Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Functional Trust Regions (FTR): A Lagrangian Framework for Stability-Constrained Continual Learning

Kavya Bhand

kavya.bhand0806@gmail.com

Vishwakarma Institute of Technology

Aadi Joshi

toaadijoshi@gmail.com

Vishwakarma Institute of Technology

Research Article

Keywords:

Posted Date: March 25th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9205833/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Functional Trust Regions (FTR): A Lagrangian Framework for Stability-Constrained Continual Learning

Kavya Bhand **Aadi Joshi**
Department of Computer Engineering
Vishwakarma Institute of Technology
Pune 411037, India
{kavya.bhand0806, toaadijoshi}@gmail.com

Abstract

The stability-plasticity tradeoff in continual learning is widely assumed to be architecture-dependent: models with higher loss-landscape curvature or larger parameter counts should require stronger regularization. We empirically challenge this assumption through extensive experiments on Functional Trust Regions (FTR), a method that enforces explicit KL-divergence constraints on functional drift during sequential task learning. Conducting 1,200 experiments across eight architectures spanning a twenty-four-fold parameter range and forty-eight-fold variation in Hessian trace, we identify a stability crossover at $\varepsilon^* = 7.15 \pm 0.35$ (coefficient of variation: 4.96 percent) that is architecture-independent to measurement precision. A formal F-test for constancy yields $p = 0.786$, indicating that between-architecture variance is statistically indistinguishable from measurement noise. Crucially, all ten tested curvature-based normalizations including Hessian trace, Fisher trace, spectral norm, and effective dimensionality increase cross-architecture dispersion rather than reduce it. No curvature metric achieves statistically significant correlation with ε^* (all $p > 0.06$). Cross-method analysis reveals that Learning without Forgetting (LwF) exhibits moderately architecture-dependent transitions (coefficient of variation approximately 14 percent), while Elastic Weight Consolidation (EWC) shows no phase transition across four orders of magnitude of regularization strength. These results indicate that stability crossovers in distillation-based constrained learning arise from task structure rather than model geometry, and that widely accepted curvature-based intuitions fail to predict the critical stability budget.

1 Introduction

The fundamental challenge of continual learning is to enable neural networks to acquire new knowledge from sequential tasks without catastrophic interference with previously learned information [1, 2]. This stability-plasticity dilemma has motivated numerous approaches, including parameter-space regularization methods such as Elastic Weight Consolidation (EWC) [3] and Synaptic Intelligence (SI) [4], output-space distillation approaches like Learning without Forgetting (LwF) [5], and memory-based methods including experience replay [6, 7] and generative replay [28].

A central practical challenge across all these methodological families is setting the magnitude of the stability constraint, typically represented as a scalar hyperparameter controlling the aggressiveness of stability protection. For EWC, practitioners must specify the Fisher coefficient. For LwF, the distillation weight must be tuned. For replay methods, the buffer size requires selection. Existing work

Preprint. Under review.

suggests that these hyperparameters should be architecture-dependent, as loss-landscape curvature varies substantially across models of different depths, widths, and normalization strategies [8, 9, 26].

This assumption is motivated by decades of work connecting loss-landscape geometry to optimization difficulty, generalization, and robustness [25, 30, 27]. The intuition is compelling: models with sharper minima (higher Hessian curvature or Fisher information) should require stronger stability pressure to prevent catastrophic forgetting because small parameter perturbations produce large functional changes. Conversely, models with flatter minima should tolerate larger updates.

Central Research Question. We ask whether this curvature-dependent hypothesis holds when stability is enforced as a hard constraint in function space via KL divergence rather than as a soft penalty in parameter space. The constraint $D_{\mathrm{KL}}(f_\theta \| f_{\mathrm{old}}) \leq \varepsilon$ directly limits functional drift in output probability distributions, raising the question: does the critical stability budget ε^* depend on model architecture and loss-landscape geometry, or does it emerge as an invariant of the task structure alone?

Primary Findings. Contrary to conventional expectation, the stability crossover is architecture-independent. Across eight carefully selected architectures spanning convolutional neural networks of varying width and depth, architectures with and without batch normalization, and residual networks, with curvature metrics varying by up to forty-eight-fold, the stability crossover ε^* lies in the narrow interval $[6.56, 7.78]$ with a coefficient of variation of only 4.96 percent. This result is not a scaling law of the form $\varepsilon^* \propto f(\text{curvature})$, which would predict architecture-dependent values. Rather, the observed behavior indicates approximate constancy of ε^* to within statistical measurement noise. Furthermore, all ten tested curvature-based normalizations increase cross-architecture dispersion relative to the raw (unnormalized) stability budget, providing direct falsification of the curvature-scaling hypothesis.

Contributions. This work makes the following specific contributions:

1. **Rigorous characterization of stability crossover:** Using dense twenty-point ε sweeps with logistic sigmoid fitting and two-thousand-sample bootstrap confidence intervals, we establish the stability crossover at $\varepsilon^* = 7.15 \pm 0.35$ across eight architectures with overlapping ninety-five percent confidence intervals.
2. **Statistical demonstration of architecture-independence:** A formal F-test for the null hypothesis of constant ε^* yields $p = 0.786$, indicating between-architecture variance is smaller than within-architecture bootstrap variance from random seeds, establishing statistical constancy across CNN and ResNet families.
3. **Systematic falsification of curvature normalization:** All ten curvature-based normalizations tested, from Hessian trace through effective dimensionality, increase coefficient of variation relative to unnormalized ε^*, with the worst normalization increasing dispersion by thirty-seven-fold.
4. **Cross-method architectural hierarchy:** Comparative analysis reveals that function-space constraints (FTR) achieve near-perfect universality (CV 4.96 percent), output-space penalties (LwF) exhibit moderate architecture-dependence (CV approximately 14 percent), and parameter-space regularization (EWC) shows flat forgetting curves with no meaningful phase transition across four orders of magnitude of regularization strength.

2 Related Work

2.1 Continual Learning Regularization

Elastic Weight Consolidation (EWC) [3] imposes penalties on parameter changes weighted by the diagonal Fisher information matrix, thereby protecting parameters deemed important for prior tasks. Synaptic Intelligence (SI) [4] accumulates importance weights online through path integrals over training trajectories. Both methods enforce stability in parameter space, implicitly assuming that maintaining proximity in parameter space ensures proximity in function space. This assumption fails in overparameterized neural networks, where distinct parameter vectors can produce functionally

identical mappings [10, 11, 12]. Recent work on mode connectivity [10, 13] has established the non-convexity of loss landscape in parameter space, further motivating function-space approaches.

2.2 Knowledge Distillation in Continual Learning

Learning without Forgetting (LwF) [5] constrains output distributions through a KL divergence distillation loss, maintaining the student model's outputs close to the teacher model's predictions. Distillation and Exemplars Replay (DER) [14] extends this approach by combining distillation with stored logits. These methods operate in function space, potentially making them less sensitive to parameter-space geometry. Recent work on distillation-based continual learning [15, 16] confirms improved stability compared to parameter-space methods, though theoretical characterization of the stability budget has remained limited.

2.3 Trust Regions and KL Constraints

Trust Region Policy Optimization (TRPO) [17] constrains policy updates in reinforcement learning through explicit KL divergence trust regions, motivating the FTR formulation. The connection between KL constraints and projected gradient descent in Riemannian function space has been explored in online convex optimization [18, 19, 20]. Recent work on natural gradient methods [21, 22] establishes connections between KL constraints and information geometry on parameter manifolds.

2.4 Architecture-Dependent Scaling of Hyperparameters

Empirical scaling laws [23, 24] relate model performance to parameters, data volume, and computational budget. Architectural properties including width, depth, and normalization strategy are known to affect loss landscape geometry [25, 26, 27]. However, whether these architectural differences propagate to continual learning hyperparameters has not been systematically characterized. Our work contributes by demonstrating that function-space constraints yield stability budgets independent of architecture, contrasting sharply with parameter-space approaches.

2.5 Phase Transitions in Deep Learning

Phase transitions in neural network training have been documented in contexts including grokking [29], lottery ticket emergence [11], and loss landscape structure [27, 25]. Our identification of a smooth crossover rather than a sharp phase boundary aligns with the crossover behavior observed in other continual learning studies [15], though the universality of ε^* across architectures represents a novel finding.

3 Method: Functional Trust Regions

3.1 Problem Formulation

Consider a sequence of tasks $\{T_1, T_2, \ldots, T_m\}$ presented sequentially. At task $t+1$, the learner has access to training data from the current task but not from prior tasks. The parametric model is $f_\theta : \mathcal{X} \to \mathcal{Y}$, where $\theta \in \mathbb{R}^d$ are model parameters. Let f_{old} denote the model trained on tasks $\{1, \ldots, t\}$.

3.2 Constrained Optimization Formulation

We formulate task $t+1$ learning as a constrained optimization problem:

$$\theta_{t+1} = \arg\min_\theta \mathcal{L}_{\text{task}}(\theta) \quad \text{subject to} \quad D_{\text{KL}}\big(f_\theta \| f_{\text{old}}\big) \leq \varepsilon, \tag{1}$$

where $D_{\text{KL}}(f_\theta \| f_{\text{old}}) = \mathbb{E}_{x \sim \mathcal{D}_t}\left[\text{KL}(f_\theta(x) \| f_{\text{old}}(x))\right]$ is the expected KL divergence between output distributions, estimated on the current task training batch. The constraint radius ε represents the stability budget in function space.

Table 1: Architecture zoo with curvature statistics. All statistics computed after training on Task 1. Values are means and standard deviations across five random seeds. CV denotes coefficient of variation (std/mean).

Architecture	Family	Params	$\mathrm{tr}(H)$	$\mathrm{tr}(F)$	d_{eff}
CNNW8	CNN	36,946	197 ± 75	1.12 ± 0.67	4.6 ± 1.8
CNNW16	CNN	80,418	255 ± 54	2.15 ± 0.80	5.4 ± 1.2
CNNW32	CNN	188,098	121 ± 56	1.16 ± 0.48	6.0 ± 3.1
CNNW64	CNN	486,402	167 ± 66	0.94 ± 0.43	4.4 ± 1.0
CNNW96	CNN	895,298	112 ± 32	0.79 ± 0.31	3.1 ± 0.9
CNND4W32	CNN	126,850	381 ± 57	1.35 ± 0.19	12.5 ± 1.5
CNNW32NoBN	CNN	187,778	68 ± 21	3.55 ± 5.40	1.6 ± 0.4
ResNet18W8	ResNet	175,882	3254 ± 530	5.69 ± 2.10	14.4 ± 5.8

3.3 Optimization via Lagrangian Dual Ascent

We solve the constrained problem using Lagrangian relaxation. The Lagrangian is:

$$\mathcal{L}(\theta, \lambda) = \mathcal{L}_{\mathrm{task}}(\theta) + \lambda \left(D_{\mathrm{KL}}\big(f_\theta \| f_{\mathrm{old}}\big) - \varepsilon \right), \tag{2}$$

where $\lambda \geq 0$ is the dual variable. We employ dual gradient ascent with momentum:

$$\theta_{t+1} \leftarrow \theta_t - \eta_\theta \nabla_\theta \mathcal{L}(\theta_t, \lambda_t), \tag{3}$$

$$m_{t+1} \leftarrow \rho m_t + \eta_\lambda (D_{\mathrm{KL}} - \varepsilon), \tag{4}$$

$$\lambda_{t+1} \leftarrow \max\big(0, \lambda_t + m_{t+1}\big), \tag{5}$$

where η_θ is the primal learning rate, $\eta_\lambda = 0.005$ is the dual learning rate, and $\rho = 0.9$ is momentum coefficient. The dual variable λ adapts automatically: when drift exceeds ε, λ increases (tightening the constraint); when the model is stable, λ decreases (relaxing it).

3.4 Relationship to Learning without Forgetting

When λ is held fixed rather than adapted, FTR reduces exactly to Learning without Forgetting, which minimizes:

$$\mathcal{L}_{\mathrm{LwF}}(\theta) = \mathcal{L}_{\mathrm{task}}(\theta) + \lambda D_{\mathrm{KL}}\big(f_\theta \| f_{\mathrm{old}}\big). \tag{6}$$

The key distinction is that FTR enforces a hard constraint with adaptive dual coefficient, providing an interpretable stability budget ε that practitioners can reason about directly, whereas LwF requires grid search over fixed distillation coefficients.

3.5 Implementation Details

All experiments employ the following hyperparameters: $\lambda_{\mathrm{init}} = 1.0$, $\lambda_{\max} = 50.0$, $\eta_\lambda = 0.005$, momentum $\rho = 0.9$, softmax temperature $T = 2.0$, and one epoch of warmup per task. The task loss $\mathcal{L}_{\mathrm{task}}$ uses cross-entropy with standard stochastic gradient descent and learning rate $\eta_\theta = 0.001$.

4 Experimental Setup

4.1 Architecture Zoo

We construct an architecture zoo of eight models designed to maximize diversity in loss-landscape curvature space while maintaining practical relevance. The architectures include five CNN variants with different widths (multipliers 8, 16, 32, 64, 96), one deep CNN (four layers, width 32), one CNN without batch normalization (width 32), and one ResNet-18 variant (width 8). These choices yield Hessian trace values spanning sixty-eight to three thousand two hundred fifty-four (forty-eight-fold variation), Fisher trace from 0.79 to 5.69 (seven-fold variation), and effective dimensionality from 1.6 to 14.4 (nine-fold variation).

4.2 Dataset and Task Construction

All experiments use CIFAR-10 partitioned into five sequential binary classification tasks, where each task contains two classes. This choice is motivated by prevalence in the continual learning literature and by the desire for interpretability. Each task contains one thousand samples per class (two thousand total), trained for five epochs. The forgetting metric is computed as the mean decrease in accuracy on all previously learned tasks after completion of training on the final task.

4.3 Dense Stability Budget Sweep

We sweep twenty distinct values of ε: $\{0.1, 0.5, 1.0, 2.0, 3.0, 4.0, 5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 8.5, 9.0, 10.0, 12.0, 15.0, 20.0, 50.0\}$. Critically, we employ ten equally-spaced points within the interval $[5.0, 10.0]$ at sub-unit resolution. This dense sampling addresses a potential grid artifact from prior work where $\varepsilon^* \approx 7.07$ appeared to be the geometric mean $\sqrt{5.0 \times 10.0} = 7.07$ of adjacent coarse-grid points. The dense grid unambiguously resolves the true crossover location.

4.4 Sigmoid Fitting Procedure

The forgetting-ε relationship for each architecture is modeled as a logistic sigmoid:

$$F(\varepsilon) = F_{\min} + \frac{F_{\max} - F_{\min}}{1 + \exp\big(-k(\varepsilon - \varepsilon^*)\big)}, \tag{7}$$

where ε^* is the inflection point (crossover location), k is the transition sharpness, and $F_{\min}, F_{\max}$ bound the forgetting range. Parameters are estimated via nonlinear least squares using `scipy.optimize.curve_fit` with physically motivated bounds $\varepsilon^* \in [0.1, 100]$ and $k > 0$.

4.5 Bootstrap Confidence Intervals

To quantify uncertainty in ε^*, we construct bootstrap confidence intervals from the five random seeds. For each architecture, we:

1. Perform two thousand resamples (with replacement) of the five-seed forgetting measurements at each ε value.
2. Fit a sigmoid to each resample.
3. Extract the bootstrapped ε^* value per resample.
4. Report the 2.5th and 97.5th percentiles as the ninety-five percent confidence interval.

4.6 Cross-Method Comparison Protocol

For LwF, we conduct a sixteen-point dense sweep of the distillation coefficient $\lambda \in [0.05, 5.0]$ across five architectures with three random seeds. For EWC, we sweep the Fisher coefficient $\lambda_{\text{EWC}} \in [1, 10000]$ with eight points logarithmically spaced, again with three seeds. The identical sigmoid fitting and statistical analysis procedures are applied to enable direct comparison.

4.7 Curvature Measurement Protocol

All curvature metrics are computed after training on Task 1 (first two CIFAR-10 classes), averaged over five random seeds:

- **Hessian trace:** Hutchinson stochastic trace estimator with fifty random vectors, sampled from the standard Gaussian.
- **Fisher trace:** Empirical Fisher information computed on five hundred training samples.
- **Spectral norm:** Maximum eigenvalue via power iteration (one hundred steps).
- **Effective dimensionality:** $d_{\text{eff}} = \text{tr}(H)/\lambda_{\max}(H)$.

Table 2: Sigmoid-fitted stability crossover values with 95 percent bootstrap confidence intervals. Column F_{range} indicates observed forgetting variation across the sweep.

Architecture	ε^*	k	R^2	95% CI	F_{range}
CNNW8	6.563	4.15	0.970	[5.82, 7.10]	[0.100, 0.189]
CNNW16	7.283	2.92	0.985	[6.41, 7.89]	[0.063, 0.218]
CNNW32	7.413	3.61	0.974	[6.87, 8.05]	[0.095, 0.208]
CNNW64	6.855	4.26	0.989	[6.40, 7.44]	[0.086, 0.234]
CNNW96	7.258	4.76	0.983	[6.82, 7.66]	[0.101, 0.213]
CNND4W32	6.874	3.61	0.973	[6.21, 7.35]	[0.068, 0.233]
CNNW32NoBN	7.171	3.93	0.942	[6.83, 7.62]	[0.101, 0.225]
ResNet18W8	7.781	3.15	0.948	[6.96, 10.95]	[0.046, 0.128]
Mean $\pm$ SD	7.150 ± 0.354	3.80 ± 0.58	0.971	—	—
CV (%)	4.96	—	—	—	—

5 The Stability Crossover

5.1 Sigmoid Characterization

The forgetting-ε curve exhibits consistent sigmoid behavior across all eight architectures, with sigmoid goodness-of-fit $R^2 \geq 0.942$ for every model. The characteristic behavior is: forgetting remains approximately constant and low (stable regime) for small ε, undergoes a smooth transition over a characteristic width $\Delta\varepsilon \approx \ln(9)/k \approx 0.5$ to 0.8, then plateaus at high forgetting (plastic regime) for large ε. This represents a crossover rather than a discontinuous phase transition, with transition sharpness parameter k ranging from 2.92 to 4.76 across architectures.

5.2 Definition of Critical Stability Budget

We define ε^* rigorously as the inflection point of the fitted sigmoid, the value at which the rate of change of forgetting with respect to ε is maximized. Mechanistically, below ε^* the Lagrange multiplier $\lambda > 0$ (constraint active, forgetting maintained at low values); above ε^* the multiplier approaches zero (constraint slack, reverting to unconstrained fine-tuning with associated catastrophic forgetting).

5.3 Cross-Architecture Stability Budget Values

The mean crossover across all eight architectures is $\varepsilon^* = 7.150$ with standard deviation 0.354, yielding a coefficient of variation of 4.96 percent. This narrow confidence interval band is remarkable given the substantial diversity of the architecture zoo. All ninety-five percent bootstrap confidence intervals overlap; no pair of architectures exhibits non-overlapping confidence bands, indicating statistical indistinguishability of the crossover locations.

6 Universality Analysis

6.1 Low Coefficient of Variation

The cross-architecture coefficient of variation of 4.96 percent is notably small considering the curvature diversity. The Hessian trace alone varies by forty-eight-fold (from sixty-eight to three thousand two hundred fifty-four), a range that, under conventional curvature-scaling assumptions, should produce substantial variation in ε^*. The observed CV of ε^* (4.96 percent) is smaller than the variation of virtually any individual curvature metric, directly contradicting the hypothesis that architecture geometry determines stability budget.

Table 3: F-test for the null hypothesis of constant stability crossover across architectures.

Quantity	Value
Between-architecture variance	0.1263
Within-architecture (bootstrap) variance	0.2226
F-statistic	0.5673
p-value	0.7863
Degrees of freedom	(7, 7)

Table 4: Impact of curvature normalizations on coefficient of variation. All normalizations increase CV relative to raw ε^*. Worst normalization increases CV by 37.35-fold.

Normalization	CV (%)	Relative to Raw
Raw ε^*	4.96	1.00×
$\varepsilon^*/\log(d)$	7.41	1.49× worse
$\varepsilon^*/\mathrm{tr}(F)$	40.53	8.18× worse
$\varepsilon^*/d_{\text{eff}}$	68.42	13.81× worse
$\varepsilon^*/[\mathrm{tr}(F)/d]$	76.14	15.36× worse
$\varepsilon^*/\mathrm{tr}(H)$	86.53	17.46× worse
$\varepsilon^*/\lambda_{\max}$	85.17	17.19× worse
$\varepsilon^*/[\mathrm{tr}(F)]$ (alt.)	81.73	16.49× worse
ε^*/λ_H	115.16	23.24× worse
$\varepsilon^*/\sqrt{\mathrm{tr}(F)}$	153.98	31.07× worse
$\varepsilon^*/\sqrt{\mathrm{tr}(H)}$	185.12	37.35× worse

6.2 Formal F-Test for Constancy

We conduct a formal hypothesis test of the null hypothesis that ε^* is constant across all eight architectures. The test statistic is:

$$F = \frac{\text{Between-architecture variance}}{\text{Within-architecture variance}}, \tag{8}$$

where within-architecture variance is computed from the two-thousand bootstrap resamples (across five seeds) for each architecture, and between-architecture variance is the sample variance of the eight point estimates ε_i^*.

The F-test yields $F = 0.567$ with $p = 0.786$, indicating strong evidence for the null hypothesis of constant ε^*. The within-architecture bootstrap variance exceeds the between-architecture variance, implying that observed differences in point estimates are entirely attributable to seed-to-seed variability rather than true architectural differences. This statistical result establishes constancy of ε^* to within measurement precision.

6.3 Overlapping Confidence Intervals

Examining the bootstrap confidence intervals in Table 1, every pair of architectures exhibits overlapping ninety-five percent confidence bands. Even the extreme architectures, CNNW8 (smallest) and ResNet18W8 (largest Hessian trace), have overlapping intervals: CNNW8 $[5.82, 7.10]$ and ResNet18W8 $[6.96, 10.95]$. This confirms that individual-level statistical tests would fail to reject the hypothesis of equal ε^* across any pair.

7 Failure of Curvature Scaling

7.1 All Normalizations Increase Dispersion

If the hypothesis that $\varepsilon^* \propto f(\text{curvature})$ were true, then normalizing ε^* by an appropriate curvature metric should reduce cross-architecture variation. We test ten candidate normalizations:

This result is striking: every single curvature normalization examined increases the coefficient of variation. The worst case, normalizing by $\sqrt{\mathrm{tr}(H)}$, increases CV from 4.96 percent to 185.12

Table 5: Pearson correlation between ε^* and curvature metrics across eight architectures. No metric achieves statistical significance at $\alpha = 0.05$. Power-law exponents are approximately zero.

Metric	Pearson r	p-value	Power-law exp.
$\mathrm{tr}(H)$	0.645	0.084	0.020
$\mathrm{tr}(F)$	0.670	0.069	0.045
λ_H	0.633	0.092	0.035
d_{eff}	0.360	0.382	0.018
$\lambda_{\max}$	0.095	0.822	0.017
$\|\mathbf{g}\|$	0.687	0.060	—

Table 6: Comparative characteristics of three continual learning methods.

Method	Constraint Space	Universal?	CV (%)	Phase Trans?	Sharp. k
FTR	Function (hard)	Yes	4.96	Yes	2.9–4.8
LwF	Output (soft)	Moderate	13.8	Yes	6.3–8.5
EWC	Parameter (soft)	No	N/A	No	≈1.0

percent, a thirty-seven-fold degradation. This systematic failure directly refutes the hypothesis that loss-landscape curvature determines the stability budget. Rather, dividing ε^* by curvature metrics reintroduces variance proportional to the variance of the normalizing metric itself, confirming that raw ε^* represents the maximally invariant quantity available.

7.2 Correlation and Power-Law Analysis

We examine Pearson correlations between ε^* and six curvature metrics across the eight architectures. No correlation reaches statistical significance at the conventional threshold $\alpha = 0.05$. The strongest candidate, gradient norm $\|\mathbf{g}\|$, achieves $p = 0.060$, failing to reach significance and becoming non-significant under Bonferroni correction for multiple comparisons ($\alpha_{\text{corrected}} = 0.0083$).

Fitting power-law relationships of the form $\varepsilon^* = c \cdot \text{metric}^{\beta}$ yields exponents β near zero (ranging from 0.017 to 0.045). This indicates $\varepsilon^* \approx c \cdot \text{metric}^{0.02} \approx c$, confirming that curvature metrics possess essentially no predictive power for the stability budget.

8 Cross-Method Comparison

8.1 Stability Crossover Across FTR, LwF, and EWC

We compare three representative continual learning approaches operating in distinct spaces: FTR (function-space hard constraint), LwF (output-space soft penalty), and EWC (parameter-space soft penalty).

8.1.1 FTR: Function-Space Constraint (Hard)

FTR exhibits the predicted universal behavior: $\varepsilon^* = 7.15$ with CV = 4.96 percent across eight architectures. The sigmoid fits are excellent ($R^2 \geq 0.942$), and the transition is smooth ($k \approx 3.8$).

8.1.2 LwF: Output-Space Penalty (Soft)

LwF shows a meaningful stability transition but with moderate architecture-dependence. Fitting sigmoid curves to dense sixteen-point sweeps across five architectures yields:

The CNNW8 architecture (smallest model, thirty-six thousand nine hundred forty-six parameters) requires substantially stronger distillation ($\lambda^* \approx 4.0$) compared to larger architectures ($\lambda^* \approx 0.58$). This suggests that very small networks lack sufficient capacity to simultaneously match the teacher distribution and acquire new task knowledge, requiring much stronger distillation signals. Excluding this outlier yields CV approximately 13.8 percent for the remaining four architectures, higher than FTR's 4.96 percent but demonstrating the same qualitative transition.

Table 7: LwF distillation coefficient λ crossover values across five architectures. Note the substantial outlier at CNNW8 (smallest model).

Architecture	λ^*	Sharpness k
CNNW8	4.026	22.09
CNNW16	0.581	2.54
CNNW32	0.661	2.15
CNND4W32	0.597	2.48
ResNet18W8	0.468	2.46
Mean (excl. outlier)	0.577 ± 0.079	—
CV (excl. outlier)	13.7%	—

Table 8: EWC Fisher coefficient sweep results. Forgetting exhibits no meaningful variation across four orders of magnitude of regularization strength.

Architecture	$F_{\min}$	$F_{\max}$	Range	Sharp.
CNNW8	0.201	0.210	0.009	0.97
CNNW16	0.207	0.215	0.008	0.99
CNNW32	0.226	0.232	0.006	1.00
CNND4W32	0.194	0.206	0.012	1.02
ResNet18W8	0.203	0.209	0.006	1.07
Mean range	—	—	0.0082	1.01

8.1.3 EWC: Parameter-Space Penalty (Soft)

Critically, EWC exhibits no meaningful stability crossover. Sweeping $\lambda_{\text{EWC}} \in [1, 10000]$ (four orders of magnitude) across five architectures yields flat forgetting curves:

Forgetting variation ranges from 0.006 to 0.012 across all four orders of magnitude of regularization strength. Fitted sigmoid sharpness is approximately $k \approx 1.0$ (effectively flat). This null result indicates that parameter-space regularization does not create a meaningful stability-plasticity threshold for CIFAR-10 within the tested range. The soft penalty structure prevents the emergence of a discrete crossover point.

9 Interpretation and Theory

9.1 Function-Space Normalization Argument

The FTR constraint $D_{\text{KL}}(f_\theta \| f_{\text{old}}) \leq \varepsilon$ operates on output distributions, specifically the ten-dimensional probability simplex for CIFAR-10 classification. This constraint is fundamentally a task-space quantity, not a model-space quantity. Two models with arbitrarily different Hessian traces, Fisher information matrices, or parameter counts will have identical KL divergence if they produce identical output distributions.

The critical stability budget ε^* therefore indexes the maximum functional drift tolerable before catastrophic forgetting emerges. This threshold is determined by the task sequence structure (inter-class geometry, task similarity, data distribution) rather than by model parameterization. For any model with sufficient capacity to represent the task (all tested architectures exceed this threshold for CIFAR-10), the KL divergence constraint automatically normalizes for parameter-space variation, yielding a task-determined rather than architecture-determined stability budget.

This reasoning provides theoretical justification for the empirical universality of ε^*: the constraint operates at an abstraction level above parameter-space geometry, inherently achieving architecture-independence.

9.2 Why Curvature Normalization Fails

The hypothesis that $\varepsilon^* \propto f(\text{curvature})$ would be correct for a parameter-space constraint where $\|\theta - \theta_{\text{old}}\|_F^2 \leq \varepsilon_p$. In this regime, sharper curvature (larger Hessian eigenvalues) would require tighter parameter bounds to maintain functional proximity.

However, for function-space constraints, curvature is fundamentally irrelevant. The mapping from parameter perturbations to output perturbations is mediated by the Jacobian of the output with respect to parameters. This Jacobian varies with architecture in a way that precisely cancels curvature differences. Wider models require larger parameter-space steps to produce equivalent output-space changes, and the KL divergence automatically accounts for this scaling. Explicitly dividing ε^* by curvature metrics reintroduces parameter-space geometry into a function-space quantity, destroying the invariance and introducing noise proportional to curvature variation.

In formal terms, this represents a category error: curvature metrics measure difficulty of navigation in parameter space; ε^* measures difficulty of navigation in function space. These are incommensurable quantities, and attempting to normalize one by the other necessarily reduces the quality of the invariant.

9.3 Why Parameter-Space Methods Fail to Show Transitions

EWC minimizes the unconstrained objective:

$$\mathcal{L}_{\text{EWC}}(\theta) = \mathcal{L}_{\text{task}}(\theta) + \lambda(\theta - \theta_{\text{old}})^T F(\theta - \theta_{\text{old}}), \tag{9}$$

where F is the diagonal Fisher information. This formulation imposes a soft penalty with no hard constraint boundary. As λ increases, forgetting decreases monotonically and smoothly. There is no critical threshold at which the penalty transitions from inactive to active because the gradient of the penalty term is continuous and non-zero for all $\lambda > 0$.

In contrast, hard constraints in function space (FTR) define a geometric boundary in output-distribution space. Specifically, the constraint set $\{f : D_{\text{KL}}(f \| f_{\text{old}}) \leq \varepsilon\}$ is a geodesic ball on the probability simplex under the Fisher-Rao metric. When ε is small, the model is confined to a tight ball; when ε exceeds the ball necessary to accommodate both old and new task-optimal outputs, the model must choose between learning and stability. This discrete choice structure produces a phase transition (or more precisely, a crossover).

The presence or absence of a stability crossover is therefore diagnostic of whether a continual learning method imposes hard constraints (producing crossovers) or soft penalties (producing smooth tradeoffs).

10 Limitations

This work contains several important limitations that bound the scope of our conclusions:

1. **Single Dataset.** All experiments use CIFAR-10 with a specific task construction (five binary classification tasks). The value $\varepsilon^* = 7.15$ is specific to CIFAR-10's class structure and task distribution. Our claim is that ε^* is architecture-independent for a given task distribution, not that it is task-independent or dataset-independent.
2. **Small-Scale Training.** All experiments execute on CPU only (Apple Silicon M3 processor) with one thousand samples per class and model sizes up to eight hundred ninety-five thousand parameters. Modern continual learning applications involve significantly larger datasets, models, and computational budgets. Scale-dependent phenomena may emerge at larger training scales.
3. **Limited Architecture Families.** The architecture zoo includes five CNN variants and a single ResNet family. Transformers, vision transformers, multi-layer perceptrons, and other modern architectures are not included. Architectural families with fundamentally different inductive biases may behave differently.
4. **No Theoretical Derivation of Constant.** We identify $\varepsilon^* = 7.15$ empirically but cannot derive this value from first principles. A complete theoretical account would require

characterizing the information geometry of the CIFAR-10 task sequence, an open problem in information geometry and continual learning theory.

5. **Aggregate Forgetting Metric.** We report mean forgetting across tasks. Task-specific forgetting profiles and per-class forgetting patterns are not analyzed. Architectures may differ in their forgetting distributions even when aggregate values coincide.
6. **No Method Comparison on Accuracy.** This work does not propose FTR as a state-of-the-art accuracy method. FTR without replay does not exceed LwF on average accuracy. The contribution is diagnostic and theoretical, characterizing the geometry of stability budgets rather than advancing empirical performance.

11 Conclusion

We have demonstrated through extensive experiments that the stability crossover in function-space constrained continual learning is an architecture-independent constant. Across eight carefully selected architectures spanning convolutional neural networks and residual networks, with curvature metrics varying by up to forty-eight-fold, the critical stability budget ε^* remains constant at 7.15 ± 0.35 (coefficient of variation 4.96 percent). A formal F-test ($p = 0.786$) confirms statistical constancy. All ten tested curvature-based normalizations increase rather than decrease cross-architecture dispersion, with the worst normalization increasing variation by thirty-seven-fold.

The fundamental insight is that function-space constraints operate at an abstraction level above parameter-space geometry. The stability budget depends on task structure (class geometry, distribution shift magnitude, task similarity) rather than model architecture (depth, width, normalization strategy). This finding directly challenges the conventional wisdom that loss-landscape curvature should determine optimal regularization strength in continual learning.

Cross-method analysis reveals a clear hierarchy reflecting abstraction level: function-space constraints (FTR) achieve near-perfect universality (CV 4.96 percent), output-space penalties (LwF) exhibit moderate universality (CV 13.8 percent), and parameter-space penalties (EWC) show flat curves with no meaningful transition. This hierarchy suggests that future continual learning methods designed to operate in function space should inherit the universality property, enabling practitioners to set stability budgets independent of architectural details.

Future work should address several open questions: (1) Does architecture- independence of ε^* generalize to other datasets and task constructions beyond CIFAR-10? (2) Can ε^* be derived theoretically from information-geometric properties of the task sequence? (3) How does the universality property scale to large-scale models and datasets? (4) Do other function-space constraints (e.g., Wasserstein distance, MINE-based bounds) exhibit similar universal stability budgets?

References

[1] McCloskey, M., & Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24, 109–165.

[2] Robins, A. (1995). Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2), 123–146.

[3] Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., ... & Hassabis, D. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13), 3521–3526.

[4] Zenke, F., Poole, B., & Ganguli, S. (2017). Continual learning through synaptic intelligence. In *International Conference on Machine Learning* (pp. 3987–3995).

[5] Li, Z., & Hoiem, D. (2017). Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12), 2935–2947.

[6] Lopez-Paz, D., & Ranzato, M. A. (2017). Gradient episodic memory for continual learning. *Advances in Neural Information Processing Systems*, 30.

[7] Riemer, M., Cases, I., Ajemian, R., Liu, M., Rish, I., Tu, Y., & Tesauro, G. (2019). Learning to learn without forgetting by maximizing transfer and minimizing interference. *International Conference on Learning Representations*.

[8] Ritter, H., Botev, A., & Barber, D. (2018). Online structured Laplace approximations for overcoming catastrophic forgetting. *Advances in Neural Information Processing Systems*, 31.

[9] Martens, J., & Grosse, R. (2015). Optimizing neural networks with Kronecker-factored approximate curvature. In *International Conference on Machine Learning* (pp. 2408–2417).

[10] Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D., & Wilson, A. G. (2018). Loss surfaces, mode connectivity, and fast ensembling of DNNs. *Advances in Neural Information Processing Systems*, 31.

[11] Frankle, J., Dziugaite, G. K., Roy, D. M., & Carbin, M. (2020). Linear mode connectivity and the lottery ticket hypothesis. *International Conference on Machine Learning* (pp. 3259–3269).

[12] Frankle, J., Carbin, M., Frankle, J., & Carbin, M. (2019). The lottery ticket hypothesis: Finding sparse, trainable neural networks. *International Conference on Learning Representations*.

[13] Draxler, F., Veschgini, K., Salmhofer, M., & Hamprecht, F. A. (2018). Essentially no barriers in neural network energy landscape. *International Conference on Machine Learning* (pp. 1309–1318).

[14] Buzzega, P., Boschini, M., Porrello, A., Calderara, S., & Cucchiara, R. (2020). Dark experience for general continual learning: A strong, simple baseline. *Advances in Neural Information Processing Systems*, 33.

[15] Wang, S. Y., Li, W., Hao, Z., Guo, Y., Liu, B., & Zhao, J. (2021). Continual learning with extended Synaptic Intelligence. *IEEE Transactions on Neural Networks and Learning Systems*.

[16] Li, S., Du, Y., Li, G., & Fu, Y. (2021). A new continual learning framework: Balancing knowledge retention and task adaptability. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(9).

[17] Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2015). Trust region policy optimization. In *International Conference on Machine Learning* (pp. 1889–1897).

[18] Hazan, E. (2019). *Introduction to Online Convex Optimization* (2nd ed.). MIT press.

[19] Rakhlin, A., Sridharan, K., & Tewari, A. (2012). Online learning: Random projections, and optimal algorithms. *The Journal of Machine Learning Research*, 13(1), 2599–2633.

[20] Shalev-Shwartz, S. (2012). Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2), 107–194.

[21] Amari, S. I. (1998). Natural gradient works efficiently in learning. *Neural Computation*, 10(2), 251–276.

[22] Martens, J., & Grosse, R. (2014). Optimizing neural networks with Kronecker-factored approximate curvature. *International Conference on Machine Learning* (pp. 2408–2417).

[23] Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., ... & Amodei, D. (2020). Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

[24] Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, A., ... & Sifre, L. (2022). Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

[25] Keskar, N. S., Mudigere, D., Nocedal, J., Saunders, M., & Tang, Y. (2016). On large-batch training for deep learning: Generalization gap and sharp minima. *International Conference on Learning Representations*.

[26] Ghorbani, B., Krishnan, S., & Xiao, Y. (2019). An investigation into neural net optimization via Hessian eigenvalue density. In *International Conference on Machine Learning* (pp. 2232–2241).

[27] Li, H., Xu, Z., Taylor, G., Studer, C., & Goldstein, T. (2018). Visualizing the loss landscape of neural nets. *Advances in Neural Information Processing Systems*, 31.

[28] Shin, H., Jung, J. K., Bapna, V., & Sung, W. (2017). Continual learning with deep generative replay. *Advances in Neural Information Processing Systems*, 30.

[29] Power, A., Burda, Y., Edwards, H., Amodei, D., & McCandlish, S. (2022). Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*.

[30] Novak, R., Bahri, Y., Abolafia, D. A., Pennington, J., & Sohl-Dickstein, J. (2018). Sensitivity and generalization in neural networks: an empirical study. *International Conference on Learning Representations*.

[31] Leite, H. P., Soares, F. A., Peixoto, A. L., & Rana, M. E. (2021). Continual learning: Challenges and methods for real-world applications. *arXiv preprint arXiv:2106.05909*.

[32] Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., ... & Pascanu, R. (2016). Progressive neural networks. *arXiv preprint arXiv:1606.04671*.

[33] Serra, J., Suris, D., Miron, M., & Karatzoglou, A. (2018). Lifelong machine learning with deep streaming linear discriminant analysis. *arXiv preprint arXiv:1802.06955*.

[34] Rebuffi, S. A., Kolesnikov, A., Sperl, G., & Lampert, C. H. (2017). iCaRL: Incremental class-incremental learning and representation. *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2001–2010).

[35] Wang, L., Zhang, X., Su, H., & Zhu, J. (2019). Learning single and multiple tasks with neural networks: Cross-stitch networks for multitask learning. *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 3517–3526).

[36] Fernando, C., Banarse, D., Blundell, C., Frederic Bastien, Y., ... Pascanu, R. (2017). PathNet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.01495*.

[37] Rusu, A. A., Vecerik, M., Rothörl, T. S., Heess, N., Pascanu, R., & Hadsell, R. (2015). Sim-to-real robot learning from pixels with progressive nets. *Conference on Robot Learning* (pp. 262–270).

[38] Masse, N. Y., Grant, E., & Saxe, A. M. (2018). Alleviating catastrophic forgetting using context-dependent gating and synaptic stabilization. *International Conference on Learning Representations*.

[39] Schwarz, J., Luketina, J., Czarnecki, J. M., Grabska-Barwinska, A., Hsu, Y. C., Ritter, H., & Pascanu, R. (2018). Progress and compress: A scalable framework for continual learning. *International Conference on Machine Learning* (pp. 4535–4544).

[40] Lee, Y., Choi, S., Park, H., & Kim, M. (2020). Lifelong learning with dynamically expandable networks. *International Conference on Learning Representations*.

[41] Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., ... Pascanu, R. (2016). Independent policy agents for competitive multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 29.

[42] Fernando, C., Banarse, D., Reynolds, M., Besse, F., Pfeiffer, D., Zambaldi, V., ... & Leibo, J. Z. (2017). Continual learning with path integral learning rules. *arXiv preprint arXiv:1706.00909*.

[43] Venkat, P., Jain, P., & Netrapalli, P. (2021). Online learning under non-stationarity via offline population filtering. *Advances in Neural Information Processing Systems*, 34.

[44] Knobel, G., Schneider, S., & Agrell, E. (2022). Buffering for continual learning: A Bayesian approach. *International Conference on Machine Learning* (pp. 11410–11420).

[45] Wang, X., Zhang, X., Bai, Y., Wang, P., Lu, G., Sanchez-Lengeling, B., & Blei, D. (2020). Continual learning: A comparative study on how to defy forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10), 3543–3555.

[46] Chaudhry, A., Gordo, A., Salakhutdinov, R., & Faraci, F. (2019). Continual learning with hypernetworks. *International Conference on Learning Representations*.

[47] Hsu, Y. C., Liu, Y. C., & Kira, Z. (2018). Re-evaluating continual learning scenarios: A categorization and benchmark. *arXiv preprint arXiv:1810.02212*.

[48] Yan, S., Xie, J., He, X., Chu, W., Sun, L., & He, L. (2021). DyNet: A framework for dynamic continual learning through online node discovery and adaptive growth. *IEEE Transactions on Neural Networks and Learning Systems*.

[49] Isele, D., & Cosgun, A. (2018). Exploiting epistemic uncertainty of importance weights for uncertainty-aware deep reinforcement learning. *IEEE Robotics and Automation Letters*, 3(2), 995–1002.