Evaluating the Effect of Numerical Solvers on Learning in Neural ODEs

Anonymous authors
Paper under double-blind review

Abstract

Continuous-depth neural networks, or Neural ODEs, are usually evaluated based on accuracy, speed, or memory use, and the numerical solver is often treated as a technical implementation choice. In this paper, we provide evidence that this assumption can fail under controlled training conditions: solver choice is associated with measurable changes in the learned representation dynamics. We first examine this using a frozen-weight intervention: a model is trained with one solver and then evaluated with another, without changing the weights. This replacement can lead to a large drop in performance, indicating that the solver is not always functionally interchangeable after training under the tested setup. To investigate this behavior, we introduce a third-order Neural ODE formulation that enables direct measurement of higher-order trajectory dynamics and representation evolution without requiring nested differentiation through the solver. Across CIFAR-10, CIFAR-100, and Tiny-ImageNet, we observe that different solvers produce distinct representation dynamics, even when final clean accuracy is closely matched. We also use adversarial robustness as a downstream diagnostic of these learned representations. In some regimes, models with larger higher-order trajectory-norm quantities tend to exhibit improved robustness under iterative attacks. However, at larger integration horizons, robustness appears to depend more strongly on how the learned dynamics interact with the solver stability envelope, making the relationship regime-dependent. Finally, we provide evidence that even solvers with the same external stability polynomial can lead to different learned behaviors due to differences in internal stage realization. Overall, our results suggest that numerical integration can act as an important factor in Neural ODE training, with measurable associations to learned representation geometry and robustness beyond clean predictive accuracy.

1 Introduction

Continuous-depth neural networks model feature evolution as the solution of an ordinary differential equation (ODE) and are commonly implemented using explicit Runge–Kutta (RK) integrators Chen et al. (2018). Prior work has primarily evaluated solver choice in terms of predictive accuracy, computational efficiency, adjoint consistency, and memory usage Chen et al. (2018); Zhuang et al. (2020); Kidger (2022). In this view, the numerical integrator is treated as a computational tool for approximating trajectories. Much less is known about whether solver choice influences the representation dynamics learned during training.

In this work, we provide intervention-based evidence that the choice of numerical integrator is associated with learned representation dynamics in Neural ODEs. In particular, we find that models trained with one solver can degrade severely when evaluated with another, even when weights, architecture, and data are held fixed. This indicates that the solver is not acting as a fully interchangeable post hoc computational component under the tested setup. To examine a plausible explanation for this effect, we analyze how the stability polynomial of explicit Runge–Kutta methods interacts with the learned dynamics. In classical numerical analysis, the stability region determines how linear modes are amplified or damped Hairer et al. (1993). Modern extended-stability Runge–Kutta methods construct these polynomials using orthogonal families such as Chebyshev or Gegenbauer polynomials to enlarge the stability interval of the real axis Abdulle (2002); O'Sullivan (2019;

2015). We interpret this structure as a plausible source of solver-dependent bias in representation dynamics during training.

We introduce a third-order Neural ODE formulation that makes feature velocity and acceleration explicit, enabling direct measurement of higher-order trajectory dynamics without requiring nested differentiation through the solver. This allows us to treat representation geometry as the primary object of analysis, rather than inferring solver effects only from predictive accuracy. Across CIFAR-10 and CIFAR-100, multiple integration horizons, and tightly matched clean accuracy, we find that solvers with different stability polynomials are associated with systematically different geometric regimes during training. We then use adversarial robustness under iterative attacks as a downstream diagnostic of these learned representations. The resulting association is regime-dependent: at moderate integration horizons, higher-order trajectory-norm quantities and related geometric measures are positively associated with iterative robustness, whereas at extended horizons, behavior is increasingly shaped by the interaction between the learned Jacobian spectrum and the solver's stability envelope.

We also examine the role of solver realization. Although algebraically equivalent Runge–Kutta realizations share the same stability polynomial under linear analysis, they can produce distinct geometric dynamics during training despite comparable predictive performance. This realization-level comparison is a central part of the empirical design: it shows that even when two Runge–Kutta implementations share the same order, stage count, NFE, and external stability polynomial, they can still induce different learned dynamics under the same training setup. This suggests that solver behavior may depend not only on the global structure of the stability polynomial, but also on the realization-level numerical effects.

Overall, these results support the view that numerical integration is not fully reducible to an implementation detail in continuous-depth models. The solver-swap experiment provides intervention-based evidence that solver replacement is not functionally neutral after training under the tested setup, while the remaining analyses identify consistent but non-causal associations between solver-induced geometry and robustness.

- We introduce a frozen-weight solver-swap intervention in which models trained with one integrator are evaluated using another without changing parameters. This replacement can produce large degradation in predictive performance, providing intervention-based evidence that trained Neural ODE checkpoints are not functionally invariant to post hoc solver replacement under the tested setup.
- We use a third-order Neural ODE formulation to measure higher-order trajectory-norm dynamics directly, without reconstructing these quantities through nested differentiation through the solver.
- We provide evidence that different explicit Runge–Kutta stability polynomials are associated with systematically different learned geometric regimes under matched architecture, optimization, and clean predictive accuracy.
- We use iterative adversarial robustness as a downstream diagnostic of the learned representations. The observed geometry–robustness association is regime-dependent, appearing strongest in moderate operating regimes and weakening or changing when learned dynamics approach the boundary of the solver stability region.
- We isolate realization-level effects by comparing ESRKG-15 with a $2S$–1 low-storage realization that shares the same order, stage count, NFE, and external stability polynomial. Despite this external equivalence, the two realizations are associated with distinct learned dynamics, suggesting that solver effects can extend beyond the stability polynomial to internal stage propagation.

2 Related Work

Our work connects Neural ODEs, Runge–Kutta stability theory, representation geometry, and adversarial robustness. Rather than surveying each area exhaustively, we focus on the specific gaps most relevant to our study.

2.1 Neural ODEs and Solver Choice

Neural Ordinary Differential Equations (Neural ODEs) introduced a continuous-depth formulation of residual networks Chen et al. (2018). Subsequent work has emphasized memory-efficient adjoint methods Kidger (2022), adaptive time-stepping Zhuang et al. (2020), stiff-aware training Massaroli et al. (2020), and dynamical-systems interpretations of deep networks Haber & Ruthotto (2018); Ruthotto & Haber (2020). In most prior work, the numerical solver is treated primarily as a computational mechanism for approximating trajectories, with solver comparisons motivated by efficiency, memory usage, or stiffness. Much less attention has been given to whether the algebraic structure of the integrator itself acts as a structural inductive bias that shapes learned representation dynamics. Our work studies the solver from this perspective.

2.2 Runge–Kutta Stability, Extended-Stability Methods, and Realization Effects

Classical Runge–Kutta theory characterizes explicit methods through their order conditions and stability polynomial $R(z)$, which determines the region of absolute stability Hairer et al. (1993); Butcher (2016). Methods sharing the same stability polynomial are equivalent on linear test equations, but this equivalence does not by itself determine their behavior in nonlinear learning systems, where internal stage structure, floating-point accumulation, and repeated Jacobian compositions can affect training dynamics. Extended-stability Runge–Kutta methods enlarge the negative real-axis stability interval using Chebyshev- or Gegenbauer-based constructions Abdulle (2002); O'Sullivan (2015; 2019). In particular, the Runge–Kutta–Gegenbauer framework introduces a parameter that changes the shape of the stability region while preserving order and stage structure. Low-storage realizations preserve the same external stability polynomial while reorganizing intermediate stage computations Williamson (1980); Carpenter & Kennedy (1994). These properties make ESRK methods a useful setting for isolating the roles of stability-envelope geometry and realization-level numerical effects.

2.3 Representation Geometry and Adversarial Robustness

A growing body of work studies geometric properties of deep networks, including sharpness, Lipschitz behavior, and Jacobian regularization Finlay et al. (2020); Béthune (2024); Moosavi-Dezfooli et al. (2019). Other work interprets residual networks and Neural ODEs through dynamical-systems perspectives Massaroli et al. (2020); Ruthotto & Haber (2020). Adversarial robustness research has focused primarily on training objectives and regularization strategies Goodfellow et al. (2014); Madry et al. (2017); Croce & Hein (2020). Connections between robustness and local geometric quantities have been proposed, but the numerical integration scheme is typically held fixed. In contrast, we isolate the integrator as the varying component while controlling architecture, optimization, and clean accuracy, and studying how solver-dependent geometry is associated with robustness across datasets and integration regimes.

3 Methodology

We describe the continuous-depth model used in our experiments and motivate the use of a third-order Neural ODE formulation that enables stable and solver-consistent analysis of higher-order representation dynamics.

Continuous-Depth Model We consider continuous-depth neural networks formulated as Neural Ordinary Differential Equations (Neural ODEs). Given an initial representation $z(0) \in \mathbb{R}^d$, the latent state evolves according to

$$\dot{z}(t) = r_\theta(z(t)), \quad t \in [0, \Delta T], \tag{1}$$

where $r_\theta : \mathbb{R}^d \to \mathbb{R}^d$ is a neural network parameterized by θ. The terminal state $z(\Delta T)$ is passed to a linear classifier to produce class logits.

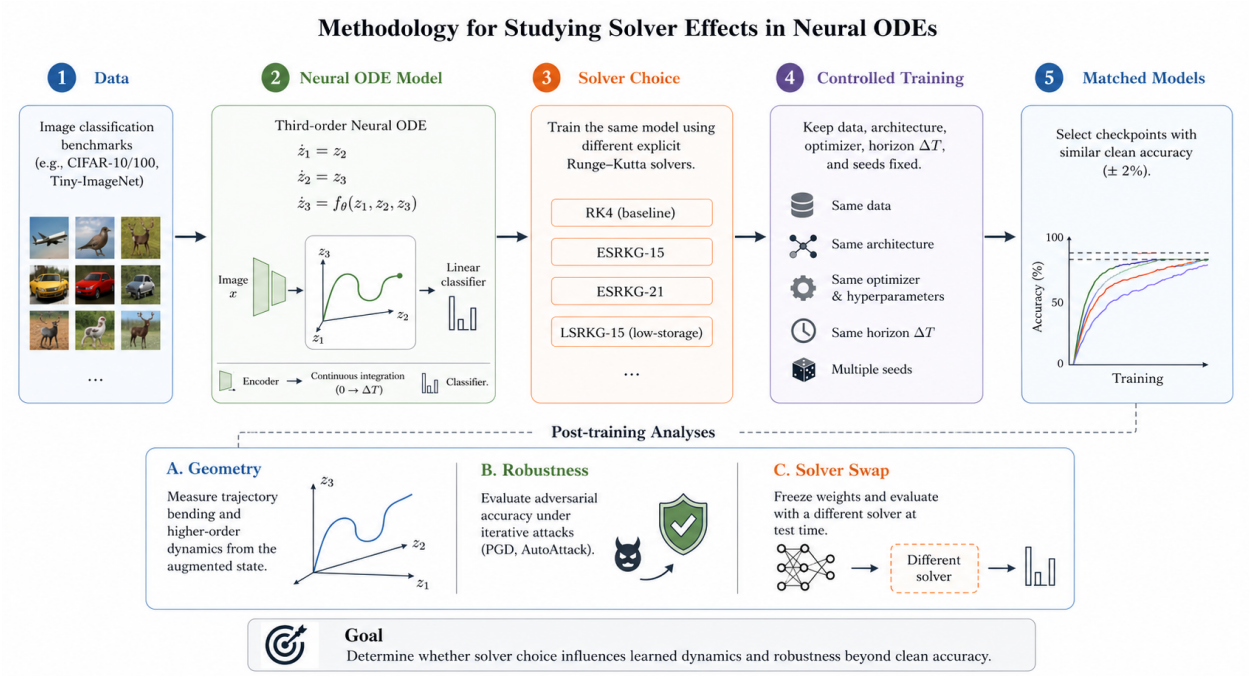


Figure 1: Overview of the experimental methodology used to study the effect of numerical solvers in Neural ODEs. A shared model architecture is trained using different Runge–Kutta integrators under controlled conditions, followed by post-training analyses of higher-order representation dynamics, adversarial robustness, and solver-swap interventions. The goal is to isolate whether solver choice influences learned dynamics beyond predictive accuracy.

To expose higher-order representational dynamics, we extend the first-order formulation to a third-order Neural ODE by augmenting the state:

$$\dot{z}_1(t) = z_2(t), \tag{2}$$
$$\dot{z}_2(t) = z_3(t), \tag{3}$$
$$\dot{z}_3(t) = f_\theta(z_1(t), z_2(t), z_3(t)), \tag{4}$$

where $z_1(t) \in \mathbb{R}^d$ represents latent features, $z_2(t) = \dot{z}_1(t)$ their velocity, and $z_3(t) = \ddot{z}_1(t)$ their acceleration. The neural network f_θ parameterizes the third derivative, or jerk, of the latent trajectory.

Embedding velocity and acceleration directly in the system state, while parameterizing the third derivative through f_θ, allows higher-order representation dynamics to be measured during training without requiring nested differentiation through the numerical solver.

Geometric and Higher-Order Dynamic Quantities The third-order embedding enables direct measurement of raw jerk and a velocity-normalized higher-order trajectory norm proxy. Since the augmented state satisfies

$$\dot{z}_1(t) = z_2(t), \qquad \dot{z}_2(t) = z_3(t), \qquad \dot{z}_3(t) = f_\theta(z_1(t), z_2(t), z_3(t)),$$

we compute

$$\kappa_3(t) = \frac{\|f_\theta(z_1(t), z_2(t), z_3(t))\|}{\|z_2(t)\| + \epsilon}, \qquad j(t) = \|\dot{z}_3(t)\| = \|f_\theta(z_1(t), z_2(t), z_3(t))\|. \tag{5}$$

Here $j(t)$ is the raw third-order forcing, or jerk magnitude, while $\kappa_3(t)$ measures the magnitude of this third-order response relative to the instantaneous representation velocity. We refer to κ_3 as a *higher-order trajectory norm proxy*: it captures a velocity-normalized third-order trajectory response rather than an intrinsic curvature measure or classical acceleration-based geometric higher-order trajectory norm. We use $\epsilon = 10^{-12}$, which is negligible relative to observed velocity magnitudes and is included only for numerical stability.

The purpose of the third-order formulation is not to increase model expressivity, but to expose velocity, acceleration, and jerk-related quantities directly in the state dynamics. In first- and second-order Neural ODEs, such higher-order quantities would need to be reconstructed through repeated differentiation of the learned vector field, which can couple the estimates to solver discretization and numerical differentiation artifacts. By embedding velocity and acceleration in the state and using f_θ as the third-order forcing term, we obtain solver-consistent measurements of higher-order representation dynamics across integrators.

Adversarial Robustness Evaluation To assess whether higher-order dynamic differences across solvers correspond to differences in adversarial behavior, we evaluate adversarial accuracy under standard ℓ_∞-bounded attacks. After training, model parameters are fixed, and robustness is evaluated on the test set without further optimization. We consider perturbations bounded by $\varepsilon = 8/255$ under the ℓ_∞ norm.

Robustness is evaluated using three complementary attacks:

- **FGSM**: one-step Fast Gradient Sign Method with $\varepsilon = 8/255$.
- **PGD**: Projected Gradient Descent with $\varepsilon = 8/255$, step size $\alpha = 2/255$, and 10 steps.
- **AutoAttack**: a parameter-free ensemble attack combining APGD-CE, APGD-DLR, FAB, and Square Attack.

Adversarial examples are generated with gradients enabled, and accuracy is computed without gradient tracking. Clean accuracy is controlled within 2% across solvers so that robustness differences can be interpreted relative to matched predictive performance. FGSM reflects the local gradient structure due to its single-step nature and is therefore treated as a secondary diagnostic, while our main robustness conclusions rely on multi-step attacks such as PGD and AutoAttack, which probe iterative decision-boundary geometry more directly.

3.1 Stability Polynomial and Higher-Order trajectory norm Amplification

We analyze how the stability polynomial of an explicit Runge–Kutta method interacts with higher-order dynamic quantities in the third-order Neural ODE formulation, and motivate a qualitative prediction for how solver-dependent behavior varies across training regimes.

Linearized Dynamics Consider the local linearization of the learned dynamics:

$$\dot{z}(t) = Jz(t),$$

where J is the Jacobian evaluated along the trajectory. An explicit Runge–Kutta method advances the solution as

$$z_{n+1} = R(hJ)z_n.$$

Because $R(hJ)$ is a matrix polynomial, it can be written as

$$R(hJ) = \sum_{k=0}^{s} a_k (hJ)^k,$$

where s is the number of stages. For a method of order p, the coefficients through degree p agree with the Taylor expansion of $\exp(hJ)$, while the remaining higher-degree coefficients determine the stability-polynomial structure. Thus, time stepping induces structured amplification of Jacobian powers weighted by the coefficients of the stability polynomial.

Operating Fraction To characterize the effective stability regime, we define the operating fraction

$$\phi = \frac{h\sigma}{R},$$

where σ is the spectral radius of the learned Jacobian, estimated using JVP-based power iteration, and R is the real-axis stability radius of the integrator. Intuitively, ϕ measures how close the learned Jacobian spectrum lies to the boundary of the solver's stability region.

Regime Prediction This formulation yields a qualitative prediction for solver-dependent behavior:

- **Low operating fraction ($\phi \ll 1$):** The dynamics lie deep within the stability region, and the stability polynomial acts as a smooth approximation to the exponential. In this regime, amplification is governed primarily by low-order Jacobian terms, and higher-order quantities such as raw jerk and the higher-order trajectory norm proxy κ_3 provide stable descriptors of the learned dynamics. We therefore expect these dynamic measures to be associated with adversarial robustness.
- **High operating fraction ($\phi \approx 1$ or larger):** The dynamics approach or exceed the stability boundary, and the behavior is increasingly influenced by the global geometry of the stability envelope rather than by local polynomial approximation. In this regime, higher-order amplification becomes non-uniform across the spectrum, and local dynamic proxies such as κ_3 may no longer serve as reliable standalone descriptors of robustness. Instead, behavior is increasingly shaped by interaction with the stability boundary.

Importantly, this prediction is expected to hold most directly within a shared operating regime. When different solvers converge to substantially different operating fractions, comparisons become cross-regime rather than within-regime. In that setting, raw jerk and the higher-order trajectory norm proxy need not induce a monotone solver-level ordering of robustness, because behavior is increasingly shaped by interaction with the solver stability envelope.

The framework is necessarily approximate, since the Jacobian varies along the trajectory and the spectral radius provides only a coarse summary of the eigenvalue distribution. It should therefore be interpreted as a qualitative regime description rather than a precise predictive model. Nevertheless, it is falsifiable: it suggests that (i) in low-ϕ regimes, higher-order dynamic quantities are expected to be associated with improved robustness, and (ii) in high-ϕ regimes, this relationship may weaken or invert. These expectations are then evaluated empirically in the experiments that follow.

Higher-order trajectory norm proxy In the third-order formulation,

$$z_2 = \dot{z}_1, \quad z_3 = \ddot{z}_1, \quad \dot{z}_3 = f_\theta(z_1, z_2, z_3).$$

The reported higher order trajectory norm proxy is

$$\kappa_3 = \frac{\|\dot{z}_3\|}{\|z_2\| + \epsilon} = \frac{\|f_\theta(z_1, z_2, z_3)\|}{\|z_2\| + \epsilon}.$$

Under a local linearization of the learned feature dynamics, one may heuristically write

$$z_2 \approx J z_1, \qquad z_3 \approx J^2 z_1, \qquad \dot{z}_3 \approx J^3 z_1.$$

Thus, the higher-order proxy can be interpreted approximately as

$$\kappa_3 \approx \frac{\|J^3 z_1\|}{\|J z_1\| + \epsilon}.$$

After a Runge–Kutta update,

$$z_1^{(n+1)} = R(hJ) z_1^{(n)},$$

this gives the qualitative relation

$$\kappa_{3,n+1} \approx \frac{\|J^3 R(hJ) z_1^{(n)}\|}{\|JR(hJ) z_1^{(n)}\| + \epsilon}.$$

This expression should not be read as an exact identity for the nonlinear network. Rather, it provides a local spectral interpretation of why the measured higher-order trajectory norm proxy can depend on the numerical solver. Different Runge–Kutta methods apply different polynomial filters to the local Jacobian spectrum through $R(hJ)$, reweighting modes according to $R(h\lambda)$ during integration. Since the reported proxy depends on a third-order response normalized by velocity, these solver-dependent spectral weights provide a plausible mechanism through which the integrator can influence higher-order representation dynamics and robustness.

In the low-ϕ regime, this amplification is relatively smooth and well described by the low-order approximation of the exponential. In the high-ϕ regime, it becomes increasingly shaped by the global stability envelope. This argument is local and approximate, and captures stability-polynomial effects; solvers with the same external stability polynomial may still differ through internal stage realization and numerical amplification.

4 Experiments

This section presents a systematic empirical evaluation of solver-induced geometric and robustness effects in continuous-depth networks. We first describe the experimental setup, including architecture, optimization, integration configurations, and datasets. We then report clean and adversarial performance across integration horizons, followed by an analysis of representation geometry, spectral behavior, and gradient-flow dynamics. All comparisons are conducted under fixed integration horizons to reduce confounding from horizon choice and to study the influence of solver stability geometry. In addition to comparing distinct solver families, we perform a controlled perturbation of the stability polynomial by varying the Gegenbauer parameter while keeping method order and stage count fixed. This allows us to test whether systematic changes in stability-envelope geometry are associated with corresponding changes in representation geometry and robustness.

4.1 Datasets

We evaluate our proposed method on three standard image classification benchmarks, selected for their varying degrees of scale and complexity.

CIFAR-10 comprises 60,000 32×32 color images distributed across 10 balanced classes, with 50,000 training and 10,000 test images. It serves as a fundamental baseline for assessing generalization and robustness in low-resolution settings Krizhevsky et al. (2009).

CIFAR-100 maintains the same resolution and total volume as CIFAR-10 but introduces 100 classes. The resulting increase in label density offers a more fine-grained and challenging environment to assess model performance under high class complexity Krizhevsky et al. (2009).

Tiny-ImageNet is a subset of the ImageNet dataset containing 200 classes with images resized to 64×64. It functions as an intermediate-scale benchmark, bridging the gap between small-scale CIFAR datasets and high-resolution vision tasks through increased visual diversity and task difficulty Le et al. (2015).

All benchmarks utilize their standard training and testing partitions. To prevent overfitting and ensure consistency with prior work, we employ standard data augmentation techniques, specifically, random cropping and horizontal flipping during the training phase.

4.2 Experimental Setup

All experiments use the same third-order Neural ODE architecture implemented in PyTorch. Model depth, width, initialization, and optimization settings are fixed across solver configurations to isolate the effect of the numerical integrator under matched training conditions. Experiments are conducted on CIFAR-10 and CIFAR-100 with the continuous integration horizon fixed to $\Delta T \in \{10, 20, 30\}$. Tiny-ImageNet is used as an

external validation benchmark at $\Delta T = 20$. Standard training protocols described in Section 4.1 are used throughout.

We compare RK4, ESRKG-15, ESRKG-21, and a $2S$–1 low-storage realization of ESRKG-15 (LSRKG-15). ESRKG-15 and ESRKG-21 integrate the entire horizon in a single step ($h = \Delta T$), reflecting their enlarged real-axis stability intervals. RK4 cannot stably use $h = \Delta T$ at these horizons and therefore integrates the same continuous interval using four equal substeps: $h = 2.5$ for $\Delta T = 10$, $h = 5$ for $\Delta T = 20$, and $h = 7.5$ for $\Delta T = 30$. This difference is intentional: rather than enforcing identical step counts, we study each solver in its natural operating regime while keeping the continuous horizon fixed.

ESRKG-21 is included not as a compute-matched baseline, but as a controlled variation in stability-envelope geometry. Its larger stage count yields a substantially larger stability region and therefore probes a different operating regime from ESRKG-15. Compute-matched isolation of solver effects is addressed separately through the Gegenbauer sweep and the ESRKG-15/LSRKG-15 realization comparison. The ESRKG-15, ESRKG-21, and LSRKG-15 configurations all use the same Gegenbauer parameter $\nu = \frac{1}{4}$, so differences among them reflect stage count and/or realization rather than a change in the polynomial-family parameter itself.

Differences in step count and the Number of Function Evaluations (NFE) are treated as inherent properties of the methods rather than tunable hyperparameters. RK4 performs $4 \times 4 = 16$ function evaluations per forward pass, while ESRKG-15 and ESRKG-21 perform 15 and 21, respectively. Because RK4 and ESRKG-15 have nearly identical NFE (16 versus 15), differences between these methods are unlikely to be explained solely by evaluation count, and are more consistent with structural differences in the integrators.

To separate stability-polynomial structure from implementation realization, we also compare ESRKG-15 with LSRKG-15. These methods share the same order, stage count, NFE, and stability polynomial, but differ in the realization of the coefficients and intermediate-stage accumulation. As a result, they can exhibit different internal amplification, which provides a plausible realization-level mechanism for distinct geometric and robustness behaviour despite external linear equivalence.

Taken together, the experimental design provides three complementary controls: (i) near compute-matched comparisons such as RK4 versus ESRKG-15, where NFE is similar but the stability geometry differs; (ii) realization-matched comparisons such as ESRKG-15 versus LSRKG-15, where stage count, NFE, order, and stability polynomial are identical; and (iii) regime-probing comparisons such as ESRKG-21, which intentionally varies the stage count to alter the stability envelope. The Gegenbauer sweep provides an additional strictly computed-matched intervention in which stage count and NFE are fixed, while only the stability polynomial is varied. All solvers are implemented using the same generic Runge–Kutta update loop, so differences cannot be attributed to extraneous codepath variation. All models are trained with AdamW using the learning rate 3×10^{-4}, weight decay 5×10^{-4}, cosine learning rate scheduling, label smoothing 0.05, and gradient clipping with norm 5. All models are trained using direct backpropagation through the unrolled Runge Kutta solver steps; adjoint sensitivity methods are not used. Each configuration is evaluated over three independent random seeds. Following prior observations of run-to-run variability on CIFAR-scale benchmarks Banerjee et al. (2025); Qian et al. (2021), we treat solver configurations within $\pm 2\%$ test accuracy as approximately performance matched.

Adversarial robustness is evaluated using FGSM, multi-step PGD, and AutoAttack under the ℓ_∞ threat model with perturbation budget $\epsilon = 8/255$. Because the models are trained without adversarial objectives, absolute robustness levels are modest and should be interpreted comparatively rather than as absolute robustness claims. FGSM is treated as a secondary diagnostic, while our main conclusions focus on iterative attacks (PGD and AutoAttack). Relative gains are reported in Table 12.

Limits of the operating-fraction description. The operating fraction $\phi = h\sigma/R$ provides a coarse description of where the learned dynamics sit relative to the stability envelope. It is therefore useful for comparing solvers with different stability radii or different learned spectral scales. However, ϕ does not uniquely determine solver behavior. When two methods occupy similar operating fractions, realization-level effects can become important. This is particularly relevant for ESRKG-15 and LSRKG-15, which share the same stability polynomial but differ in their internal stage propagation.

4.3 Experiment I: Solver-Dependent Geometry and Robustness Across Horizons

We present a systematic evaluation of solver-induced geometric and robustness effects on CIFAR-10 and CIFAR-100 across multiple integration horizons $\Delta T \in \{10, 20, 30\}$. All experiments use the identical third-order Neural ODE architecture, optimization schedule, data preprocessing, and training hyperparameters described in Section 4.2. Solver configurations differ only in the numerical integration scheme and corresponding step decomposition required to remain within each method's stability region. Clean validation accuracy is tightly controlled (within $\pm 2\%$ across three seeds) to ensure that observed differences in representation dynamics and adversarial robustness cannot be attributed to disparities in predictive performance. For each ΔT, we analyze (i) clean convergence behavior, (ii) higher-order trajectory-norm dynamics and third-order dynamics (jerk), and (iii) adversarial robustness under FGSM, PGD, and AutoAttack. This design allows us to isolate how the stability polynomial of the integrator influences internal representation geometry and whether these geometric effects systematically correspond to changes in robustness under matched accuracy conditions.

4.3.1 CIFAR-10 at $\Delta T = 10$

We evaluate solver-induced effects on CIFAR-10 using an integration horizon $\Delta T = 10$. All solvers are trained under identical optimization settings. Clean validation accuracy is tightly matched (within $\pm 1\%$ across three seeds) to isolate solver-dependent geometric and robustness effects.

Clean Accuracy. Figure 2 (top-left) shows validation accuracy over training. All solvers converge smoothly to comparable final performance (approximately 84–85%), with no statistically significant separation across seeds. This is consistent with the interpretation that differences observed in subsequent geometric and robustness analyses are not attributable to disparities in clean predictive performance.

Representation higher-order trajectory norm. Figure 2 (top-right) reports the evolution of the higher-order trajectory proxy

$$\kappa_3(t) = \frac{\|f_\theta(z_1(t), z_2(t), z_3(t))\|}{\|z_2(t)\| + \epsilon}.$$

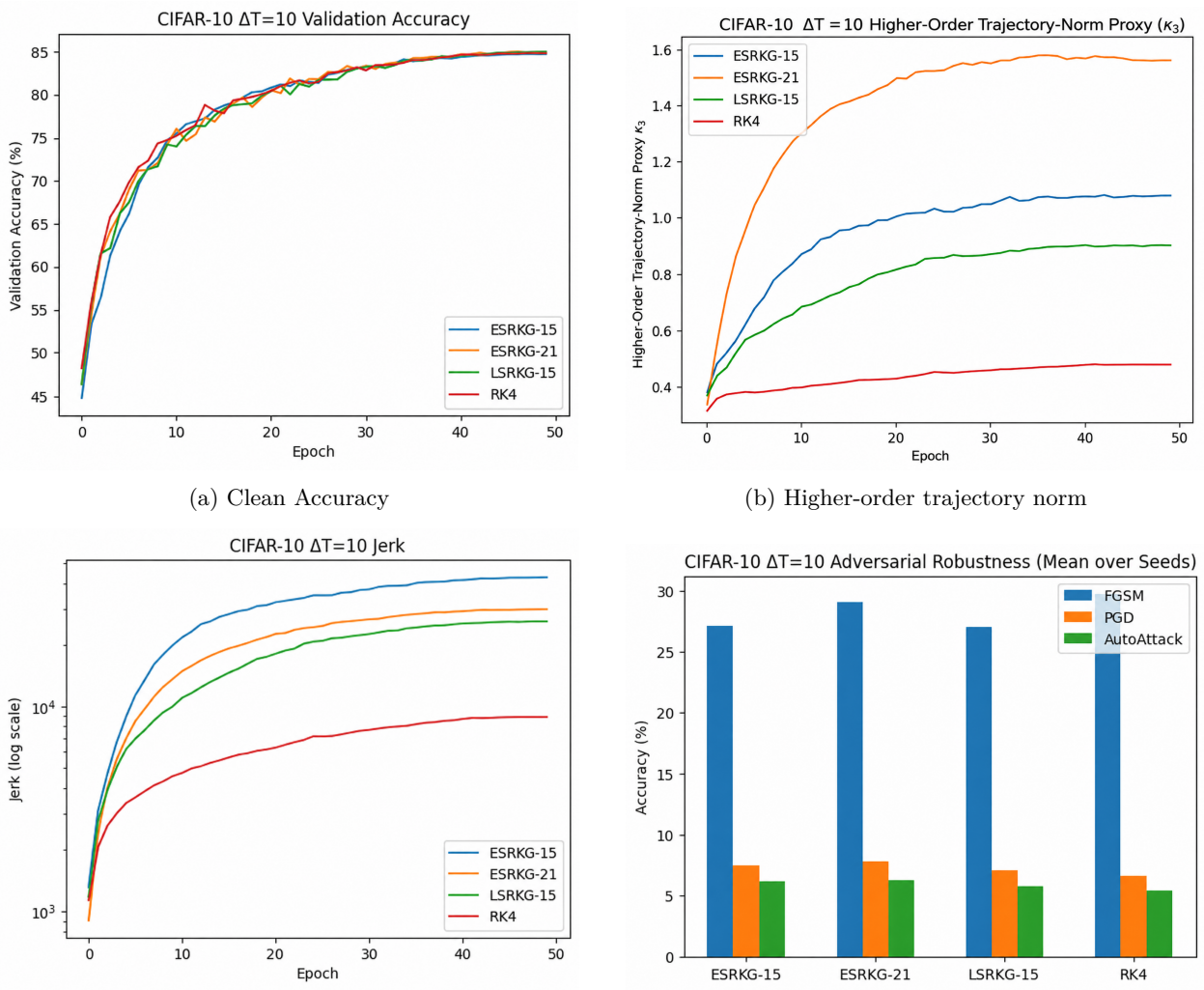
For readability, we occasionally refer to this quantity as a higher-order trajectory in the empirical discussion. However, it should be interpreted as a velocity-normalized third-order trajectory norm proxy rather than as an intrinsic curvature or acceleration-based geometric interpretation. A clear solver-dependent ordering emerges

$$\text{RK4} < \text{LSRKG-15} < \text{ESRKG-15} < \text{ESRKG-21}.$$

RK4 produces the smallest higher-order trajectory norm proxy, whereas ESRKG-21 yields the largest value. Notably, LSRKG-15 and ESRKG-15 share the same stability polynomial yet exhibit distinct higher-order trajectories, indicating that stage realization influences higher-order representation dynamics even when external stability regions coincide.

Third-Order Dynamics (Jerk). Figure 2 (bottom-left, log scale) shows the magnitude of the third derivative. Extended-stability solvers accumulate substantially larger jerk magnitudes than RK4. However, ESRKG-15 exhibits slightly higher jerk than ESRKG-21 despite marginally lower robustness (see below), suggesting that robustness does not depend solely on jerk magnitude but rather reflects structured amplification behavior associated with the stability polynomial. While jerk magnitudes differ substantially across solvers, robustness does not appear to be determined by jerk alone. Rather, jerk and higher-order trajectory norm jointly reflect the higher-order structure of the learned dynamics, with higher-order trajectory norm exhibiting a more stable association with iterative robustness in moderate-horizon regimes.

The divergence between jerk and the higher-order norm proxies indicates that the two quantities capture complementary aspects of the learned dynamics. Since $\kappa_3 = \|f_\theta(z_1, z_2, z_3)\|/(\|z_2\| + \epsilon)$, differences between raw jerk and higher-order trajectory norms can arise from changes in the representation velocity $\|z_2\|$, not only from changes in third-order forcing magnitude. Thus, a solver may exhibit larger higher-order trajectory norms either by increasing higher-order forcing or by evolving through a slower representation-velocity regime.



(a) Clean Accuracy

(b) Higher-order trajectory norm

(c) Jerk (log scale)

(d) Adversarial Robustness

Figure 2: Solver-dependent training dynamics and adversarial robustness on CIFAR-10 at $\Delta T = 10$. (a) Validation accuracy remains closely matched across solvers, indicating that clean predictive performance is controlled. (b) The higher-order trajectory proxy $\kappa_3 = \|f_\theta(z_1, z_2, z_3)\|/(\|z_2\| + \epsilon)$ shows solver-dependent separation in learned trajectory dynamics. For readability, we refer to this quantity as a higher order trajectory norm. (c) Jerk magnitude, shown on a logarithmic scale, measures $\|f_\theta(z_1, z_2, z_3)\|$ and indicates stronger higher-order trajectory dynamics for the extended-stability and low-storage solvers than for RK4. (d) Adversarial accuracy under FGSM, PGD, and AutoAttack differs across solvers despite matched clean accuracy, with iterative attacks used as the main robustness comparison.

Adversarial Robustness. Figure 2 (bottom-right) summarizes adversarial robustness under FGSM, PGD, and AutoAttack. A consistent solver ordering is observed across iterative attacks:

$$\text{ESRKG-21} > \text{ESRKG-15} > \text{LSRKG-15} > \text{RK4}.$$

Although absolute differences are moderate in magnitude, the ordering is stable across seeds and across both PGD and AutoAttack. Importantly, these robustness differences persist despite tightly matched clean accuracy (within $\approx 0.3\%$ across solvers), indicating that adversarial behavior is not an artefact of predictive performance disparities.

Table 1: Three-seed summary on CIFAR-10 at $\Delta T = 10$. Values report mean $\pm$ standard deviation across three random seeds. All solvers achieve comparable clean accuracy while exhibiting distinct geometric and robustness characteristics. Bold indicates the best performing solver.

Solver	Val Acc (Best)	Val Acc (Final)	higher-order trajectory norms	Jerk	FGSM	PGD	AutoAttack	Time / Epoch (s)
ESRKG-15	84.80 ± 0.25	84.72 ± 0.32	1.08 ± 0.12	42840 ± 4242	27.08 ± 1.54	7.49 ± 0.56	6.19 ± 0.54	150.76 ± 12.37
ESRKG-21	$\mathbf{85.09 \pm 0.30}$	$\mathbf{84.95 \pm 0.31}$	$\mathbf{1.56 \pm 0.15}$	$\mathbf{29920 \pm 3366}$	$\mathbf{29.01 \pm 1.14}$	$\mathbf{7.82 \pm 0.30}$	$\mathbf{6.27 \pm 0.37}$	$\mathbf{247.04 \pm 12.25}$
LSRKG-15	85.02 ± 0.13	84.96 ± 0.15	0.90 ± 0.10	26100 ± 3269	27.02 ± 1.38	7.09 ± 0.44	5.74 ± 0.35	154.76 ± 5.01
RK4	84.90 ± 0.27	84.80 ± 0.23	0.48 ± 0.02	8880 ± 757	29.69 ± 0.99	6.64 ± 0.34	5.45 ± 0.22	79.40 ± 1.40

Table 2: Three-seed summary on CIFAR-10 at $\Delta T = 20$. Values report mean $\pm$ standard deviation across three random seeds.

Solver	Clean (%)	higher-order trajectory norms	Jerk	FGSM (%)	PGD (%)	AutoAttack (%)
RK4	84.73 ± 0.11	0.296 ± 0.004	$(1.40 \pm 0.03) \times 10^4$	25.6 ± 0.5	3.52 ± 0.09	2.80 ± 0.08
LSRKG-15	84.61 ± 0.07	0.454 ± 0.006	$(2.01 \pm 0.04) \times 10^4$	26.1 ± 0.4	3.63 ± 0.09	3.01 ± 0.07
ESRKG-21	$\mathbf{84.78 \pm 0.07}$	$\mathbf{0.737 \pm 0.001}$	$\mathbf{(2.30 \pm 0.03) \times 10^4}$	$\mathbf{25.12 \pm 0.4}$	$\mathbf{5.75 \pm 0.08}$	$\mathbf{4.47 \pm 0.07}$
ESRKG-15	83.95 ± 0.25	1.11 ± 0.08	$(7.73 \pm 0.59) \times 10^4$	24.12 ± 1.76	4.27 ± 0.15	3.34 ± 0.11

4.3.2 CIFAR-10 at $\Delta T = 20$

Next, we examine solver-induced behavior at the extended integration horizon $\Delta T = 20$. As in the $\Delta T = 10$ setting, clean validation accuracy remains tightly matched across solvers (83.95–84.78%), ensuring that differences in geometric and adversarial metrics are not attributable to disparities in predictive performance.

Geometric structure. Final representation higher-order trajectory norms exhibits clear separation across solver families:

$$\text{RK4} < \text{LSRKG-15} < \text{ESRKG-21} < \text{ESRKG-15}.$$

Classical RK4 produces the lowest higher-order trajectory norm (0.296), while ESRKG-15 yields the largest higher-order trajectory norm magnitude (1.11). The low-storage realization remains geometrically distinct from its classical counterpart despite sharing the same stability polynomial, and ESRKG-21 occupies an intermediate regime (0.737). Jerk magnitudes amplify this separation further, ranging from 1.40×10^4 (RK4) to 7.73×10^4 (ESRKG-15), indicating substantial solver-dependent higher-order trajectory effects.

Although LSRKG-15 exhibits substantially lower higher-order trajectory norm than ESRKG-15, it does not exhibit a corresponding advantage in iterative robustness: its PGD robustness (3.63%) remains close to RK4 (3.52%) and well below ESRKG-15 (4.27%). This comparison is notable because ESRKG-15 and LSRKG-15 share the same stability polynomial, yet converge to different effective operating fractions during training. The observed gap is therefore most consistent with a combination of operating-regime shift and realization-level effects. Consistent with the latter, the two realizations remain sharply separated under the diagnostic $\|(I - zA)^{-1}\|_\infty$ at the experimentally observed operating points. For CIFAR-10 at $\Delta T = 20$, these values are approximately 6.58×10^5 for ESRKG-15 and 2.02×10^4 for LSRKG-15, indicating substantially different stage-propagation behaviour despite external polynomial equivalence; see Section 4.4.2 and Figure 5.

Adversarial robustness. Under strong iterative attacks, robustness displays structured variation. PGD robustness increases from RK4 (3.52%) and LSRKG-15 (3.63%) to ESRKG-15 (4.27%), and is highest for ESRKG-21 (5.75%). AutoAttack follows the same qualitative ordering, while FGSM shows weaker separation between solver families.

Interestingly, ESRKG-21 achieves higher PGD robustness than ESRKG-15 despite exhibiting lower higher-order trajectory norm magnitude. Rather than contradicting the operating-fraction framework, this difference is consistent with the fact that the two solvers converge to substantially different spectral radii on the same dataset. ESRKG-21 consistently settles to $\sigma \approx 0.77$ across seeds, while ESRKG-15 converges to $\sigma \approx 2.75$. Because the effective stability argument is $h\sigma$, the two solvers occupy qualitatively distinct operating regimes: ESRKG-15 lies in the outer-envelope regime, where higher-order polynomial structure more strongly influences amplification, whereas ESRKG-21 operates deep in the stability interior, where the

Table 3: Estimated operating fraction $\phi = \frac{h\sigma}{R}$ for CIFAR-10 at $\Delta T = 20$. ESRKG methods integrate the full horizon in a single stabilised step ($h = 20$), while RK4 uses four steps of size $h = 5$.

Solver	Steps	h	σ	$h\sigma$	ϕ
ESRKG-15	1	20	≈ 2.75	55.0	0.76
ESRKG-21	1	20	≈ 0.77	15.4	0.11
LSRKG-15	1	20	≈ 2.35	47.0	0.65
RK4	4	5	≈ 0.87	4.35	1.56

Table 4: Summary on CIFAR-10 at $\Delta T = 30$. Values report mean $\pm$ standard deviation across three seeds.

Solver	Clean (%)	higher-order trajectory norm	Jerk	FGSM (%)	PGD (%)	AutoAttack (%)
RK4	83.53 ± 0.11	0.206 ± 0.004	$(1.40 \pm 0.02) \times 10^4$	25.6 ± 0.5	3.02 ± 0.09	1.95 ± 0.08
LSRKG-15	84.50 ± 0.50	0.655 ± 0.109	$(7.36 \pm 1.63) \times 10^4$	23.96 ± 0.53	2.28 ± 0.24	1.77 ± 0.25
ESRKG-21	82.88 ± 0.07	0.477 ± 0.001	$(1.82 \pm 0.02) \times 10^4$	25.12 ± 0.40	4.09 ± 0.08	3.64 ± 0.07
ESRKG-15	$\mathbf{84.52 \pm 0.06}$	$\mathbf{0.654 \pm 0.001}$	$(\mathbf{6.85 \pm 0.08}) \times \mathbf{10^4}$	$\mathbf{27.4 \pm 0.5}$	$\mathbf{3.37 \pm 0.06}$	$\mathbf{2.98 \pm 0.06}$

polynomial behaves more like a low-order approximation to the exponential. The robustness advantage of ESRKG-21 is therefore more consistent with this regime difference than with a simple higher-order trajectory norm-based ordering. Notably, the solver itself appears to contribute to the learned spectral scale: under the same architecture, dataset, and optimiser, different integrators converge to different spectral radii. This is consistent with the broader view that the numerical integrator influences the learned spectral regime and associated dynamics. To make this regime shift explicit, Table 3 reports the estimated operating fraction $\phi = h\sigma/R$ for each solver.

4.3.3 CIFAR-10 at $\Delta T = 30$

The $\Delta T = 30$ setting tests the regime-transition prediction from Section 3.1. Unlike the moderate-horizon cases, the solvers no longer occupy a common operating regime: ESRKG-21 remains comparatively deep in the stability interior, ESRKG-15 and LSRKG-15 move toward the outer-envelope region, and RK4 operates near its classical stability boundary. Therefore, the higher-order trajectory norm–robustness relationship should be interpreted as a within-regime association, not as a solver-independent monotone ordering across regimes.

Table 4 reports mean $\pm$ standard deviation across three seeds for each solver. As the integration horizon increases, the effective argument of the stability polynomial $h\sigma$ approaches the boundary of the stability region. In this regime, solver behavior is governed less by the local Taylor approximation of the exponential and more by the global geometry of the stability envelope. Consequently, geometric statistics such as higher-order trajectory norm no longer provide a sufficient summary of the dynamics, and robustness becomes increasingly influenced by directional amplification effects induced by the stability polynomial.

Notably, the learned spectral radius σ is not determined solely by the dataset or architecture but also depends on the numerical integrator used during training. While σ ultimately emerges from the interaction between dataset, architecture, optimisation dynamics, and solver discretisation, the integrator influences the spectral regime in which training stabilises. This is reflected in the systematic differences in σ observed across solvers under otherwise identical training configurations.

Clean validation accuracy remains tightly controlled across solvers (82.88–84.52%), with seed-level standard deviations at or below approximately 0.5%, confirming that robustness differences are not driven by predictive performance gaps.

Geometric structure At this extended horizon, the geometric ordering differs from the $\Delta T = 20$ regime. While RK4 still exhibits the smallest higher-order trajectory norm (0.206), the remaining solvers cluster more closely together. Both ESRKG-15 and LSRKG-15 exhibit similar higher-order trajectory norm magnitudes (≈ 0.65), while ESRKG-21 occupies a lower intermediate value (0.477). Jerk magnitudes remain strongly

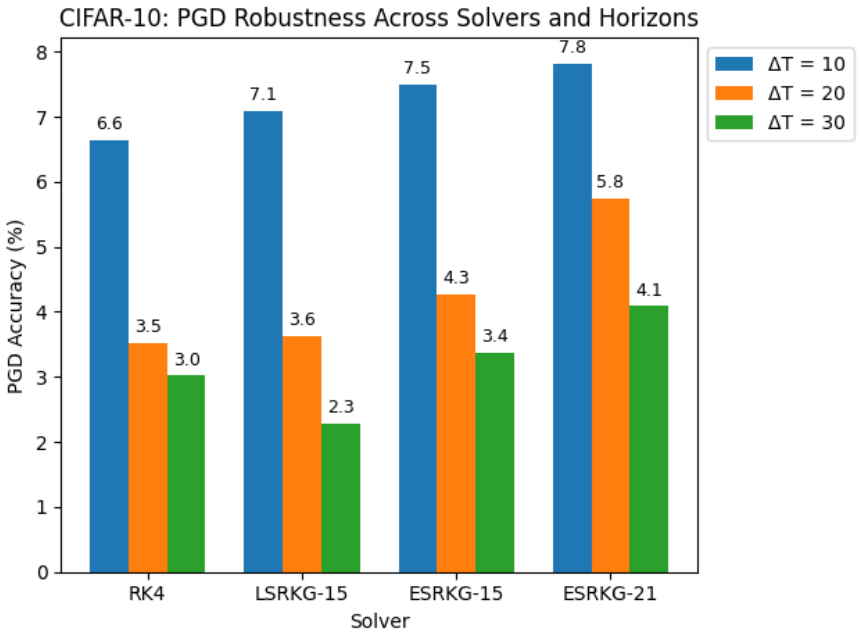


Figure 3: CIFAR-10 robustness across solvers and integration horizons. PGD adversarial accuracy (%) is reported for RK4, LSRKG-15, ESRKG-15, and ESRKG-21 at $\Delta T \in \{10, 20, 30\}$. Clean accuracy is controlled across solvers. Extended-stability methods generally improve iterative robustness relative to RK4, with ESRKG-21 achieving the highest PGD accuracy across the tested horizons.

solver-dependent, ranging from 1.40×10^4 (RK4) to 7.36×10^4 (LSRKG-15), indicating substantially higher-order trajectory variation even when higher-order trajectory norm values partially overlap.

Adversarial robustness Robustness patterns remain structured but differ from the moderate-horizon regime. ESRKG-21 achieves the strongest robustness under iterative attacks (PGD 4.09%, AutoAttack 3.64%), substantially exceeding both RK4 and LSRKG-15, despite exhibiting lower higher-order trajectory norm than ESRKG-15. This is consistent with the observation that robustness is not solely associated with higher-order trajectory norm magnitude. A small deviation from the iterative-attack ordering occurs under FGSM, where ESRKG-15 achieves the highest accuracy. Because FGSM is a secondary diagnostic and is sensitive to local one-step gradient effects, we do not treat this inversion as primary evidence of robustness. Nevertheless, the pattern is consistent with the outer-envelope operating regime of ESRKG-15 at this horizon: amplification of higher-order Jacobian modes may increase local input-gradient magnitude and thereby alter the behavior of a single-step attack, an effect not observed for ESRKG-21, which operates deeper in the stability interior.

Overall, these results indicate a transition in the dominant dynamics. At moderate horizons, higher-order trajectory norm and jerk provide useful empirical descriptors of adversarial robustness, with the strongest seed-level associations appearing in regimes where solvers occupy comparable stability-envelope regions. At extended horizons, however, the interaction between the Jacobian spectrum and the stability-envelope geometry of the integrator becomes increasingly important, weakening the direct higher-order trajectory norm–robustness coupling observed at smaller integration horizons.

Although RK4 operates near or beyond its classical linear stability envelope in this configuration, training remains stable in practice. Optimisation and regularisation mechanisms such as weight decay, gradient clipping, and the AdamW update can reduce the effective spectral radius of the learned Jacobian, helping prevent numerical instability even when the nominal linear stability condition is exceeded.

Table 5: Summary on CIFAR-100 at $\Delta T = 10$. Values report mean $\pm$ standard deviation across three runs.

Solver	Clean (%)	higher-order trajectory norm	Jerk	FGSM (%)	PGD (%)	AutoAttack (%)
RK4	54.97 ± 0.06	0.30 ± 0.01	$(1.38 \pm 0.06) \times 10^4$	9.26 ± 0.21	2.97 ± 0.09	2.30 ± 0.13
LSRKG-15	54.97 ± 0.06	1.01 ± 0.06	$(2.52 \pm 0.23) \times 10^4$	9.53 ± 0.23	3.02 ± 0.09	2.33 ± 0.12
ESRKG-15	54.33 ± 0.26	1.08 ± 0.05	$(3.92 \pm 0.16) \times 10^4$	8.93 ± 0.48	3.03 ± 0.11	2.33 ± 0.13
ESRKG-21	$\mathbf{54.64 \pm 0.22}$	$\mathbf{0.74 \pm 0.03}$	$(\mathbf{2.31 \pm 0.05}) \times \mathbf{10^4}$	$\mathbf{9.11 \pm 0.31}$	$\mathbf{3.08 \pm 0.07}$	$\mathbf{2.39 \pm 0.11}$

4.3.4 CIFAR-100 at $\Delta T = 10$

We first examine solver behavior at the moderate integration horizon $\Delta T = 10$. Table 5 reports the final validation accuracy, geometric statistics, and adversarial robustness for each solver.

Clean accuracy remains tightly matched across solver families (54.33–54.97%), indicating that differences in robustness or geometry cannot be attributed to predictive performance gaps.

Geometric behavior. Clear separation emerges in the geometry induced by different solvers. RK4 produces the lowest higher-order trajectory norm and jerk, while the extended-stability methods generate substantially larger trajectory higher-order trajectory norm. ESRKG-15 yields the highest higher-order trajectory norm and jerk magnitudes, while the low-storage realization also occupies a high-higher-order trajectory norm regime despite sharing the same stability polynomial.

Adversarial robustness. Robustness under iterative attacks exhibits structured variation across solvers. PGD and AutoAttack robustness are lowest for RK4 and increase for the extended-stability schemes, with ESRKG-21 achieving the highest PGD and AutoAttack performance. FGSM differences are smaller, consistent with its single-step nature.

Interestingly, the solver ordering in robustness does not perfectly match the higher-order trajectory norm magnitudes. While ESRKG-15 produces the largest higher order trajectory norm, ESRKG-21 achieves slightly higher PGD robustness. This indicates that robustness is not determined solely by higher-order trajectory norm magnitude, but is influenced by the interaction between the stability polynomial and the training dynamics.

4.3.5 CIFAR-100 at $\Delta T = 20$

We next examine solver behavior on CIFAR-100 at the extended integration horizon $\Delta T = 20$. Table 6 reports final clean accuracy, geometric statistics, and adversarial robustness for each solver.

Clean validation accuracy remains closely matched across methods (53.4–54.1%), ensuring that differences in geometric and adversarial metrics are not attributable to predictive performance gaps.

Table 6: CIFAR-100 results at $\Delta T = 20$.

Solver	Clean (%)	higher-order trajectory norm	Jerk	FGSM (%)	PGD (%)	AutoAttack (%)
RK4	53.41 ± 0.22	0.785 ± 0.003	$6.41 \times 10^3 \pm 2.1 \times 10^2$	8.02 ± 0.41	2.35 ± 0.06	1.61 ± 0.05
LSRKG-15	54.07 ± 0.20	0.328 ± 0.018	$9.19 \times 10^3 \pm 9.03 \times 10^2$	8.75 ± 0.39	1.37 ± 0.17	1.03 ± 0.11
ESRKG-15	53.86 ± 0.21	1.14 ± 0.03	$1.12 \times 10^4 \pm 6.2 \times 10^2$	9.81 ± 0.36	2.72 ± 0.09	2.06 ± 0.07
ESRKG-21	$\mathbf{54.12 \pm 0.19}$	$\mathbf{1.34 \pm 0.04}$	$\mathbf{1.46 \times 10^4 \pm 7.1 \times 10^2}$	$\mathbf{10.22 \pm 0.40}$	$\mathbf{2.96 \pm 0.11}$	$\mathbf{2.22 \pm 0.08}$

One notable exception occurs for the low-storage realization LSRKG-15, which exhibits lower higher-order trajectory norm than RK4 at this horizon. This reversal is consistent with the operating-fraction framework introduced in Section 3.1 Realization-level amplification influences the learned spectral radius σ, and thus the effective stability argument $h\sigma$. When this results in a lower operating fraction ϕ, the method can occupy a lower-higher-order trajectory norm regime than a classical solver operating at a larger fraction. The dataset-specific spectral regime on CIFAR-100 at this horizon is consistent with LSRKG-15 operating in this lower-ϕ regime.

Solver-induced geometry. Representation higher-order trajectory norm differs substantially across solver families. Classical RK4 produces moderate higher-order trajectory norm (0.785), while the Gegenbauer-based schemes (ESRKG-15 and ESRKG-21) generate substantially larger geometric deformation. In contrast, the low-storage realization LSRKG-15, despite sharing the same stability polynomial as ESRKG-15, exhibits markedly reduced higher-order trajectory norm (0.33). This pattern is consistent with realization-level numerical effects interacting with dataset-specific spectral placement, rather than with a stability-polynomial explanation alone.

Jerk magnitudes exhibit a similar solver-dependent pattern, increasing from 6.4×10^3 for RK4 to 1.46×10^4 for ESRKG-21, highlighting structured higher-order trajectory amplification across integrators.

Adversarial robustness. Clear robustness differences emerge under adversarial evaluation. PGD accuracy increases from RK4 (2.35%) and LSRKG-15 (1.37%) to ESRKG-15 (2.72%), with ESRKG-21 achieving the strongest robustness (2.96%). AutoAttack follows a similar trend, while FGSM shows a consistent but less pronounced separation across solvers.

Overall, these results indicate that solver-induced geometric structure is associated with iterative robustness on CIFAR-100 at the level of the broader solver pattern, while important exceptions remain at the realization level. This relationship is therefore solver- and regime-dependent rather than uniform across all extended-stability realizations.

4.3.6 CIFAR-100 at $\Delta T = 30$

Table 7 summarizes solver behavior at the largest integration horizon. Clean validation accuracy remains within the pre-declared tolerance used throughout the paper (approximately 1.74%), allowing meaningful comparison of geometric statistics and adversarial robustness.

Table 7: Three-seed summary on CIFAR-100 at $\Delta T = 30$. Values report mean $\pm$ standard deviation across three random seeds.

Solver	Val Acc (%)	higher-order trajectory norm	Jerk	FGSM (%)	PGD (%)	AutoAttack (%)
ESRKG-21	$\mathbf{53.74 \pm 0.42}$	$\mathbf{0.451 \pm 0.011}$	$(\mathbf{7.13 \pm 0.12}) \times \mathbf{10^3}$	$\mathbf{8.95 \pm 0.19}$	$\mathbf{1.86 \pm 0.16}$	$\mathbf{1.20 \pm 0.08}$
ESRKG-15	53.56 ± 0.07	0.415 ± 0.016	$(1.41 \pm 0.03) \times 10^4$	9.20 ± 0.12	1.26 ± 0.10	0.76 ± 0.09
LSRKG-15	55.30 ± 0.18	0.332 ± 0.011	$(2.47 \pm 0.07) \times 10^4$	8.79 ± 0.39	1.04 ± 0.09	0.71 ± 0.09
RK4	53.87 ± 0.05	0.870 ± 0.140	$(3.06 \pm 0.59) \times 10^5$	7.02 ± 0.63	1.47 ± 0.26	1.15 ± 0.24

Clean Accuracy. All solvers converge to a similar performance regime. LSRKG-15 achieves the highest validation accuracy (55.30 ± 0.18), followed by RK4 (53.87 ± 0.05), ESRKG-21 (53.74 ± 0.42), and ESRKG-15 (53.56 ± 0.07). Despite this spread, the results remain within the controlled tolerance, indicating that the observed robustness differences are not driven by predictive performance.

Geometric Structure. Representation geometry varies substantially across solver families. The higher-order trajectory norm ordering across methods is

$$\text{LSRKG-15} < \text{ESRKG-15} < \text{ESRKG-21} < \text{RK4}.$$

This reverses the moderate-horizon pattern, in which extended-stability methods typically exhibit larger higher-order trajectory norm. At $\Delta T = 30$, RK4 shows the largest higher-order trajectory norm, consistent with the fact that it operates close to the boundary of its classical stability region. As discussed in Section 3.1, higher-order trajectory norm depends on spectral amplification induced by the stability polynomial. When the effective argument $h\sigma$ approaches the stability boundary, the polynomial can strongly amplify higher-order Jacobian modes, producing large higher-order trajectory norm even for methods without extended stability intervals. The elevated RK4 higher-order trajectory norm in this regime is therefore more consistent with boundary-induced amplification than with the structured geometric bias typically associated with extended-stability integrators in interior regimes. Consistent with this interpretation, RK4 also exhibits substantially larger higher-order dynamics, with jerk reaching the 10^5 scale.

Adversarial Robustness. Adversarial robustness follows a structured pattern across solver families. PGD robustness increases from LSRKG-15 (1.04) to ESRKG-15 (1.26) and ESRKG-21 (1.86), with RK4 (1.47) lying between ESRKG-15 and ESRKG-21. AutoAttack follows the same qualitative ordering. FGSM, however, shows a different and weaker pattern, with RK4 achieving the lowest FGSM accuracy despite not having the lowest iterative robustness.

Stability-boundary regime (CIFAR-100, $\Delta T = 30$). At $\Delta T = 30$, the effective stability coordinate $h\sigma$ is elevated across the solvers considered, with h defined as the actual solver step size. In particular, RK4 is evaluated using its substep size $h = \Delta T/4 = 7.5$, while the ESRKG methods use $h = \Delta T = 30$. Under this convention, the solvers occupy distinct locations relative to their stability envelopes: RK4 operates near or beyond its classical stability boundary, ESRKG-21 remains comparatively deep within its extended-stability region, and ESRKG-15 and LSRKG-15 approach the outer portion of their extended-stability envelopes. As a result, solver behavior becomes increasingly influenced by the global structure of the stability region rather than by low-order geometric statistics alone.

Regime Interpretation. In general, the $\Delta T = 30$ results indicate that the solver-induced geometric structure continues to influence adversarial behavior even in highly stiff training regimes. Extended–stability Runge-Kutta methods maintain more controlled representation dynamics, whereas classical RK4 exhibits amplified higher-order behavior near the boundary of its stability region. Despite this shift in regime, clear solver-dependent robustness differences remain.

Thus, higher-order trajectory norm remains informative within a common regime, but not as a universal solver-independent ordering once operating fractions diverge.

It is important to distinguish between pooled seed-level associations and solver-mean comparisons. At the pooled seed level across all twelve runs, higher higher-order trajectory norm is generally associated with increased iterative robustness. However, comparisons between solver means can violate this ordering because different solvers may operate at distinct spectral operating points. In these cases, differences in operating fraction and stability-envelope interaction can dominate the higher-order trajectory norm signal, producing apparent inversions in solver-level robustness ordering.

4.3.7 External validation on Tiny-ImageNet

This experiment provides an external validation of the solver-induced geometry hypothesis beyond CIFAR-scale datasets. To assess whether the solver-dependent effects observed on CIFAR persist in a more challenging setting, we evaluate the same Neural ODE architecture on Tiny-ImageNet. We compare RK4, ESRKG-21, ESRKG-15, and the low-storage realization LSRKG-15 under an identical training configuration with a fixed integration horizon of $\Delta T = 20$. Each solver is trained with three independent seeds, yielding twelve runs in total.

Clean validation accuracy remains tightly clustered across all runs. The models achieve a mean validation accuracy of 33.9% with a standard deviation of 0.55%, indicating that predictive performance is closely matched between solvers. Consequently, differences in adversarial behavior are unlikely to be explained by disparities in clean accuracy.

Despite similar predictive performance, the internal dynamics induced by the solvers differ substantially. RK4 produces the smallest trajectory higher-order trajectory norm and jerk values (higher-order trajectory norm $\approx$ 0.10–0.12), while the extended-stability methods generate substantially larger geometric magnitudes. ESRKG-21 exhibits the strongest higher-order trajectory norm dynamics (higher-order trajectory norm $\approx$ 1.65–1.95), ESRKG-15 occupies an intermediate regime (higher-order trajectory norm $\approx$ 0.77–0.81), and the low-storage realization lies between RK4 and the ESRK variants (higher-order trajectory norm $\approx$ 0.44–0.50). The geometric ordering RK4 $<$ LSRKG-15 $<$ ESRKG-15 $<$ ESRKG-21 is therefore preserved on Tiny-ImageNet, matching the qualitative geometric pattern observed on CIFAR.

Adversarial robustness remains low overall due to the difficulty of Tiny-ImageNet and the absence of adversarial training. Nevertheless, a clear solver-dependent structure is present. RK4 and LSRKG-15 occupy the

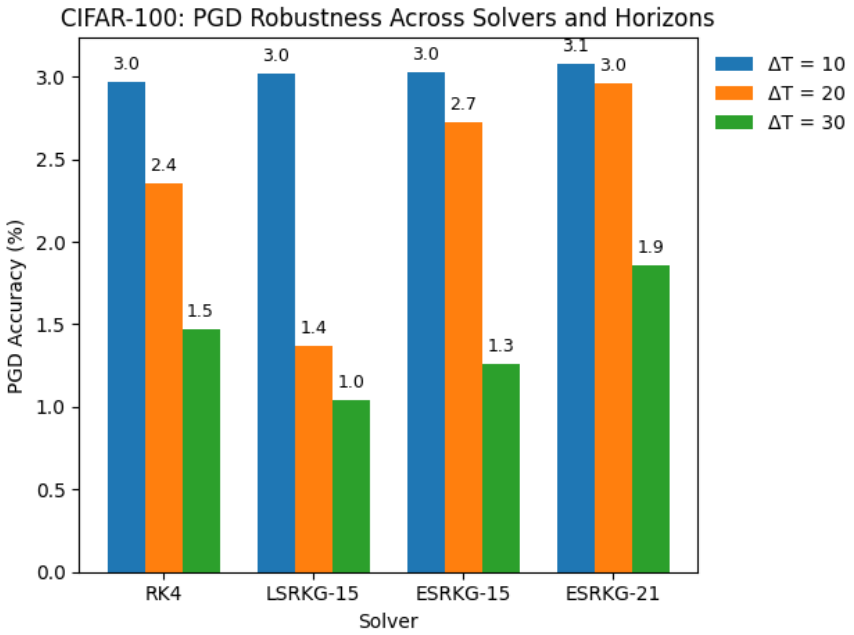


Figure 4: CIFAR-100 robustness across solvers and integration horizons. PGD adversarial accuracy (%) under increasing task difficulty. Solver-dependent robustness persists, with ESRKG-21 achieving the strongest performance. Notably, LSRKG-15 underperforms despite sharing the same stability polynomial as ESRKG-15, consistent with realization-level effects.

lowest iterative-robustness regime, while ESRKG-21 yields moderate improvements, and ESRKG-15 attains the strongest PGD and AutoAttack robustness.

Notably, although ESRKG-21 achieves the largest higher-order trajectory norm magnitude, ESRKG-15 attains the highest iterative robustness. This is consistent with the operating-fraction interpretation introduced above: on Tiny-ImageNet, robustness depends not only on higher-order trajectory norm magnitude, but also on where the learned dynamics sit within each solver's stability envelope. In particular, ESRKG-21 operates well within the stability interior ($\phi = 0.109$), whereas ESRKG-15 remains in the outer-envelope regime ($\phi = 0.687$). The Tiny-ImageNet results are therefore better interpreted in terms of operating regime, with operating fraction providing the primary coarse descriptor, rather than through higher-order trajectory norm magnitude alone.

Operating fraction and spectral regime. To characterize the effective operating regime of each solver, we compute the dimensionless operating fraction

$$\phi = \frac{h\sigma}{R},$$

where h is the step size, σ is the mean final-epoch spectral radius of the learned Jacobian, and R is the real-axis stability radius of the method. This quantity indicates where the learned dynamics lie relative to the boundary of each solver's stability envelope, and therefore which region of the stability polynomial influences the observed geometric amplification.

Table 8 reports ϕ for each solver at $\Delta T = 20$ on Tiny-ImageNet, alongside the corresponding robustness results.

The operating fractions reveal that the four solvers occupy three qualitatively distinct regimes. RK4, with $\phi = 1.37$, operates moderately beyond its classical stability envelope. Rather than inducing stable geometric amplification within the interior of the stability region, this places RK4 in a near-boundary/post-boundary regime where training must implicitly control the learned Jacobian spectrum to avoid numerical instabil-

Table 8: Spectral operating point and adversarial robustness on Tiny-ImageNet at $\Delta T = 20$. The operating fraction $\phi = h\sigma/R$ characterises where each solver operates relative to its stability boundary. RK4 operates moderately beyond its classical stability envelope. ESRKG-21 operates deep in the interior, and both ESRKG-15 and LSRKG-15 operate in the outer-envelope region.

Solver	σ	$h\sigma$	R	$\phi = h\sigma/R$	Zone	PGD (%)	AutoAttack (%)
RK4	0.764	3.82	2.78	1.37	outside / near-boundary	0.15 ± 0.04	0.07 ± 0.02
ESRKG-21	0.774	15.48	142	0.109	deep interior	0.30 ± 0.03	0.15 ± 0.04
ESRKG-15	2.473	49.46	72	0.687	outer envelope	0.41 ± 0.02	0.26 ± 0.01
LSRKG-15	2.572	51.44	72	0.714	outer envelope	0.14 ± 0.02	0.06 ± 0.01

ity. This is consistent with the comparatively suppressed higher-order trajectory norm and low iterative robustness observed for RK4. ESRKG-21, with $\phi = 0.109$, operates deep inside its stability interior, where the stability polynomial behaves approximately as a low-order Taylor approximation to the exponential and robustness is less dominated by outer-envelope effects. ESRKG-15 and LSRKG-15 both operate in the outer-envelope region ($\phi \approx 0.69$–0.71), where higher-order polynomial structure increasingly influences the spectral reweighting of Jacobian eigenmodes, enabling stronger envelope-driven amplification.

The comparison between ESRKG-15 and LSRKG-15 illustrates this hierarchy. Both methods occupy nearly the same operating regime ($\phi = 0.687$ and $\phi = 0.714$), so their robustness difference cannot be explained by stability-envelope position alone. Instead, the approximately threefold PGD gap is consistent with realization-level amplification differences, as measured by the diagnostic $\|(I - zA)^{-1}\|_\infty$. Thus, ϕ explains broad regime placement, while realization-level amplification distinguishes methods within the same regime.

Realization-level effects within the envelope zone. The comparison between ESRKG-15 and LSRKG-15 on Tiny-ImageNet provides one of the clearest empirical examples of realization-level effects in the paper. Both solvers share numerically equivalent stability polynomials and converge to nearly identical operating fractions ($\phi = 0.687$ and $\phi = 0.714$, respectively), placing them in the same outer-envelope zone. Despite this, their adversarial robustness differs by approximately threefold: ESRKG-15 achieves PGD $0.41 \pm 0.02\%$ while LSRKG-15 reaches only PGD $0.14 \pm 0.02\%$.

This gap is most consistent with a realization-level difference in internal stage amplification, since it is unlikely to be explained by the stability zone, operating fraction, higher-order trajectory norm ordering, or clean accuracy in these experiments. As established in Section 4.4.2, the two realizations remain sharply separated under the diagnostic $\|(I - zA)^{-1}\|_\infty$ even at the experimentally observed operating points. For Tiny-ImageNet at $\Delta T = 20$, the corresponding values are approximately 3.44×10^5 for ESRKG-15 and 3.29×10^4 for LSRKG-15, indicating an order-of-magnitude difference in stage propagation behaviour despite external polynomial equivalence. In nonlinear Neural ODE training settings, such realization-level amplification differences may modulate how the Jacobian compositions accumulate across stages, potentially producing distinct representation geometry within the same stability envelope.

Seed-level correlation analysis. To quantify the relationship between trajectory geometry and adversarial robustness across all twelve runs, we computed seed-level Pearson correlations using final-epoch metrics. Final higher-order trajectory norm shows a positive association with PGD robustness ($r = 0.67$) and AutoAttack robustness ($r = 0.74$), while validation accuracy exhibits negligible correlation with robustness ($|r| < 0.1$). The solver-mean inversion between ESRKG-21 and ESRKG-15 does not contradict this positive seed-level association, because the former reflects differences in the operating regime across solvers, whereas the latter captures within-solver variation across seeds. This distinction is consistent with the operating-fraction framework developed above.

4.3.8 Regime Comparison: CIFAR-10 vs CIFAR-100 at $\Delta T = 30$

Although both datasets are trained with identical architectures, optimizers, and integration horizons, the behavior at $\Delta T = 30$ differs between CIFAR-10 and CIFAR-100 in a quantitatively identifiable way. On

Table 9: Tiny-ImageNet results at $\Delta T = 20$ (mean $\pm$ standard deviation over three seeds).

Solver	Val Acc (%)	higher-order trajectory norm	Jerk	FGSM (%)	PGD (%)	AutoAttack (%)
RK4	33.57 ± 0.52	0.109 ± 0.012	$(8.1 \pm 0.7) \times 10^2$	6.53 ± 0.41	0.15 ± 0.04	0.07 ± 0.02
LSRKG-15	34.45 ± 0.48	0.471 ± 0.032	$(9.9 \pm 0.8) \times 10^3$	6.19 ± 0.37	0.14 ± 0.02	0.06 ± 0.01
ESRKG-15	$\mathbf{33.72 \pm 0.63}$	$\mathbf{0.793 \pm 0.021}$	$(\mathbf{1.17 \pm 0.09}) \times \mathbf{10^4}$	$\mathbf{5.43 \pm 0.35}$	$\mathbf{0.41 \pm 0.02}$	$\mathbf{0.26 \pm 0.01}$
ESRKG-21	33.87 ± 0.57	1.80 ± 0.15	$(4.61 \pm 0.32) \times 10^4$	5.66 ± 0.40	0.30 ± 0.03	0.15 ± 0.04

CIFAR-10, higher-order trajectory norm no longer exhibits a positive association with PGD robustness across solvers. Seed-level analysis yields $r = -0.41$ ($n = 12$), indicating that higher-order trajectory norm ceases to serve as a reliable standalone descriptor of robustness in this regime. Spectral diagnostics show that the solvers separate into distinct operating bands: ESRKG-21 remains in a low-σ regime ($\sigma \approx 0.6$), RK4 occupies an intermediate band ($\sigma \approx 0.8$–0.9), while ESRKG-15 and LSRKG-15 converge to substantially larger spectral radii ($\sigma \approx 1.6$–2.1). When combined with the respective stability radii of the methods, these values correspond to markedly different operating fractions $\phi = h\sigma/R$, indicating that the solvers are no longer operating within a common stability regime. In this cross-regime setting, robustness variation is no longer monotonically aligned with higher-order trajectory norm magnitude, consistent with a transition toward stability-envelope-mediated dynamics.

On CIFAR-100, a closely comparable spectral separation is observed: ESRKG-21 operates at $\sigma \approx 0.6$, RK4 at $\sigma \approx 0.8$–0.9, LSRKG-15 at $\sigma \approx 1.7$–1.8, and ESRKG-15 at $\sigma \approx 1.9$–2.0. Despite this similar spectral structure, higher-order trajectory norm remains positively associated with PGD robustness ($r = 0.628$, $p = 0.028$, $n = 12$). Thus, while both datasets exhibit solver-dependent spectral separation at $\Delta T = 30$, the geometry–robustness coupling weakens in CIFAR-10 but persists in CIFAR-100.

These results indicate that the dataset-level divergence at $\Delta T = 30$ is not explained solely by the magnitude of the Jacobian spectrum. Instead, it is consistent with an interaction between dataset-specific spectral statistics and the stability-envelope geometry of the integrator. CIFAR-10 exhibits a regime transition in which higher-order trajectory norm is no longer the dominant standalone descriptor of robustness, whereas CIFAR-100 retains a statistically significant geometry–robustness association. We interpret this as evidence consistent with a regime-dependent explanation rather than definitive proof of causality. Nevertheless, the results are consistent with the central claim of the paper: solver-induced geometric effects persist across datasets, but their functional manifestation depends on how dataset-specific Jacobian spectra interact with the stability-envelope geometry of the integrator. This positive association is computed at the seed level across all twelve runs, and should not be interpreted as requiring a monotone ordering of solver means in Table 7.

4.4 Experiment II: Controlled Perturbations of Solver Structure

The experiments presented thus far compare solvers drawn from a fixed set of methods: RK4, ESRKG-15, ESRKG-21, and LSRKG-15. A natural question is whether the observed geometric and robustness differences reflect only discrete variation across solver families, or whether systematic changes in the stability polynomial are associated with corresponding changes in the learned dynamics.

4.4.1 Controlling Stability Geometry with Gegenbauer Polynomials

To investigate this, we vary the Gegenbauer parameter ν used in the RKG construction described in Section 2.2. The parameter ν determines the shape of the stability polynomial and therefore the geometry of the stability region. Changing ν modifies the stability envelope while preserving the order of accuracy, stage count, and training configuration.

We study four configurations at fixed stage count $s = 15$, as summarized in Table 10, using the original ESRKG-15 construction with $\nu = \frac{1}{4}$ as the baseline. The Gegenbauer sweep is run as an independent controlled experiment: the $\nu = \frac{1}{4}$ row is a separately trained baseline within the sweep and is not intended to duplicate the ESRKG-15 aggregate reported in Table 6. All models are trained using identical architectures, optimization settings, stage counts, and numbers of function evaluations. Altering only the Gegenbauer

Table 10: Effect of modifying the stability polynomial using the Gegenbauer parameter on CIFAR-100 at $\Delta T = 20$.

Configuration	ν	Stability interval	PGD (%)	AutoAttack (%)
Shortened	2	$[-51, 0]$	1.22 ± 0.02	0.87 ± 0.05
Moderate	$\frac{1}{2}$	$[-66, 0]$	1.13 ± 0.19	0.86 ± 0.17
Baseline	$\frac{1}{4}$	$[-72, 0]$	2.00 ± 0.14	1.37 ± 0.09
Extended	$\frac{1}{10}$	$[-80, 0]$	2.23 ± 0.05	1.55 ± 0.02

Table 11: Operating fraction $\phi = h\sigma/R$ for the Gegenbauer sweep on CIFAR-100 at $\Delta T = 20$.

ν	R	σ	$h\sigma$	ϕ	higher-order trajectory norm	PGD
2	51	1.17	23.4	0.46	0.43	1.22%
$\frac{1}{2}$	66	0.68	13.5	0.21	0.18	1.13%
$\frac{1}{4}$	72	2.35	46.9	0.65	0.89	2.00%
$\frac{1}{10}$	80	2.32	46.4	0.58	0.74	2.23%

parameter produces substantial changes in the resulting robustness. Relative to the shortened configurations ($\nu = 2$ and $\nu = \frac{1}{2}$), the baseline $\nu = \frac{1}{4}$ and extended $\nu = \frac{1}{10}$ configurations achieve substantially stronger robustness under both PGD and AutoAttack. This indicates that robustness does not depend solely on the length of the stability interval, but on the operating regime induced by the interaction between the stability polynomial and the learned spectrum.

To better understand this behavior, we examine the effective stability coordinate $h\sigma$, where σ denotes the learned Jacobian spectral radius and h is the integration step size. Table 11 reports the resulting operating fraction $\phi = h\sigma/R$ for each configuration.

These results show that the different configurations occupy distinct operating regimes within their stability envelopes. In particular, the $\nu = \frac{1}{4}$ and $\nu = \frac{1}{10}$ configurations operate at larger effective operating fractions ($\phi \approx 0.65$ and $\phi \approx 0.58$, respectively) than the $\nu = 2$ and $\nu = \frac{1}{2}$ configurations, and also exhibit stronger geometric amplification and improved iterative robustness. By contrast, the lower-ϕ configurations remain deeper inside the stability region and exhibit weaker higher-order trajectory norm and substantially reduced robustness.

Notably, extending the stability interval from $\nu = \frac{1}{4}$ to $\nu = \frac{1}{10}$ does not simply produce a monotonic effect through interval length alone. Instead, both configurations occupy nearby high-ϕ regimes and yield comparably strong robustness, with the extended configuration slightly outperforming the baseline in these runs. This is consistent with the view that robustness is influenced by the resulting operating regime rather than by stability radius in isolation.

The baseline configuration ($\nu = \frac{1}{4}$) operates slightly closer to the stability boundary and exhibits stronger geometric amplification. However, robustness does not vary monotonically with boundary proximity alone. The extended configuration ($\nu = \frac{1}{10}$) remains in a similarly high-ϕ regime and, in these runs, yields slightly higher PGD and AutoAttack robustness.

Although all configurations use the same ESRKG-15 scheme and are identical training setup, we observe substantial variation in the learned spectral radius σ across values of ν. This suggests that the stability polynomial actively shapes the spectral regime reached during training, rather than merely acting on a fixed underlying spectrum.

This interpretation also helps clarify the behavior of the extended configuration ($\nu = \frac{1}{10}$). Although the stability interval increases, the resulting operating fraction decreases, moving the learned dynamics slightly further from the stability boundary. However, both configurations remain in a high-ϕ regime, and in these runs the extended configuration is observed to slightly improve robustness.

Overall, these experiments provide controlled empirical evidence that systematic changes in the stability polynomial are associated with corresponding changes in operating regime, representation geometry, and

adversarial robustness. By modifying a single scalar parameter in the polynomial construction while holding architecture, optimizer, and training procedure fixed, we observe systematic variation in the learned dynamics and their robustness properties.

Collectively with the earlier solver-family comparisons, these results are consistent with the view that the stability polynomial functions as an additional design dimension in Neural ODE training. At the same time, the resulting operating fraction remains dependent on the learned spectrum and is therefore best interpreted diagnostically rather than as a fully predictive design rule.

4.4.2 External Equivalence of ESRKG-15 Realizations

To verify algebraic equivalence between the classical and $2S$–1 (LSRKG-15) realizations of ESRKG-15, we perform a series of numerical checks on their Butcher tableaux.

Stability Polynomial. We compute the stability function

$$R(z) = 1 + zb^\top (I - zA)^{-1}\mathbf{e}$$

for both realizations over a dense real-axis grid restricted to the stability interval. The maximum observed difference satisfies

$$\max_z |R_1(z) - R_2(z)| \approx 10^{-4},$$

while the maximum coefficient-wise difference in the corresponding stability polynomials is on the order of 10^{-7}. These discrepancies are small relative to the stability-scale considered here, and we therefore treat the two realizations as externally equivalent within the operating regime studied in our experiments.

Complex Stability Region. To verify equivalence beyond the real axis, we compute the stability boundary defined by $|R(z)| = 1$ in the complex plane. Figure 5 shows the resulting contours for both realizations. The stability regions overlap to plotting precision across the entire complex domain, confirming that the two methods share the same global stability geometry. Minor visible deviations are attributable to numerical evaluation and contour discretization rather than differences in the underlying stability polynomial.

Order Conditions and B-Series. Both realizations satisfy third-order conditions up to machine precision. The corresponding third-order B-series tree coefficients agree to within 5×10^{-7}. These results confirm formal equivalence up to order three. Thus, agreement at the level of the stability polynomial and low-order B-series coefficients does not guarantee identical learned dynamics under nonlinear training.

Nonlinear Test Equation. When applied to the nonlinear scalar test problem $y' = y^2$, the two methods produce final states differing by less than 10^{-8} over multiple steps. Thus, at the level of classical test equations, the realizations are effectively indistinguishable.

Internal Amplification. Despite this external equivalence, the two realizations remain sharply separated under the realization-level diagnostic

$$\|(I - zA)^{-1}\|_\infty.$$

We interpret this quantity as a numerical diagnostic of intermediate-stage propagation rather than as a direct measurement of the states visited during training. In our implementation, ESRKG-15 consistently attains larger values than LSRKG-15 over the experimentally relevant portion of the negative real axis. Over the interval $z \in [-72, 0]$, the maximum values are approximately 5.996×10^6 for ESRKG-15 and 1.188×10^5 for LSRKG-15. At common test values, the same ordering persists: for example,

$$\|(I + 10A_{\text{ESRKG}})^{-1}\|_\infty \approx 4.11 \times 10^2, \qquad \|(I + 10A_{\text{LSRKG}})^{-1}\|_\infty \approx 9.12 \times 10^1,$$

with the separation increasing further at $z = -20$ and $z = -30$.

More importantly, this ordering also persists at the operating points observed in training. For CIFAR-10 at $\Delta T = 20$, the experimentally relevant values $z \approx -h\sigma$ yield internal amplification of approximately 6.58×10^5

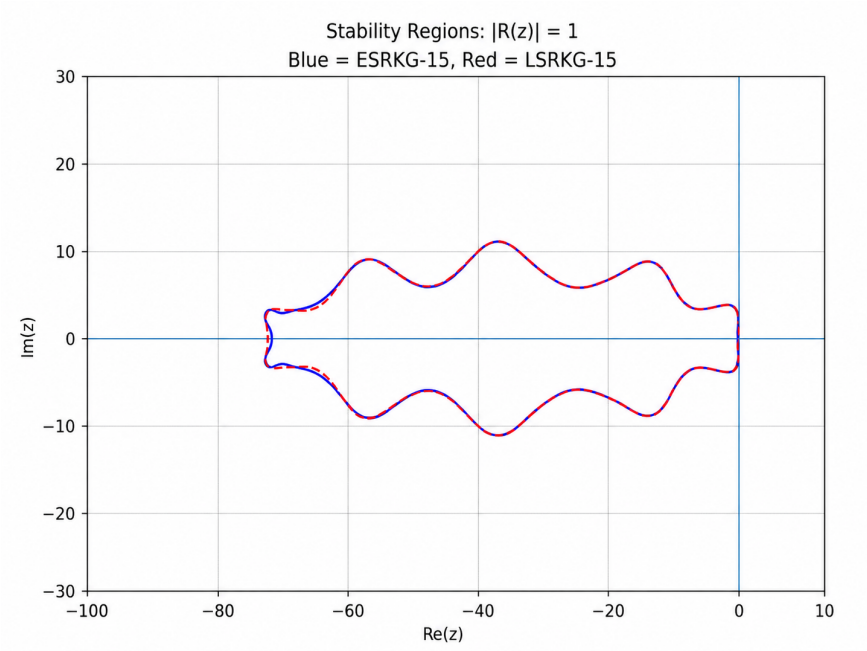


Figure 5: Stability regions defined by $|R(z)| = 1$ for the full-storage and $2S$–1 (LSRKG-15) realizations of ESRKG-15. The boundaries overlap to plotting precision across the complex plane, confirming the external equivalence of the stability polynomial.

for ESRKG-15 and 2.02×10^4 for LSRKG-15. For Tiny-ImageNet at $\Delta T = 20$, the corresponding values are approximately 3.44×10^5 and 3.29×10^4. Thus, although the two realizations share the same stability polynomial, they remain strongly separated in their stage-propagation behaviour at the operating points actually visited during training.

These results confirm that while the two ESRKG-15 realizations are algebraically equivalent under classical stability analysis, they exhibit substantially different internal amplification structure. In nonlinear Neural ODE training, where Jacobians are repeatedly composed and evaluated at stage-dependent states, such realization-level differences provide a plausible mechanism by which higher-order geometric quantities may be weighted differently across realizations.

Although the two realizations share the same stability polynomial, their internal stage amplification can interact differently with the Jacobian of the learned dynamics. This provides a realization-level mechanism consistent with distinct geometric behaviour even when the external stability region remains identical. In particular, the operating-point diagnostic is more relevant than a purely worst-case comparison for interpreting the training dynamics, since it evaluates each realization at the values $z \approx -h\sigma$ actually reached in the experiments.

In addition, we verify this equivalence at the step sizes used in our experiments ($\Delta T \in \{10, 20, 30\}$). For the corresponding values $z = h\lambda$, we obtain

$$|R_1(z) - R_2(z)| \leq 2.5 \times 10^{-4},$$

confirming that the two realizations are numerically indistinguishable within the operating regime of interest. For the step sizes used in the experiments, the corresponding linear-test-equation values satisfy $|R_1(z) - R_2(z)| = 3.87 \times 10^{-6}$ at $z = -10$, 1.76×10^{-5} at $z = -20$, and 2.46×10^{-4} at $z = -30$. Over 10 repeated steps, the resulting trajectory differences remain small and decay to below 10^{-7}, confirming practical external equivalence across the operating regime considered in this work.

These observations suggest a hierarchical interpretation of solver behaviour. The stability polynomial organizes the global operating regime through the operating fraction ϕ, while realization-level internal amplifi-

Table 12: Relative improvement (%) in adversarial robustness of ESRKG-21 over RK4 under matched clean accuracy, computed directly from the paper's reported solver means and rounded to the nearest integer.

	CIFAR-10			CIFAR-100		
ΔT	FGSM	PGD	AutoAttack	FGSM	PGD	AutoAttack
10	-2	$+18$	$+15$	-2	$+4$	$+4$
20	-2	$+63$	$+60$	$+27$	$+26$	$+38$
30	-2	$+35$	$+87$	$+27$	$+27$	$+4$

cation refines the detailed dynamics within that regime. Consequently, methods sharing the same stability polynomial can still produce different geometric behaviour because of differences in their internal stage structure. In particular, two realizations sharing the same stability polynomial, such as ESRKG-15 and LSRKG-15, can exhibit different geometric behaviour depending on how strongly higher-order Jacobian modes are amplified during stage propagation at the operating points reached in training. The precise ordering between such realizations therefore depends not only on the common stability polynomial, but also on realization-level amplification within the dataset-specific spectral regime of the learned dynamics. For both realizations, this worst-case value is attained near the stability boundary at $z \approx -72$. More importantly for the present experiments, the same ordering persists at the observed operating points $z \approx -h\sigma$: ESRKG-15 remains approximately one to two orders of magnitude larger than LSRKG-15, with ratios ranging from about $10\times$ on Tiny-ImageNet at $\Delta T = 20$ to about $32\times$ on CIFAR-10 at $\Delta T = 20$.

4.4.3 Relative Robustness Gains from Extended-Stability Integrators

Models are trained without adversarial objectives to isolate solver-induced effects, so absolute robustness levels are modest. The results in this subsection should therefore be interpreted as relative differences under matched predictive performance rather than as standalone robustness claims. The relative gain is defined as

$$\text{Relative Gain}(\%) = \frac{R_{\text{solver}} - R_{\text{RK4}}}{R_{\text{RK4}}} \times 100.$$

Table 12 summarizes the relative gains of ESRKG-21 over RK4 across datasets and horizons. On CIFAR-10, ESRKG-21 improves iterative robustness relative to RK4 at all horizons. PGD relative gains peak at $\Delta T = 20$, while AutoAttack relative gains increase further at $\Delta T = 30$.On CIFAR-100, AutoAttack gains are strongest at $\Delta T = 20$ and drop at $\Delta T = 30$, while PGD relative gains remain similar after rounding. The absolute PGD improvement is larger at $\Delta T = 20$, but the small absolute robustness values at $\Delta T = 30$ make percentage gains sensitive to the RK4 baseline.

4.4.4 Seed-Level Geometry Robustness Analysis

To assess whether solver-induced geometric quantities are associated with adversarial robustness, we perform a seed-level correlation analysis rather than relying on solver-averaged statistics. All correlations are computed across individual runs (four solvers $\times$ three seeds, $n = 12$) using final-epoch metrics. No solver-level averaging is performed.

We compute Pearson correlation coefficients between higher-order trajectory norm, jerk, validation accuracy, and adversarial robustness (PGD and AutoAttack). This allows us to examine whether robustness is associated with solver-induced geometric amplification or merely with clean accuracy.

CIFAR-10 at $\Delta T = 10$. At $\Delta T = 10$, we observe strong positive seed-level associations between geometric quantities and robustness:

- higher-order trajectory norm vs PGD: $r = 0.84$, 95% CI $[0.51, 0.95]$, $p < 0.001$
- higher-order trajectory norm vs AutoAttack: $r = 0.66$, 95% CI $[0.14, 0.90]$, $p = 0.019$

- Jerk vs PGD: $r = 0.64$, 95% CI $[0.10, 0.89]$, $p = 0.026$
- Jerk vs AutoAttack: $r = 0.59$, 95% CI $[0.03, 0.87]$, $p = 0.043$

In contrast, clean validation accuracy does not exhibit a statistically significant association with robustness:

- Accuracy vs PGD: $r = -0.33$, $p = 0.29$
- Accuracy vs AutoAttack: $r = -0.46$, $p = 0.14$

. Although the PGD association falls marginally short of conventional significance thresholds at this horizon ($p = 0.056$), AutoAttack robustness exhibits a statistically significant positive association with higher-order trajectory norm. Taken together, these results remain consistent with a positive geometry–robustness relationship at $\Delta T = 10$, albeit with greater variability than observed on CIFAR-10.

CIFAR-10 at $\Delta T = 20$. At the extended horizon $\Delta T = 20$, higher-order trajectory norm exhibits a positive but weaker association with iterative robustness than at $\Delta T = 10$. Across pooled seed-level runs, the Pearson correlation between higher-order trajectory norm and PGD robustness is $r = 0.58$, 95% CI $[0.02, 0.87]$, $p = 0.047$, while the correlation between higher-order trajectory norm and AutoAttack robustness is $r = 0.56$, 95% CI $[-0.01, 0.86]$, $p = 0.058$.

Although ESRKG-21 achieves the strongest iterative robustness at this horizon, it does not exhibit the largest higher-order trajectory norm magnitude. Increased within-solver variance, together with partial overlap between intermediate methods, weakens the overall monotonic association between higher-order trajectory norm and robustness when correlations are computed across all seed-level runs rather than solver means alone. This behavior is consistent with a transitional operating regime in which stability-envelope effects begin to dominate a simple higher-order trajectory norm-based ordering.

Jerk magnitude exhibits only a weak-to-moderate association with robustness, while clean validation accuracy shows no statistically significant association with iterative robustness.

Overall, these results suggest that the higher-order trajectory norm proxy remains informative at $\Delta T = 20$, but is no longer sufficient on its own to predict robustness across all solver regimes.

CIFAR-10 at $\Delta T = 30$. To verify that the regime transition observed at the solver-mean level is not an artefact of averaging across solvers, we evaluate correlations at the seed level across all twelve individual runs (three seeds per solver).

higher-order trajectory norm–PGD association. At $\Delta T = 30$ on CIFAR-10, the seed-level Pearson correlation between final-epoch higher-order trajectory norm and PGD robustness is

$$r = -0.37, \quad 95\% \text{ CI} = [-0.78, 0.24], \quad p = 0.24.$$

Similarly, the correlation between higher-order trajectory norm and AutoAttack robustness is

$$r = -0.39, \quad 95\% \text{ CI} = [-0.79, 0.22], \quad p = 0.21.$$

These moderate negative but statistically non-significant associations indicate that second-order geometric magnitude no longer serves as a reliable predictor of adversarial robustness at this extended horizon.

This contrasts with the strong positive association observed at $\Delta T = 10$ and the weaker positive trend observed at $\Delta T = 20$, supporting a regime-transition interpretation in which higher-order trajectory norm-dominated dynamics weaken as operating-fraction effects become increasingly dominant. At $\Delta T = 30$, solver-dependent stability behavior appears to govern robustness more strongly than simple geometric magnitude alone.

Directional gradient effects. In contrast to higher-order trajectory norm magnitude, directional gradient alignment (measured via inter-epoch cosine similarity) appears more variable across solver regimes at $\Delta T = 30$. Runs with lower inter-epoch gradient alignment often coincide with higher PGD robustness, suggesting that gradient-flow directionality may become more relevant in the extended horizon regime. However, we treat this as a qualitative diagnostic rather than a quantified predictive relationship.

Regime transition. Taken together, the seed-level results indicate a shift in the dominant mechanism associated with robustness at $\Delta T = 30$. At moderate horizons, robustness is positively associated with higher-order trajectory norm-driven higher-order geometric amplification. At extended horizons, this association weakens and becomes statistically non-significant, while robustness appears increasingly shaped by solver stability-envelope interaction and directional gradient-flow structure. These findings are consistent with the interpretation that large-ΔT behavior is not governed solely by local Taylor-style geometric amplification, but also by interaction with the outer structure of each method's stability polynomial.

CIFAR-100 at $\Delta T = 10$: Seed-Level Correlation Using final-epoch values across all twelve individual runs (four solvers $\times$ three seeds), higher-order trajectory norm exhibits a moderate positive association with iterative robustness.

For PGD robustness, the seed-level Pearson correlation is

$$r = 0.56, \quad 95\% \text{ CI} = [0.00, 0.86], \quad p = 0.056,$$

while Spearman's rank correlation yields

$$\rho = 0.59.$$

For AutoAttack robustness, the Pearson correlation is

$$r = 0.63, \quad 95\% \text{ CI} = [0.08, 0.89], \quad p = 0.028,$$

with corresponding Spearman correlation

$$\rho = 0.62.$$

Clean validation accuracy shows no statistically significant association with PGD robustness.

These results indicate that even at the shortest horizon, solver-dependent geometric higher-order trajectory norm is associated with measurable differences in iterative adversarial robustness, although the association is weaker and more variable than on CIFAR-10.

CIFAR-100 at $\Delta T = 20$. Using final-epoch values across all twelve individual runs (four solvers $\times$ three seeds), higher-order trajectory norm exhibits a strong positive association with iterative robustness.

For PGD robustness, the seed-level Pearson correlation is

$$r = 0.79, \quad 95\% \text{ CI} = [0.43, 0.94], \quad p = 0.002,$$

while Spearman's rank correlation yields

$$\rho = 0.83.$$

For AutoAttack robustness, the Pearson correlation is

$$r = 0.83, \quad 95\% \text{ CI} = [0.53, 0.95], \quad p < 0.001,$$

with corresponding Spearman correlation

$$\rho = 0.85.$$

Clean validation accuracy exhibits no statistically significant association with PGD robustness.

These results indicate that at $\Delta T = 20$, solver-induced geometric higher-order trajectory norm is strongly associated with iterative adversarial robustness at the seed level, with statistically significant positive correlations across both PGD and AutoAttack evaluations.

CIFAR-100 at $\Delta T = 30$: Seed-Level Correlation Using final-epoch values across all twelve individual runs (four solvers $\times$ three seeds), higher-order trajectory norm exhibits a moderate positive association with PGD robustness:

$$r = 0.61, \quad 95\% \text{ CI} = [0.06, 0.88], \quad p = 0.035, \quad n = 12.$$

Spearman's rank correlation yields

$$\rho = 0.66,$$

indicating a broadly monotonic relationship across solver regimes.

For AutoAttack robustness, the Pearson correlation is

$$r = 0.57, \quad 95\% \text{ CI} = [0.00, 0.86], \quad p = 0.051,$$

with corresponding Spearman correlation

$$\rho = 0.59.$$

While weaker than the strong positive association observed at $\Delta T = 20$, higher-order trajectory norm remains positively associated with iterative adversarial robustness at extended integration horizons on CIFAR-100.

To address multiple comparisons across the twelve dataset–horizon–attack combinations, we applied Benjamini–Hochberg correction at FDR = 0.05. Three higher-order trajectory norm–robustness associations remain significant after correction: CIFAR-10 at $\Delta T = 10$ for higher-order trajectory norm versus PGD (raw $p = 0.0007$, $q = 0.006$), CIFAR-100 at $\Delta T = 20$ for higher-order trajectory norm versus PGD (raw $p = 0.002$, $q = 0.008$), and CIFAR-100 at $\Delta T = 20$ for higher-order trajectory norm versus AutoAttack (raw $p = 0.001$, $q = 0.006$). All three surviving associations occur in moderate operating regimes, consistent with the regime-dependent prediction of Section 3.1. The non-significant associations include the CIFAR-10 $\Delta T = 30$ tests, where the regime-transition prediction anticipates that higher-order trajectory norm should no longer act as a standalone robustness descriptor. Thus, the pattern after FDR correction supports a regime-dependent interpretation rather than a universal higher-order trajectory norm–robustness law.

4.5 Experiment III: Solver-Swap Intervention

The preceding experiments show that different solvers are associated with different learned geometric regimes under matched training conditions. We now test whether this solver dependence is merely correlational, or whether trained checkpoints are functionally tied to the discrete flow operator used during training. To do so, we perform a frozen-weight solver-swap intervention. A model is first trained to convergence with a given numerical integrator. We then freeze all learned weights and re-evaluate the same checkpoint under alternative solvers, keeping the architecture, parameters, input data, and integration horizon fixed. Thus, the numerical integrator, together with its inherent step decomposition, is the only component changed at test time.

If the solver were an interchangeable post hoc implementation detail, then replacing it after training should preserve predictive performance up to modest numerical variation. Conversely, systematic degradation under solver replacement would indicate that the trained Neural ODE has adapted to the discrete flow induced by its native integrator. The solver-swap experiment therefore provides an intervention on the deployed dynamics: it asks whether the learned representation remains functional when the discretization rule is changed, while the learned vector field is held fixed.

The solver-swap matrices report checkpoint-level evaluations from independently trained models rather than three-seed aggregate summaries. Each row corresponds to a frozen checkpoint trained with a given solver and then re-evaluated under alternative solvers; off-diagonal degradation is therefore interpreted relative to that checkpoint's native diagonal accuracy. For the ESRKG and LSRKG configurations, the same Gegenbauer construction and parameter setting used in the main experiments are retained.

4.5.1 CIFAR-100 Solver Swap Intervention ($\Delta T = 10$)

Table 13 reports the full solver-swap matrix on CIFAR-100 at $\Delta T = 10$. Rows denote the solver used during training, and columns denote the solver used only at evaluation time. Native evaluations preserve normal

Table 13: CIFAR-100 solver-swap intervention at $\Delta T = 10$. Rows denote the training solver (frozen checkpoint), and columns denote the evaluation solver. Native evaluations (diagonal) preserve performance, while off-diagonal entries show severe degradation under solver swap.

Train \ Eval	RK4	ESRKG-21	ESRKG-15	LSRKG-15
RK4	51.70	1.51	0.86	1.26
ESRKG-15	4.12	2.22	50.06	0.84
ESRKG-21	1.36	51.40	1.38	1.16
LSRKG-15	12.31	1.23	6.68	51.24

Table 14: CIFAR-100 solver-swap intervention at $\Delta T = 20$. Rows denote the training solver (frozen checkpoint), and columns denote the evaluation solver. Native evaluations (diagonal) preserve performance, while off-diagonal entries show severe degradation under solver swap.

Train \ Eval	RK4	ESRKG-21	ESRKG-15	LSRKG-15
RK4	50.22	0.99	2.00	0.86
ESRKG-15	5.98	0.84	51.77	1.13
ESRKG-21	0.19	51.97	1.01	1.00
LSRKG-15	0.93	1.07	1.12	50.54

predictive performance, with diagonal accuracies around 50–51%. In contrast, every non-native evaluation produces severe degradation. Most swaps collapse to near-random CIFAR-100 performance, while a small number of configurations retain partial accuracy, such as LSRKG-15 $\rightarrow$ RK4 and LSRKG-15 $\rightarrow$ ESRKG-15.

This pattern shows that solver mismatch is not simply binary, but graded: some pairs of discrete flows appear partially compatible in this moderate-horizon regime, while others destroy the learned representation almost completely. Nevertheless, all off-diagonal entries remain far below the corresponding native accuracy, demonstrating that the trained checkpoints are not functionally invariant to solver replacement.

4.5.2 CIFAR-100 Solver Swap Intervention ($\Delta T = 20$)

We next repeat the intervention at the larger integration horizon $\Delta T = 20$. Table 14 shows an even sharper pattern. Native evaluations again preserve normal predictive performance, with all diagonal entries near 50–52%. By contrast, almost every off-diagonal solver replacement collapses to approximately 0–2% accuracy. The only partial exception is ESRKG-15 $\rightarrow$ RK4, which retains 5.98% accuracy, still representing a severe drop from the native 51.77% checkpoint performance.

The full matrix demonstrates that this effect is bidirectional and pervasive. RK4-trained checkpoints fail under extended-stability solvers, while ESRKG-trained checkpoints also fail when evaluated under RK4. Moreover, collapse occurs not only across distinct solver families, but also between ESRKG-15 and LSRKG-15. This is especially important because these two methods share the same stability polynomial but differ in realization and internal stage propagation. Thus, the solver-swap intervention supports a stronger conclusion than stability-region mismatch alone: the deployed numerical realization can form part of the effective learned model.

Together, the $\Delta T = 10$ and $\Delta T = 20$ interventions provide intervention-based evidence that the solver is not acting as an interchangeable numerical component after training. Instead, each checkpoint appears adapted to the discrete flow induced by its training-time integrator. The degradation is strongest at $\Delta T = 20$, where nearly all non-native evaluations collapse to near-random accuracy, but the $\Delta T = 10$ results reveal an additional nuance: solver compatibility is graded, with limited partial transfer possible between certain solver pairs. This supports the broader view that solver choice shapes the effective learned dynamics of Neural ODEs, rather than merely approximating a solver-independent continuous model.

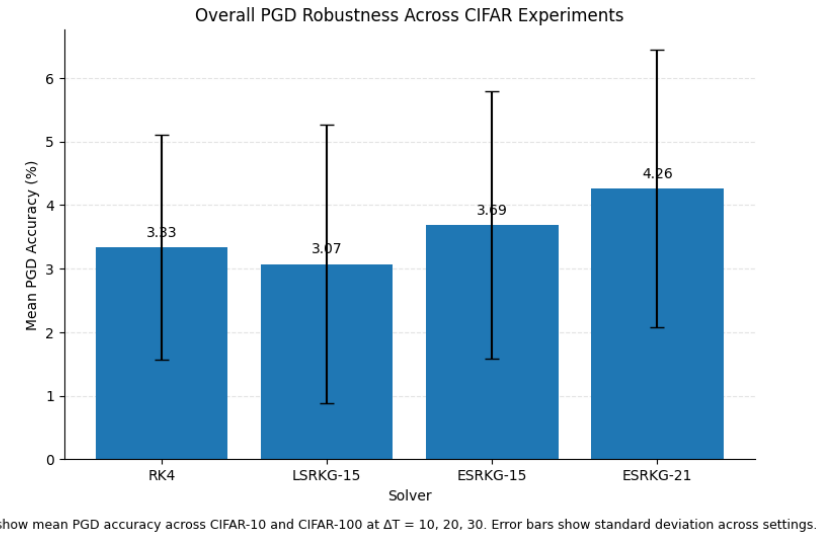


Figure 6: Overall robustness summary across all experiments. Mean PGD adversarial accuracy (%) is aggregated over CIFAR-10 and CIFAR-100 at $\Delta T \in \{10, 20, 30\}$. Error bars indicate standard deviation across settings. Extended-stability solvers generally achieve higher robustness than RK4, with ESRKG-21 providing the strongest overall performance.

5 Conclusions

This work provides intervention-based evidence that the numerical integrator is not always functionally interchangeable after training in Neural ODEs under the tested setup. In particular, we provide evidence that models trained with one solver can degrade severely when evaluated with another under a frozen-weight solver-swap intervention, indicating that trained checkpoints are not functionally invariant to post hoc solver replacement. Across CIFAR-10 and CIFAR-100, and integration horizons $\Delta T \in \{10, 20, 30\}$, we observe that solvers with different stability polynomials are associated with systematically different geometric regimes, even when clean accuracy is tightly matched. Using a third-order formulation, we directly measure trajectory higher-order trajectory norm and higher-order dynamics, and observe that these quantities are associated with adversarial robustness under iterative attacks.

higher-order trajectory norm is associated with iterative adversarial robustness in moderate operating regimes, but this association is not uniform across datasets or horizons. On CIFAR-10, the relationship is strongest at $\Delta T = 10$ and weakens as the integration horizon increases, especially at $\Delta T = 30$. On CIFAR-100, the association remains positive across horizons and is strongest at $\Delta T = 20$. At $\Delta T = 30$, stability-envelope effects become more prominent, and higher-order trajectory norm no longer provides a solver-independent ordering. We further provide evidence that algebraically equivalent Runge–Kutta realizations can produce distinct learned dynamics, suggesting that solver behavior can depend not only on the stability polynomial, but also on realization-level amplification. Together with controlled perturbations of the stability polynomial via the Gegenbauer parameter, these results support a hierarchical interpretation in which the stability polynomial is associated with the global operating regime, while internal amplification influences behavior within that regime.

While robustness differences under extended-stability solvers are most consistent under iterative attacks, differences under FGSM are smaller, reflecting the local nature of single-step adversaries relative to the higher-order geometric effects probed by multi-step attacks. Absolute robustness remains modest because the models are trained without adversarial objectives.

Several limitations remain. Experiments are restricted to CIFAR-scale datasets and ℓ_∞-bounded attacks; evaluation on larger datasets (e.g., ImageNet) and additional threat models would further clarify generality. The third-order embedding increases state dimensionality, and extended-stability solvers incur a higher

computational cost due to larger stage counts. While the observed patterns are consistent with the proposed explanation, the geometric analysis should be interpreted as empirical evidence rather than definitive proof of causality. We do not claim that these effects hold for all Neural ODE architectures, adaptive solvers, datasets, or adversarial settings; rather, our results identify a controlled setting in which solver choice measurably affects learned representation dynamics. Non-uniform degradation under solver swap also suggests that compatibility between learned representations may depend not only on stability-region overlap, but also on the alignment between the truncated B-series expansions induced by the respective solvers; a direct tree-level coefficient analysis is left to future work.

From a practical perspective, these findings suggest that solver choice may need to be treated as a model-design decision rather than only as a post-training implementation choice. Because the operating fraction $\phi = h\sigma/R$ depends on the learned spectral regime, it is best interpreted diagnostically rather than as a fully prescriptive design rule. A practical workflow is therefore to train under a fixed horizon, measure the resulting spectral scale σ, compute the corresponding operating fraction, and use this diagnostic to assess whether the model can be moved toward a more favourable regime, for example by changing the stability radius, stage count, or stability-polynomial shape.

More broadly, these findings support treating numerical integration as an additional design dimension in continuous-depth models, alongside architecture and optimization. This perspective extends naturally to related settings such as continuous normalizing flows, diffusion models, and neural-controlled differential equations, where numerical integration plays a central role.

References

Assyr Abdulle. Fourth order chebyshev methods with recurrence relation. *SIAM Journal on Scientific Computing*, 23(6):2041–2054, 2002.

Sinjini Banerjee, Tim Marrinan, Reilly Cannon, Tony Chiang, and Anand D Sarwate. Measuring training variability from stochastic optimization using robust nonparametric testing. *IEEE Journal of Selected Topics in Signal Processing*, 2025.

Louis Béthune. *Deep learning with Lipschitz constraints*. PhD thesis, Université de Toulouse, 2024.

John Charles Butcher. *Numerical methods for ordinary differential equations*. John Wiley & Sons, 2016.

Mark H. Carpenter and Christopher A. Kennedy. Fourth-order 2n-storage runge-kutta schemes. Technical report, NASA TM, 1994.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pp. 2206–2216. PMLR, 2020.

Chris Finlay, Jörn-Henrik Jacobsen, Levon Nurbekyan, and Adam Oberman. How to train your neural ode: the world of jacobian and kinetic regularization. In *International conference on machine learning*, pp. 3154–3164. PMLR, 2020.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Eldad Haber and Lars Ruthotto. Stable architectures for deep neural networks. *Inverse problems*, 34(1): 014004, 2018.

Ernst Hairer, Syvert P. Nørsett, and Gerhard Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer, 1993.

Patrick Kidger. On neural differential equations. *arXiv preprint arXiv:2202.02435*, 2022.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images.(2009), 2009.

Yann Le, Xuan Yang, et al. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Stefano Massaroli, Michael Poli, Jinkyoo Park, Atsushi Yamashita, and Hajime Asama. Dissecting neural odes. *Advances in Neural Information Processing Systems*, 33, 2020.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Robustness via curvature regularization, and vice versa. *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Stephen O'Sullivan. A class of high-order runge–kutta–chebyshev stability polynomials. *Journal of Computational Physics*, 300:665–678, 2015. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2015.07.050. URL `https://www.sciencedirect.com/science/article/pii/S0021999115005033`.

Stephen O'Sullivan. Runge–kutta–gegenbauer explicit methods for advection-diffusion problems. *Journal of Computational Physics*, 388:209–223, 2019. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2019.03.001. URL `https://www.sciencedirect.com/science/article/pii/S0021999119301706`.

Shangshu Qian, Viet Hung Pham, Thibaud Lutellier, Zeou Hu, Jungwon Kim, Lin Tan, Yaoliang Yu, Jiahao Chen, and Sameena Shah. Are my deep learning systems fair? an empirical study of fixed-seed training. *Advances in Neural Information Processing Systems*, 34:30211–30227, 2021.

Lars Ruthotto and Eldad Haber. Deep neural networks motivated by partial differential equations. *Journal of Mathematical Imaging and Vision*, 62:352–364, 2020.

John H. Williamson. Low-storage runge-kutta schemes. *Journal of Computational Physics*, 1980.

Juntang Zhuang, Nicha Dvornek, Sekhar Tatikonda, Xenophon Papademetris, and James S. Duncan. Adaptive checkpoint adjoint method for gradient estimation in neural ode. *International Conference on Machine Learning*, 2020.

A First-Order Neural ODE Control

The main experiments use a third-order Neural ODE formulation to expose higher-order trajectory geometry directly. To verify that the observed solver-dependent effects are not merely artefacts of this state augmentation, we include two first-order controls. The first is a lightweight rooted-tree/JVP diagnostic trained for eight epochs. The second is a full-training frozen-weight solver-swap experiment.

The two controls have different purposes and therefore different absolute accuracy levels. The rooted-tree diagnostic is intended only as a qualitative early-training check, whereas the solver-swap experiment is trained to convergence. Thus, the lower accuracies in Table 15 and the higher accuracies in Table 16 reflect different training budgets rather than inconsistent methodology.

A.1 Lightweight First-Order Rooted-Tree Diagnostic

We first evaluate a lightweight first-order Neural ODE control on CIFAR-10 at $\Delta T = 10$ using RK4 and ESRKG-15. The model is trained for eight epochs under identical optimization settings using both solvers. We then evaluate rooted-tree and Jacobian–vector-product diagnostics derived from the learned vector field:

$$F_1 = f(z), \qquad F_2 = Jf(z)[F_1], \qquad F_{31} = Jf(z)[F_2].$$

A lightweight first-order higher-order trajectory norm proxy is defined as

$$\kappa_{\mathrm{FO}} = \frac{\|F_{31}\|}{\|F_1\| + \epsilon},$$

together with a spectral-scale proxy estimated from Jacobian–vector-product growth.

Across all seeds, ESRKG-15 produces larger rooted-tree/JVP geometric quantities, larger higher-order trajectory norm proxies, and larger spectral-scale proxies than RK4, despite broadly comparable clean accuracy. This indicates that solver-dependent learned geometry is already visible in a standard first-order Neural ODE, although this diagnostic is not intended as a strict accuracy-matched comparison.

A.2 Frozen-Weight First-Order Solver-Swap Control

We next evaluate whether solver specialization appears in a fully trained standard first-order Neural ODE. The model is trained independently with RK4, ESRKG-15, and LSRKG-15 on CIFAR-10 at $\Delta T = 10$. After training, the learned weights are frozen and only the numerical solver is replaced at evaluation time.

RK4 is evaluated using four substeps of size $h = 2.5$, while ESRKG-15 and LSRKG-15 use a single step of size $h = 10$. Thus, all solvers integrate over the same horizon $\Delta T = 10$, while preserving their natural operating modes.

Despite closely matched native accuracies across the three training solvers (RK4: 80.11%, LSRKG-15: 80.28%, ESRKG-15: 80.35%), frozen-weight solver replacement causes substantial degradation. For example, the LSRKG-15-trained checkpoint drops from 80.28% under native evaluation to 59.19% under RK4, while the ESRKG-15-trained checkpoint drops from 80.35% to 65.58% under RK4.

The same-polynomial realization comparison is particularly notable. In this implementation, ESRKG-15 and LSRKG-15 have nearly identical external stability-polynomial However, swapping between their realizations is not functionally neutral: LSRKG-15 evaluated with ESRKG-15 drops to 69.44%, and ESRKG-15 evaluated with LSRKG-15 drops to 31.17%. This indicates that solver specialization is not explained solely by the external stability polynomial. Internal stage realization can also affect the learned checkpoint.

Overall, these first-order controls support the main conclusion of the paper: solver replacement is not functionally neutral after training. The third-order formulation used in the main experiments provides direct access to higher-order geometric diagnostics, but solver specialization already appears in a standard first-order Neural ODE.

Table 15: Lightweight first-order Neural ODE rooted-tree control on CIFAR-10 at $\Delta T = 10$. The model is trained for eight epochs. ESRKG-15 consistently produces larger rooted-tree geometric quantities and larger first-order higher-order trajectory norm proxies than RK4 across all seeds.

Solver	Seed	Acc.	F_2	F_{31}	κ_{FO}	σ
RK4	0	37.9	6.71	0.87	0.00560	0.0432
ESRKG-15	0	39.7	8.53	1.22	0.00780	0.0548
RK4	1	35.9	5.89	0.71	0.00457	0.0380
ESRKG-15	1	36.3	6.46	0.83	0.00547	0.0424
RK4	2	38.4	4.85	0.58	0.00403	0.0336
ESRKG-15	2	39.6	5.47	0.69	0.00469	0.0372

Table 16: First-order Neural ODE frozen-weight solver-swap control on CIFAR-10 at $\Delta T = 10$. Rows indicate the solver used during training, while columns indicate the solver used at evaluation. Diagonal entries correspond to native evaluation. Off-diagonal entries evaluate the same frozen checkpoint after replacing only the numerical integrator.

Train \ Eval	RK4	ESRKG-15	LSRKG-15
RK4	**80.11**	69.06	41.61
LSRKG-15	59.19	69.44	**80.28**
ESRKG-15	65.58	**80.35**	31.17