBF improves vanilla RBF by 12.1 points, and on EMNIST-B all three transformed families improve over their vanilla counterparts.

Model	SVHN	EMNIST-B	EMNIST-L	FMNIST	MNIST
B-spline	62.8 ± 1.4	59.3 ± 1.0	67.2 ± 0.9	84.9 ± 0.3	92.4 ± 0.2
B-spline + T_Δ	$\mathbf{66.0} \pm 0.7$	60.4 ± 0.9	$\mathbf{68.8} \pm 0.3$	$\mathbf{85.3} \pm 0.0$	92.0 ± 0.6
B-spline + T_{PSD}	64.6 ± 1.1	$\mathbf{63.1} \pm 1.9$	67.9 ± 0.2	$\mathbf{85.3} \pm 0.3$	$\mathbf{92.7} \pm 0.1$
Multilevel $[10, 10]$	22.2 ± 2.7	60.5 ± 0.8	71.7 ± 1.0	84.9 ± 0.4	92.6 ± 0.7
$+ T_\Delta$	23.7 ± 5.2	$\mathbf{64.4} \pm 0.8$	$\mathbf{73.2} \pm 0.1$	$\mathbf{85.6} \pm 0.2$	$\mathbf{93.5} \pm 0.4$
$+ T_{\mathrm{PSD}}$	$\mathbf{24.5} \pm 4.3$	62.1 ± 1.0	72.3 ± 0.4	85.1 ± 0.3	92.8 ± 0.6
Multilevel $[8, 7, 5]$	23.9 ± 3.7	62.3 ± 0.5	73.5 ± 1.0	85.4 ± 0.4	93.0 ± 0.6
$+ T_\Delta$	23.4 ± 3.3	$\mathbf{65.6} \pm 0.8$	$\mathbf{74.6} \pm 0.2$	$\mathbf{85.8} \pm 0.2$	$\mathbf{93.6} \pm 0.4$
$+ T_{\mathrm{PSD}}$	$\mathbf{25.9} \pm 5.8$	63.6 ± 1.1	73.8 ± 0.3	85.6 ± 0.0	93.0 ± 0.5
Chebyshev	63.8 ± 1.6	64.1 ± 0.9	73.8 ± 1.0	85.9 ± 0.5	93.3 ± 0.2
Chebyshev + T_Δ	$\mathbf{66.1} \pm 1.0$	$\mathbf{66.2} \pm 0.5$	$\mathbf{75.5} \pm 0.6$	86.1 ± 0.3	93.4 ± 0.3
Chebyshev + T_{PSD}	65.7 ± 1.1	65.5 ± 1.2	74.8 ± 0.6	$\mathbf{86.2} \pm 0.3$	$\mathbf{93.9} \pm 0.1$
RBF	38.2 ± 5.0	48.6 ± 2.0	64.7 ± 1.8	81.4 ± 2.1	90.9 ± 0.6
RBF + T_Δ	$\mathbf{50.3} \pm 4.6$	$\mathbf{60.8} \pm 1.5$	70.3 ± 1.1	86.2 ± 0.5	$\mathbf{93.3} \pm 0.6$
RBF + T_{PSD}	45.8 ± 3.0	58.3 ± 1.3	$\mathbf{70.5} \pm 0.8$	$\mathbf{86.3} \pm 0.0$	92.4 ± 0.4

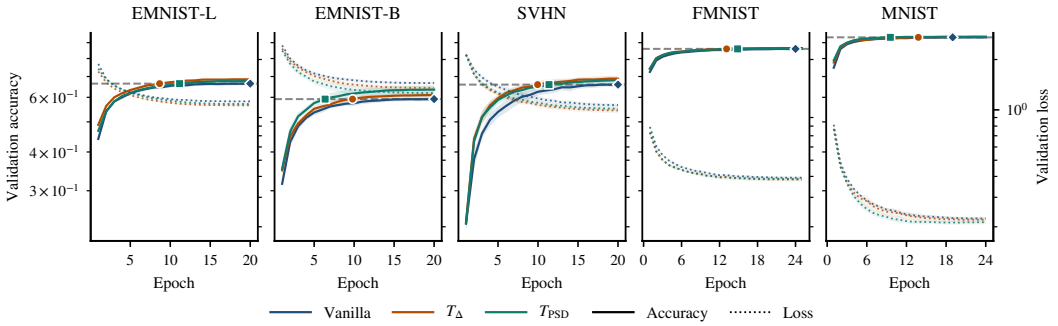


Figure 5: B-spline validation accuracy and loss over five runs with the same initialization and different minibatch orders. Curves show means and bands show one standard deviation. Markers show when the mean accuracy reaches the best parent mean accuracy.

Both transformations reach the best parent mean validation accuracy earlier on all five datasets (Figure 5). Appendix E.5 reports the crossing epochs and paired significance tests.

Learned transformations are stable across seeds. Fig. 6 shows the pairwise learned-basis difference matrix for MNIST. T_{PSD}-KAN runs learn nearly identical basis functions across runs, indicating that the constrained PSD transformation induces a stable, reproducible basis alignment. Its learned bases deviate most from the original spline basis in the first input-processing layer and the final output-facing layer, suggesting that these layers require the strongest reparameterization. T_Δ-KAN exhibits larger variation across the seeds, especially in deeper layers, consistent with its less structured unconstrained residual transformation. More details in Appendix B.1.

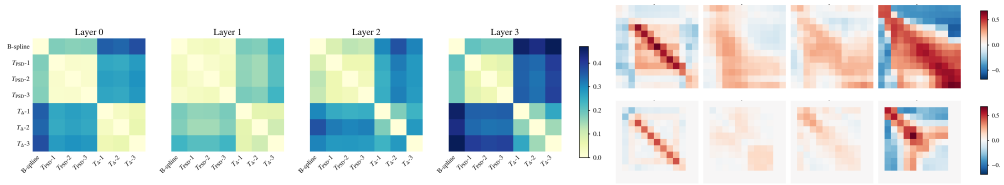


Figure 6: Left: pairwise RMS L^2 distances between learned transformed bases across MNIST runs, including the original B-spline reference; lower off-diagonal values indicate more reproducible basis learning. Right: learned transformations ($T - I$) for T_Δ-KAN and T_{PSD}-KAN for all four layers.

Additional baselines. We evaluated coefficient-block K-FAC and the official Meta Shampoo optimizer under matched datasets, architectures, basis sizes, training budgets, and checkpoint selection. Across 216 runs with RBF, B-spline, and Chebyshev KANs, adding a basis transformation improves 135 of 144 paired comparisons. The additional adaptive-backbone comparisons retain LayerNorm in every model. T-KAN improves 54 of 64 paired comparisons on the four datasets common to Free-Knots and Adaptive RBF-KAN. Appendix E reports these comparisons and their protocols.

Larger architecture and resource use. On CIFAR-100 KAN-ResNet56 with BatchNorm, mean top-one accuracy over two runs increases from 67.55% to 68.70% with T_Δ and 68.85% with T_{PSD}. Each transform adds 864 parameters (0.10%). A separate convolutional KAN forward benchmark gives time ratios of 0.980–$1.036\times$; analytical added FLOPs are below 0.08% for a full KAN-ResNet56 forward. The forward measurement protocol is given in Appendix E.4.

7 Conclusions

In this work, we proved that under SGD training, basis misalignment becomes a concrete optimization bottleneck. This work identifies fixed basis lifting as a structural source of ill-conditioning, showing that it links lifted Gramian collapse to slower convergence, lower accuracy, and seed sensitivity. T-KAN delivers a basis-coordinate transformation that targets this geometry directly. It preserves the chosen basis family while adding negligible parameters. Across bases and tasks, T-KAN improves spectra, conditioning, convergence rate and accuracy. These gains establish basis choice as both an expressivity and optimization decision. The result gives KAN design a principled route toward stable, efficient learning. **Limitations.** Our theoretical assumptions are relatively strong to account for the SGD dynamics of otherwise highly nonlinear setting. The experiments focus on small-to-moderate KANs where lifted-feature Gramian spectra, rank deficiency, and conditioning can be measured directly. While the evidence supports the mechanistic claim, testing whether the same basis-coordinate effects persist in substantially wider or deeper KANs is an important next step.

Impact statement. We do not anticipate unique ethical concerns beyond those broadly associated with more efficient ML models.

Code: `https://github.com/kaptres/kan_conditioning`

Funding and competing interests

The authors received no funding for this work and declare no competing interests.

References

Jonas A Actor, Graham Harper, Ben Southworth, and Eric C Cyr. Leveraging kans for expedient training of multichannel mlps via preconditioning and geometric refinement. *arXiv preprint arXiv:2505.18131*, 2025.

Mazlum Ferhat Arslan, Weihong Guo, and Shuo Li. Neuromanifold-regularized kans for shape-fair feature representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12790–12799, 2025.

Mazlum Ferhat Arslan, Weihong Guo, and Shuo Li. Projkan: Model compression via kan projections to bridge the hypothesis and capacity gaps. In *Advances in Neural Information Processing Systems*, 2026.

Aleksandar Botev, Hippolyt Ritter, and David Barber. Practical gauss-newton optimisation for deep learning. In *International Conference on Machine Learning*, pages 557–565. PMLR, 2017.

Roberto Cavoretto, Alessandra De Rossi, Adeeba Haider, and Amir Noorizadegan. Adaptive rbf-kan: A comparative evaluation of dynamic shape parameters in kolmogorov-arnold networks. *arXiv preprint arXiv:2605.21534*, 2026.

Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *International conference on machine learning*, pages 1675–1685. PMLR, 2019.

Grzegorz Dudek and Tomasz Rodak. Hkan: Hierarchical kolmogorov-arnold network without backpropagation. *arXiv preprint arXiv:2501.18199*, 2025.

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In *International Conference on Machine Learning*, pages 1842–1850, 2018.

Elham Kiyani, Khemraj Shukla, Jorge F Urbán, Jérôme Darbon, and George Em Karniadakis. Optimizing the optimizer for physics-informed neural networks and kolmogorov-arnold networks. *Computer Methods in Applied Mechanics and Engineering*, 446:118308, 2025.

Ziyao Li. Kolmogorov-arnold networks are radial basis function networks. *arXiv preprint arXiv:2405.06721*, 2024.

Chaoyue Liu, Han Bi, Like Hui, and Xiao Liu. Better ntk conditioning: A free lunch from (relu) nonlinear activation in wide neural networks. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y. Hou, and Max Tegmark. Kan: Kolmogorov-arnold networks. *arXiv preprint arXiv:2404.19756*, 2024. URL `https://arxiv.org/abs/2404.19756`.

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, pages 2408–2417. PMLR, 2015.

Quynh Nguyen, Marco Mondelli, and Guido F Montufar. Tight bounds on the smallest eigenvalue of the neural tangent kernel for deep relu networks. In *International Conference on Machine Learning*, pages 8119–8129. PMLR, 2021.

Jeffrey Pennington, Samuel Schoenholz, and Surya Ganguli. Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice. *Advances in neural information processing systems*, 30, 2017.

Michael Poluektov and Andrew Polar. Construction of the kolmogorov-arnold representation using the newton-kaczmarz method. *arXiv preprint arXiv:2305.08194*, 2023.

Spyros Rigas, Michalis Papachristou, Theofilos Papadopoulos, Fotios Anagnostopoulos, and Georgios Alexandridis. Adaptive training of grid-dependent physics-informed kolmogorov-arnold networks. *IEEE Access*, 12:176982–176998, 2024.

Spyros Rigas, Dhruv Verma, Georgios Alexandridis, and Yixuan Wang. Initialization schemes for kolmogorov-arnold networks: An empirical study. *arXiv preprint arXiv:2509.03417*, 2025.

Samuel S Schoenholz, Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein. Deep information propagation. *arXiv preprint arXiv:1611.01232*, 2016.

Khemraj Shukla, Juan Diego Toscano, Zhicheng Wang, Zongren Zou, and George Em Karniadakis. A comprehensive and fair comparison between mlp and kan representations for differential equations and operator networks. *Computer Methods in Applied Mechanics and Engineering*, 431:117290, 2024.

S. S. Sidharth, R. Gokul, K. P. Anas, and A. R. Keerthana. Chebyshev polynomial-based kolmogorov-arnold networks: An efficient architecture for nonlinear function approximation. *arXiv preprint arXiv:2405.07200*, 2024. URL https://arxiv.org/abs/2405.07200.

Shairoz Sohail. On training of kolmogorov-arnold networks. *arXiv preprint arXiv:2411.05296*, 2024. URL https://arxiv.org/abs/2411.05296.

Ben S Southworth, Jonas A Actor, Graham Harper, and Eric C Cyr. Multilevel training for kolmogorov arnold networks. *arXiv preprint arXiv:2603.04827*, 2026.

Aditya Varre, Margarita Sagitova, and Nicolas Flammarion. Sgd vs gd: Rank deficiency in linear networks. *Advances in Neural Information Processing Systems*, 37:60133–60161, 2024.

Lechao Xiao, Yasaman Bahri, Jascha Sohl-Dickstein, Samuel Schoenholz, and Jeffrey Pennington. Dynamical isometry and a mean field theory of cnns: How to train 10,000-layer vanilla convolutional neural networks. In *International conference on machine learning*, pages 5393–5402. PMLR, 2018.

Lechao Xiao, Jeffrey Pennington, and Samuel Schoenholz. Disentangling trainability and generalization in deep neural networks. In *International Conference on Machine Learning*, pages 10462–10472. PMLR, 2020.

Runpeng Yu, Weihao Yu, and Xinchao Wang. Kan or mlp: A fairer comparison. *arXiv preprint arXiv:2407.16674*, 2024. URL https://arxiv.org/abs/2407.16674.

Jim Zhao, Sidak Pal Singh, and Aurelien Lucchi. Theoretical characterisation of the Gauss-Newton conditioning in neural networks. *Advances in Neural Information Processing Systems*, 37:114965–115000, 2024.

Liangwewi Nathan Zheng, Wei Emma Zhang, Lin Yue, Miao Xu, Olaf Maennel, and Weitong Chen. Free-knots kolmogorov-arnold network: On the analysis of spline knots and advancing stability. *arXiv preprint arXiv:2501.09283*, 2025.

A Detailed empirical setup

Our goal in the experiments is to isolate a conditioning mechanism in KANs under controlled settings where lifted-feature Gramian spectra can be measured directly. We therefore use small-to-moderate KANs with matched depth, width, basis size, optimizer, seeds, and training budgets across vanilla and transformed variants.

Models. All image experiments use flattened inputs scaled to $[0, 1]$, the first-layer spline grid is initialized on $[0, 1]$, and hidden-layer spline grids are initialized on $[-1, 1]$ with hidden activations clipped to this range before the next layer. We use SiLU base functions, trainable spline/base scales, and initial spline noise scale 0.01. Unless stated otherwise, MNIST, FashionMNIST, EMNIST-Balanced, and EMNIST-Letters use three hidden KAN layers of width 8, while SVHN uses width 16 to match the larger input dimension. We evaluate the same depth and width across B-spline, Chebyshev, and RBF bases, and compare vanilla KANs against T_Δ-KAN and T_{PSD}-KAN variants. The reported results use $B = 6$ throughout.

Training. For the main image-classification runs, models are trained with cross-entropy loss using SGD, learning rate 10^{-1}, weight decay 10^{-4}, cosine learning-rate decay. We use batch size 256 for MNIST, FashionMNIST, EMNIST-Balanced, and EMNIST-Letters, and batch size 512 for SVHN. MNIST and FashionMNIST are trained for 25 epochs; EMNIST-Balanced, EMNIST-Letters, and SVHN are trained for 20 epochs. Validation splits are held out from the training set: 5000 examples for MNIST, FashionMNIST, and SVHN, and 10000 for EMNIST. All reported test accuracies use the official test split. Every comparison is paired: vanilla KAN, T_Δ-KAN, and T_{PSD}-KAN use the same dataset split, seed, optimizer, epoch budget, batch size, and basis family.

Implementations. For basis variants (B-spline, Chebyshev KAN, RBF KAN) we use the repositories indicated in the respective papers. T_Δ-KAN augments each layer with an input-independent residual basis transformation $T = I + \Delta$, while T_{PSD}-KAN uses a structured positive-semidefinite transformation. These transformations are trained jointly with the KAN weights.

For the multilevel baseline, all multilevel runs start from basis size $B_0 = 4$. We evaluate two refinement schedules: $[10, 10]$, which trains for 10 epochs, refines once, then trains for another 10 epochs; and $[8, 7, 5]$, which performs two refinements. These schedules end at $B = 5$ and $B = 7$, respectively.

For hybrid multilevel T-KAN, KAN layer is wrapped with either a T_Δ or T_{PSD} basis transformation during each training stage. Before each spline refinement, the learned transform is absorbed into the spline coefficients. We then refine this merged plain KAN layer using the standard multilevel grid-refinement routine, and wrap the refined layer with a fresh identity-initialized transformation for the next stage. Thus the transform participates at each resolution, is merged into the basis coefficients before refinement, and is relearned after refinement.

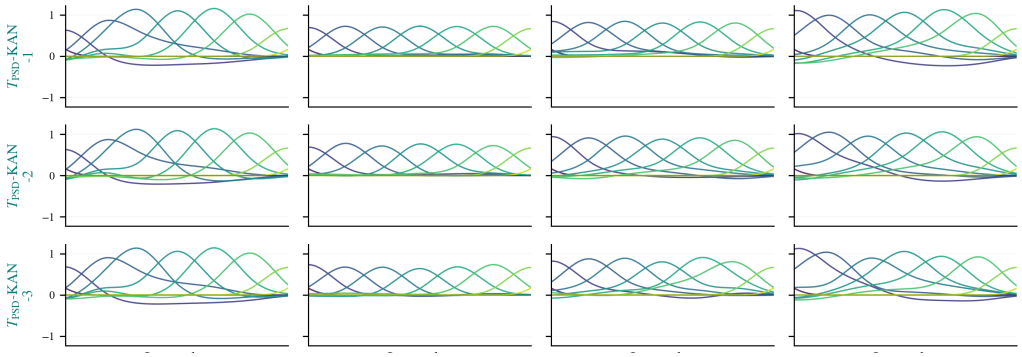
B Further results

B.1 More on learned transformations of T-KANs.

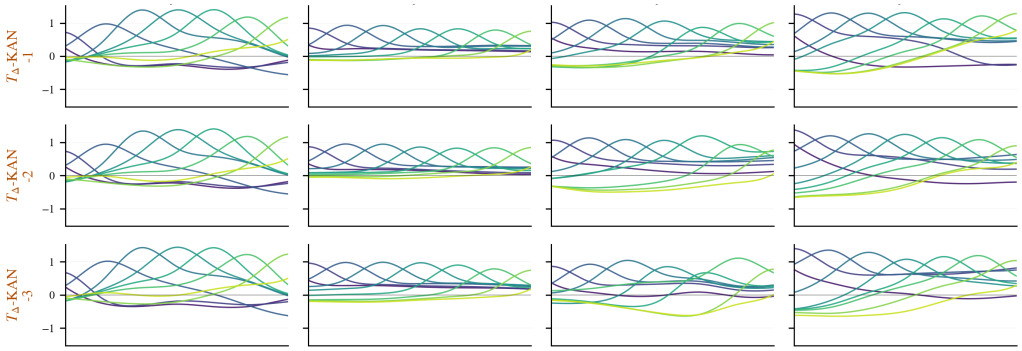
In Fig. 9, we used

$$D_{rs}^{(\ell)} = \left[\frac{1}{K|\Omega|} \sum_{i=1}^{K} \int_{\Omega} \left(\psi_{\ell,i}^{(r)}(x) - \psi_{\ell,i}^{(s)}(x) \right)^2 dx \right]^{1/2} \tag{25}$$

where $\psi_{\ell,i}^{(r)}$ denotes the i-th transformed basis function at layer ℓ for run r, and the untransformed B-spline basis is included as the reference, as a measure of basis distance to acquire the comparison heatmaps. Fig. 7a–7b displays the learned basis functions by T_{PSD}-KAN and T_Δ-KANs across seeds.



(a) T_{PSD}-KAN learned basis functions. In agreement with lower computed distances, the learned basis are nearly the same across different seeds.



(b) T_{Δ}-KAN learned basis functions. While the computed distance by Eq. 25 is high between the runs, common patterns can be observed in the learned basis functions.

Figure 7: The exact learned basis functions on MNIST across 3 seeds by T_{PSD}/T_{Δ}-KANs (B-spline). The main basis transformations occur at the input layer and output layers; the near-identity promoting design and regularization effectively aligns the hidden feature vectors.

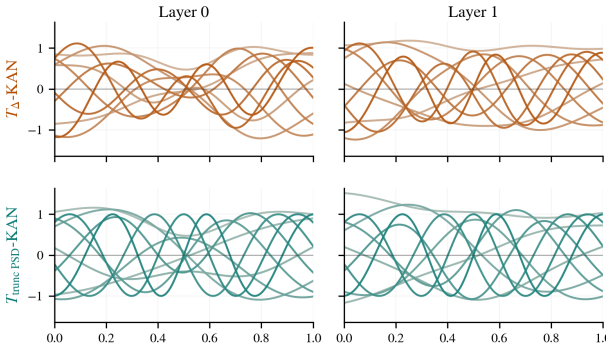


Figure 8: The learned basis functions for Chebyshev KANs with $B = 10$ on reaction-diffusion.

B.2 Transfer learning via T_{Δ}-KAN

Fig. 9 briefly explores transfer learning from MNIST to FMNIST by only updating the learned transformations, keeping more than 99% of the network fixed. (constant $lr = 3e{-}2$, 20 epochs). The results indicate T_{Δ}-KAN leads to the learning of generalizable weights W_{ℓ} on similar tasks, where $B \in \{10, 20\}$ exceed 60% FMNIST test accuracy. T_{PSD}-KAN did not show similar behavior, likely due to extra structural restrictions. This suggests that T_{Δ} can act as a lightweight task-adaptation interface, separating reusable spline

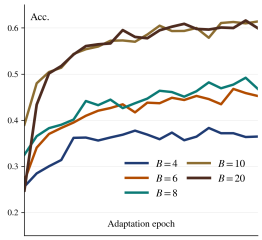


Figure 9: Updating $< 0.5\%$ of parameters can transfer T_{Δ}-KANs.

weights from dataset-specific basis realignment when the tasks are close to one another.

B.3 Diagonal preconditioning further improves task performance and convergence speeds for T-KAN

Figure 10 shows that T-KAN's task performance and convergence speed improve further by diagonal preconditioning, such as that due to BatchNorm. Although such preconditioning improves the eigenvalue spectrum, it does not resolve the underlying basis-alignment problem. In fact, the learned basis transformations in T_Δ-KAN and T_{PSD}-KAN improve the diagonal preconditioning effect: both transformed KAN variants achieve faster convergence and better final test performance than the vanilla KAN.

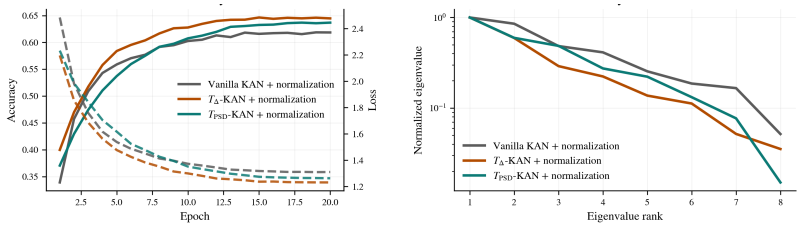


Figure 10: Effect of additional diagonal preconditioning for vanilla KAN, T_Δ-KAN, and T_{PSD}-KAN. Final test accuracies: T_Δ-KAN: **63.7%**, T_{PSD}-KAN: **63.3%**, KAN: 61.8%.

C Proofs for Section 4

C.1 Residual rank and drift proofs

Lemma 1 (Residual rank accounting)**.** *Let $U \in \mathbb{R}^{N\times d}$, let $H = q(U)$, and define $H_\perp = (I - P_U)H$. Then* $\mathrm{rank}[U, H] = \mathrm{rank}(U) + \mathrm{rank}(H_\perp)$. *Consequently,* $\mathrm{rank}(S) = \mathrm{rank}(H_\perp)$ *for* $S = N^{-1}H_\perp^\top H_\perp$.

Proof of Lemma 1. Write $H = P_U H + H_\perp$. The columns of $P_U H$ lie in $\mathrm{col}(U)$, so they add no rank beyond the columns of U. The columns of $H_\perp$ lie in $\mathrm{col}(U)^\perp$, so their rank adds independently. This gives $\mathrm{rank}[U, H] = \mathrm{rank}(U) + \mathrm{rank}(H_\perp)$. Since $S = N^{-1}H_\perp^\top H_\perp$, the Gram identity $\mathrm{rank}(A^\top A) = \mathrm{rank}(A)$ gives $\mathrm{rank}(S) = \mathrm{rank}(H_\perp)$. □

Proof of Theorem 2. Apply Itô's formula to $f(U) = \log\det S(U)$. The first derivative gives the task contribution $Df(U_t)[a_t] = \langle A_S(U_t), a_t\rangle_F$, and the martingale part is $dM_t = \sum_m Df(U_t)[V_{m,t}]\, dB_t^m$. The second order term is $\frac{1}{2}\sum_m D^2 f(U_t)[V_{m,t}, V_{m,t}]\, dt$. For the log determinant,

$$D^2 f[V, V] = \mathrm{tr}(S^{-1}D^2 S[V, V]) - \mathrm{tr}(S^{-1}DS[V]S^{-1}DS[V]).$$

Thus the drift equals $\langle A_S, a_t\rangle_F - \mathcal{K}_S + \mathcal{E}_S$. The dominance condition (9) makes this drift at most $-\rho$, which proves (10). □

Proof of Proposition 3. Let V be a deterministic feature velocity. Since $H = q(U)$, $DH[V] = q'(U) \odot V$. On a fixed-rank neighborhood, the differential of the residual block may be inserted against $H_\perp^\top$ as

$$H_\perp^\top DH_\perp[V] = H_\perp^\top\big(q'(U) \odot V - VU^\dagger H\big).$$

The terms involving the derivative of the projector that land in $\mathrm{col}(U)$ vanish because $H_\perp^\top U = 0$. Since $S = N^{-1}H_\perp^\top H_\perp$,

$$D\log\det S[V] = \mathrm{tr}(S^{-1}DS[V]) = \frac{2}{N}\,\mathrm{tr}\big(S^{-1}H_\perp^\top DH_\perp[V]\big).$$

Rewriting this last expression as a Frobenius inner product with V gives (11). If b is a simple unit eigenvector for $\lambda_{\min}(S)$, standard eigenvalue perturbation gives $D\lambda_{\min}(S)[V] = b^\top DS[V]b$, and the same calculation gives (12).

Now let $\dot{U} = M + \Xi$. The determinant drift is $D_{\det} = \langle A_{\det}, M\rangle_F + \langle A_{\det}, \Xi\rangle_F$. By assumption, the second term is sub-Gaussian with proxy $\sigma^2\|A_{\det}\|_F^2$. If $\langle A_{\det}, M\rangle_F \le -\gamma$, then

$$\mathbb{P}(D_{\det} > -\gamma/2) \le \exp\left(-\frac{\gamma^2}{8\sigma^2\|A_{\det}\|_F^2}\right),$$

and taking complements proves (13). The proof for $\lambda_{\min}(S)$ is identical with A_b in place of $A_{\det}$. □

Proof of Theorem 4. Let $F_t = \log\det S_t$, let $a_0 = F_0 - \log\delta > 0$, and stop the process at $\tau_\delta \wedge \zeta$. On the event $\tau_\delta > T$ and $\zeta > T$, the determinant has not hit δ, so $F_T - \log\det S_0 > -a_0$. The stopped drift inequality gives $F_T - \log\det S_0 \le -\rho T + M_T$. Hence this event implies $M_T \ge \rho T - a_0$. The quadratic variation bound and the standard sub-Gaussian martingale tail yield

$$\mathbb{P}(\tau_\delta > T, \zeta > T) \le \exp\left(-\frac{(\rho T - a_0)_+^2}{2\Lambda T}\right).$$

Rearranging gives (14). Finally, $\det S_t \le \delta$ implies $\lambda_{\min}(S_t)^{r_S} \le \det S_t \le \delta$, so $\lambda_{\min}(S_t) \le \delta^{1/r_S}$. □

Proof of Theorem 5. Apply Theorem 2 with the residual covariance S_B in place of S. The first derivative of $\log\det S_B$ is $A_B^{\det}$, the nonnegative log-determinant Hessian compression is $\mathcal{K}_B$, and the mixed-sign second-order term is $\mathcal{E}_B$. The dominance condition (16) therefore makes the drift at most $-\rho_{\psi'}$, giving the claimed inequality. □

Curvature is a refinement of the same residual drift decomposition in second-order. While derivative-noise compression is nonnegative, curvature needs stronger assumptions for negative drift since the ψ'' term has no fixed sign.

Lemma 2 (Curvature assisted residual compression)**.** *Under the assumptions of Theorem 5, suppose that, on the local window, the mixed-sign term admits the decomposition $\mathcal{E}_B = -\mathcal{C}_B + \mathcal{R}_B$, where $\mathcal{C}_B \ge 0$ is the favorable curvature compression and $\mathcal{R}_B \ge 0$ is the adverse curvature and projection remainder. If*

$$\mathcal{K}_B(t) + \mathcal{C}_B(t) \ge \langle A_B^{\det}(U_t), a_t\rangle_{F,+} + \mathcal{R}_B(t) + \rho_B \tag{26}$$

for some $\rho_B > 0$, then $d\log\det S_{B,t} \le -\rho_B\,dt + dM_t$ for a continuous local martingale M_t.

Proof of Lemma 2. The proof starts from the same Itô decomposition used in Theorem 5:

$$d\log\det S_{B,t} = \big(\langle A_B^{\det}, a_t\rangle_F - \mathcal{K}_B + \mathcal{E}_B\big)dt + dM_t.$$

By the assumed decomposition $\mathcal{E}_B = -\mathcal{C}_B + \mathcal{R}_B$, the drift is at most $\langle A_B^{\det}, a_t\rangle_{F,+} - \mathcal{K}_B - \mathcal{C}_B + \mathcal{R}_B$. Condition (26) bounds this quantity by $-\rho_B$, proving the claim. □

Proof of Theorem 6. Escape to β requires $\log F_t - \log F_0 \ge \log(\beta/F_0)$ for some $t \le \zeta$. The stopped drift inequality gives $\log F_t - \log F_0 \le -\rho t + M_t$, so escape before local exit requires $M_t - \rho t \ge \log(\beta/F_0)$ for some stopped time. For $\theta = 2\rho/\Lambda$, the exponential process $\exp(\theta M_t - \theta^2\langle M\rangle_t/2)$ is a supermartingale, and $\langle M\rangle_t \le \Lambda t$. Ville's inequality gives $\mathbb{P}(\sigma_\beta \le \zeta) \le \exp(-\theta\log(\beta/F_0))$, which is (17). □

Proof of Proposition 7. From (11),

$$\|A_{\det}\|_F \le \frac{2}{N}\Big(\|q'(U)\odot(H_\perp S^{-1})\|_F + \|H_\perp S^{-1}(U^\dagger H)^\top\|_F\Big).$$

Using $\|H_\perp S^{-1}\|_F \le \|H_\perp\|_F\|S^{-1}\|_{\mathrm{op}}$ and $\|S^{-1}\|_{\mathrm{op}} = 1/\lambda_{\min}(S)$ gives (18).

For the lower bound, the trace in (7) is the squared Frobenius norm of $S^{-1/2}DS[V_m]S^{-1/2}$. Testing this matrix on the unit vector b gives

$$\mathrm{tr}(S^{-1}DS[V_m]S^{-1}DS[V_m]) \ge \frac{(b^\top DS[V_m]b)^2}{\lambda_{\min}(S)^2}.$$

Summing over m and multiplying by $1/2$ proves (19); the final statement follows by keeping one noise direction with $|b^\top DS[V_m]b| \geq c$. □

D Condition Number Improvement under Basis-Coordinate Separability

Proposition 9 (Existence of an ideal basis preconditioner)**.** *Suppose the lifted feature Gramian satisfies the exact basis-coordinate factorization $G_\ell = C_\ell \otimes S_\ell$, where $C_\ell \in \mathbb{R}^{B\times B}$ and $S_\ell \in \mathbb{R}^{d_\ell \times d_\ell}$ are symmetric positive definite. Then there exists an invertible basis transformation T such that the transformed Gramian $G_{\ell,T} = (T^\top \otimes I_{d_\ell})G_\ell(T \otimes I_{d_\ell})$ has no basis-side conditioning. In particular, $T_\star = C_\ell^{-1/2}$ gives*

$$G_{\ell,T_\star} = I_B \otimes S_\ell. \tag{27}$$

Proof. Using $G_\ell = C_\ell \otimes S_\ell$ and the mixed-product rule for Kronecker products, we obtain $G_{\ell,T} = (T^\top C_\ell T) \otimes S_\ell$. Since $C_\ell \succ 0$, the inverse square root $C_\ell^{-1/2}$ exists and is invertible. Setting $T = T_\star = C_\ell^{-1/2}$ gives $T_\star^\top C_\ell T_\star = I_B$, and therefore $G_{\ell,T_\star} = I_B \otimes S_\ell$. The additive parameterization can represent this ideal transformation by taking $\Delta = C_\ell^{-1/2} - I$. The PSD parameterization can represent it whenever $\epsilon < \lambda_{\min}(C_\ell^{-1/2})$, by choosing $LL^\top = C_\ell^{-1/2} - \epsilon I$. □

Proposition 10 (Optimal condition-number improvement)**.** *Under the assumptions of the previous proposition, the best condition number achievable by any invertible basis transformation T is obtained when $T^\top C_\ell T$ is a scalar multiple of the identity. In particular, $T_\star = C_\ell^{-1/2}$ achieves the optimum, and*

$$\min_{T\in GL(B)} \kappa(G_{\ell,T}) = \kappa(S_\ell), \qquad \frac{\kappa(G_\ell)}{\kappa(G_{\ell,T_\star})} = \kappa(C_\ell). \tag{28}$$

Proof. For any invertible T, the transformed Gramian is $G_{\ell,T} = (T^\top C_\ell T) \otimes S_\ell$. Since both factors are positive definite, the condition number satisfies $\kappa(G_{\ell,T}) = \kappa(T^\top C_\ell T)\kappa(S_\ell)$. The first factor is always at least one, with equality exactly when $T^\top C_\ell T$ is a scalar multiple of I_B. Therefore the smallest achievable condition number is $\kappa(S_\ell)$, achieved by $T_\star = C_\ell^{-1/2}$. Since $\kappa(G_\ell) = \kappa(C_\ell)\kappa(S_\ell)$ and $\kappa(G_{\ell,T_\star}) = \kappa(S_\ell)$, the improvement factor is $\kappa(C_\ell)$. □

E Additional comparisons and uncertainty

We test whether T-KAN's gains persist with curvature-aware optimizers, adaptive bases, and standard normalization. We then report the larger CIFAR-100 comparison, forward time and FLOPs, and the uncertainty analysis for Figure 5.

E.1 Curvature-aware optimizer comparisons

We evaluated coefficient-block K-FAC [Martens and Grosse, 2015] and the official Meta Shampoo implementation [Gupta et al., 2018] under matched datasets, architectures, basis sizes, training budgets, and checkpoint selection. Tables 3–5 report 216 runs with RBF, B-spline, and Chebyshev KANs. Parent and transformed models share initialization and minibatch order within each comparison.

Overall, T-KAN improves 135 of 144 paired comparisons, showing that its benefit is complementary to these preconditioning methods. The reported standard deviations measure variation in minibatch order at fixed initialization.

Table 3: Gaussian RBF-KAN test accuracy (%), mean ± sample standard deviation over 5 matched runs.

Dataset	Optimizer	KAN	$+T_{\Delta}$	$+T_{\mathrm{PSD}}$
EMNIST-B	K-FAC	66.00 ± 0.87	68.76 ± 0.37	68.32 ± 0.57
EMNIST-B	Shampoo	64.23 ± 1.59	67.73 ± 0.62	66.63 ± 0.85
EMNIST-L	K-FAC	74.72 ± 0.41	76.92 ± 0.63	76.08 ± 0.45
EMNIST-L	Shampoo	74.02 ± 0.48	75.99 ± 0.35	75.28 ± 0.41
Fashion-MNIST	K-FAC	86.43 ± 0.21	86.94 ± 0.17	86.75 ± 0.07
Fashion-MNIST	Shampoo	84.26 ± 0.83	85.10 ± 0.37	85.56 ± 0.34
SVHN	K-FAC	26.33 ± 4.18	30.33 ± 3.39	32.52 ± 6.48
SVHN	Shampoo	42.86 ± 11.70	54.50 ± 8.52	52.11 ± 7.73

Table 4: Cubic B-spline KAN test accuracy (%), mean ± sample standard deviation over 2 matched runs.

Dataset	Optimizer	KAN	$+T_{\Delta}$	$+T_{\mathrm{PSD}}$
EMNIST-B	K-FAC	68.25 ± 0.24	69.52 ± 0.11	69.81 ± 0.77
EMNIST-B	Shampoo	65.11 ± 0.96	68.45 ± 0.10	68.08 ± 0.56
EMNIST-L	K-FAC	77.02 ± 0.28	77.72 ± 0.28	78.31 ± 0.06
EMNIST-L	Shampoo	74.84 ± 0.25	76.66 ± 0.93	76.13 ± 0.68
Fashion-MNIST	K-FAC	86.43 ± 0.24	86.53 ± 0.12	86.36 ± 0.12
Fashion-MNIST	Shampoo	85.86 ± 0.16	86.02 ± 0.16	85.88 ± 0.20
SVHN	K-FAC	63.51 ± 0.36	64.20 ± 0.13	64.91 ± 0.05
SVHN	Shampoo	62.13 ± 0.88	62.35 ± 0.61	64.71 ± 0.11

Table 5: Chebyshev KAN test accuracy (%), mean ± sample standard deviation over 2 matched runs.

Dataset	Optimizer	KAN	$+T_{\Delta}$	$+T_{\mathrm{PSD}}$
EMNIST-B	K-FAC	68.39 ± 0.50	69.92 ± 0.05	70.01 ± 0.66
EMNIST-B	Shampoo	67.13 ± 0.19	69.16 ± 0.03	68.39 ± 0.86
EMNIST-L	K-FAC	77.83 ± 0.07	78.07 ± 0.54	78.50 ± 0.26
EMNIST-L	Shampoo	75.57 ± 0.36	76.87 ± 0.81	76.81 ± 1.42
Fashion-MNIST	K-FAC	85.62 ± 0.01	86.25 ± 0.23	86.27 ± 0.28
Fashion-MNIST	Shampoo	85.66 ± 0.19	85.91 ± 0.09	85.84 ± 0.30
SVHN	K-FAC	64.78 ± 0.18	64.93 ± 2.31	65.22 ± 3.10
SVHN	Shampoo	63.65 ± 1.75	65.00 ± 0.08	66.06 ± 0.33

E.2 Adaptive KAN backbones with LayerNorm

The LayerNorm controls use the official Free-Knots implementation [Zheng et al., 2025] and our implementation of the learned Gaussian shape parameter from Adaptive RBF-KAN [Cavoretto et al., 2026] on the FastKAN parent [Li, 2024]. LayerNorm is retained in every parent and transformed model. Across the four common datasets, T-KAN records 54 wins, three ties, and seven losses over 64 paired comparisons (Table 6). Thus, the gains persist with both normalization and adaptive basis parameters already present.

Free-Knots uses three matched runs that vary both initialization and minibatch order. Adaptive RBF uses five runs that vary minibatch order at fixed initialization. For Adaptive RBF, the shared Gaussian shape parameter starts at $h = 0.8$, without leave-one-out cross-validation for initialization. Failed Free-Knots runs on EMNIST-B are scored as zero: the parent fails on two of three runs, and each transformed model on one.

Table 6: Adaptive backbones with LayerNorm retained. Test accuracy (%), mean $\pm$ sample standard deviation over three Free-Knots or five Adaptive-RBF runs. Free-Knots EMNIST-B includes failed runs as zero. Adaptive RBF is our implementation of learned Gaussian shape adaptation, initialized at $h = 0.8$.

Backbone	Dataset	Parent	$+T_{\Delta}$	$+T_{\mathrm{PSD}}$
Free-Knots	MNIST	94.67 ± 0.08	95.83 ± 0.18	95.66 ± 0.16
Free-Knots	Fashion-MNIST	84.91 ± 0.05	86.95 ± 0.13	86.67 ± 0.13
Free-Knots	EMNIST-B	25.13 ± 43.52	51.15 ± 44.29	50.88 ± 44.07
Free-Knots	EMNIST-L	82.36 ± 0.17	84.44 ± 0.48	84.01 ± 0.41
Adaptive RBF	MNIST	93.17 ± 0.24	93.41 ± 0.27	93.50 ± 0.08
Adaptive RBF	Fashion-MNIST	86.09 ± 0.14	86.20 ± 0.23	86.11 ± 0.17
Adaptive RBF	EMNIST-B	63.53 ± 0.76	64.35 ± 0.84	64.25 ± 1.07
Adaptive RBF	EMNIST-L	70.33 ± 1.05	72.30 ± 1.07	72.79 ± 1.17

E.3 CIFAR-100 KAN-ResNet56 with BatchNorm

We trained six KAN-ResNet56 models from scratch on CIFAR-100 using a 200-epoch schedule: three variants and two runs. All models use degree-three Chebyshev convolutions, nine residual blocks per stage, BatchNorm before and after every ConvKAN layer, and identical AdamW optimization and augmentation.

Both transformations improve top-one accuracy and test loss in both runs (Table 7), while adding only 864 parameters, approximately 0.10% of the parent model. T-KAN therefore remains useful with BatchNorm and beyond the small fully connected architectures.

Table 7: CIFAR-100 KAN-ResNet56 test top-one accuracy (%) and test loss for two runs.

Model	Parameters	Accuracy, runs 1/2	Test loss, runs 1/2
KAN-ResNet56	856,124	67.9 / 67.2	1.915 / 1.918
$+T_{\Delta}$	856,988	68.6 / 68.8	1.904 / 1.903
$+T_{\mathrm{PSD}}$	856,988	69.0 / 68.7	1.900 / 1.904

E.4 Forward time and FLOPs

Across $Q \in \{4, 6, 8, 10\}$, measured forward time is 0.980–$1.036\times$ the parent layer's time (Table 8). For a full KAN-ResNet56 forward at batch size 64, the added FLOPs range from 0.031% at $Q = 4$ to 0.076% at $Q = 10$. The learned convolutional basis transform therefore adds negligible forward-pass overhead.

Table 8: Convolutional KAN forward time and FLOPs. Parent times are in milliseconds; transformed times are relative to the parent at the same basis size Q.

Q	Parent (ms)	T_Δ time	T_{PSD} time	Parent FLOPs	Added FLOPs
4	12.75	$1.009\times$	$1.003\times$	316.1M	0.074M (0.023%)
6	18.08	$1.000\times$	$1.036\times$	481.3M	0.166M (0.035%)
8	24.75	$1.006\times$	$1.013\times$	646.4M	0.295M (0.046%)
10	33.55	$0.980\times$	$0.980\times$	811.6M	0.460M (0.057%)

E.5 Uncertainty for Figure 5

Figure 5 reports five runs with the same model initialization and different minibatch orders. Curves show means and bands show one sample standard deviation. Each paired two-sided t-test uses one epoch-averaged accuracy or loss per run. Holm correction is applied separately to the ten accuracy and ten loss comparisons, at familywise level 0.05.

The common training horizons are 20 epochs for EMNIST and SVHN, 25 for Fashion-MNIST, and 24 for MNIST. Six of ten accuracy comparisons and all ten loss comparisons remain significant after correction (Tables 9–10). The crossing epochs in Table 11 are linearly interpolated from the mean accuracy curves. Both transformations reach the best parent mean accuracy earlier on every dataset.

Table 9: Epoch-averaged validation accuracy (%), mean $\pm$ sample SD over five matched runs. p_{H} is the Holm-adjusted paired two-sided t-test value against the parent.

Dataset	Epochs	Parent	T_Δ	p_{H}	T_{PSD}	p_{H}
EMNIST-L	20	63.06 ± 0.74	65.14 ± 0.35	0.032	63.89 ± 0.45	0.186
EMNIST-B	20	54.96 ± 0.90	56.62 ± 0.91	0.164	58.97 ± 1.64	0.047
SVHN	20	58.03 ± 1.47	61.84 ± 1.28	0.016	61.05 ± 1.51	0.034
Fashion-MNIST	25	84.16 ± 0.22	84.88 ± 0.18	0.030	84.70 ± 0.34	0.115
MNIST	24	91.75 ± 0.19	92.21 ± 0.55	0.186	92.60 ± 0.19	0.032

Table 10: Epoch-averaged validation loss, mean $\pm$ sample SD, using the same common horizons as Table 9. Adjusted p-values use a separate family of ten loss comparisons.

Dataset	Parent	T_Δ	p_{H}	T_{PSD}	p_{H}
EMNIST-L	1.234 ± 0.011	1.170 ± 0.013	0.005	1.204 ± 0.009	0.044
EMNIST-B	1.599 ± 0.030	1.518 ± 0.043	0.044	1.423 ± 0.067	0.044
SVHN	1.269 ± 0.033	1.174 ± 0.027	0.018	1.198 ± 0.032	0.044
Fashion-MNIST	0.443 ± 0.004	0.425 ± 0.002	0.005	0.427 ± 0.007	0.044
MNIST	0.286 ± 0.005	0.277 ± 0.010	0.044	0.264 ± 0.005	0.044

Table 11: Interpolated epoch at which each five-seed mean accuracy curve first reaches the best parent mean accuracy within the common training horizon.

Dataset	Parent	T_{Δ}	T_{PSD}
EMNIST-L	20.00	8.67	11.16
EMNIST-B	20.00	9.80	6.38
SVHN	20.00	9.95	11.37
Fashion-MNIST	24.00	13.11	14.85
MNIST	19.00	13.80	9.56

E.6 Asset sources and code

The benchmarks are MNIST, Fashion-MNIST, EMNIST, SVHN, and CIFAR-100, with the original owners credited at these sources. Fashion-MNIST and Free-Knots use the MIT license; FastKAN uses Apache-2.0. SVHN is restricted to non-commercial use. Code: `https://github.com/kaptres/kan_conditioning`.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Sections 4 and 6 state the theoretical assumptions and report the empirical comparisons.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The limitations in Section 7 discuss the strength of the theoretical assumptions and the small-to-moderate scale of the experiments.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: Assumptions are stated with the results. Appendix C gives the drift proofs, and Appendix D gives the ideal-transform proofs.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Appendices A and E describe the base experiments, matched comparisons, and statistical procedures.

   Guidelines:

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

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

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

Answer: [No]

Justification: `https://github.com/kaptres/kan_conditioning`; experimental settings are described in Appendices A and E.

Guidelines:

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

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: Appendices A, E.1, E.2, and E.3 describe the experimental settings and matched comparisons.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: Figure 5 and Appendix E.5 define the sample-standard-deviation bands, fixed initialization, paired tests, and Holm corrections. Tables report their seed counts and sample standard deviations.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

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

8. **Experiments compute resources**

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

   Answer: [No]

   Justification: Appendix E.4 reports forward time and FLOPs. Total training compute is not reported.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The work uses established benchmarks and does not collect new human-subject data.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The impact statement in Section 7 discusses the scope of societal impacts for this optimization study.

Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper does not release high-risk data or pretrained models.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [No]

    Justification: Appendix E.6 credits the existing assets and gives their known licenses.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: N/A

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The work does not involve crowdsourcing or direct interaction with human participants.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The work does not involve a new human-subject study.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs are not a component of the proposed architecture or optimization method.

    Guidelines:

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