DLCC #47

Beyond Depth vs Width: A Controlled Diagnostic of Generalization under Iso-Parameter CNN Trade-offs

Anonymous DLCC submission

Paper ID 47

Abstract

Depth and width yield different generalisation outcomes even under identical parameter budgets, yet the mechanisms driving this asymmetry remain poorly understood. Most prior studies compare architectures at unequal budgets or with separately tuned optimiser protocols, confounding structural effects with optimisation ones. Rather than proposing a new architecture or universal law for depth and width, we present a controlled empirical diagnostic that jointly fixes parameter count, optimiser protocol, and SGD noise scale $g = \eta N/(B(1-\beta))$, using iso-parameter VGG-style plain CNNs on CIFAR-10/100. The fixed-budget frontier is non-monotonic: the intermediate depth $D = 14$ gives the best iso-parameter accuracy, while the deepest plain model ($D = 26$) loses 1.17 percentage points on CIFAR-10 and 3.52 points on CIFAR-100 relative to $D = 14$. Critically, the two extremes fail differently. Convergence slows steadily with depth, and a matched-budget ResNet diagnostic strengthens the trainability explanation: CIFAR-100 residual reruns at the same 150-epoch schedule reach 78.00% at $D = 26$, compared with 73.66% for the plain model. A shallow-wide compensation diagnostic shows the opposite failure mode: $D = 8$ reaches 78.38% on CIFAR-100 only after widening to 19.96M parameters, roughly matching $D = 14$ at 11.32M parameters. Because these effects persist while g is fixed, they are not explained by fixed-protocol SGD noise-scale variation alone. The resulting picture is conditional: the shallow extreme is fast but parameter-expensive, while the deep plain extreme is most consistent with a trainability bottleneck.

1. Introduction

Modern convolutional networks operate across an enormous range of depth-to-width ratios, yet the choice between deeper and wider architectures under a fixed parameter budget is still driven by heuristics more than by understanding. The empirical literature consistently reports that neither deeper nor wider is universally preferable [16, 6], but the conditions under which each is advantageous, and the mechanisms that produce those differences, remain poorly characterised.

Most prior studies examine depth or width in isolation, or compare architectures at unequal compute budgets or separately tuned optimiser protocols, confounding structural effects with optimisation ones. A more subtle confound, identified in [13, 8], is that changing learning rate, batch size, momentum, or data budget changes the SGD noise scale $g \approx \eta N/(B(1-\beta))$. If these choices vary across architectures, apparent structural advantages may partly reflect optimiser implicit regularisation. Work that controls these protocol-level quantities while also offering a mechanistic account of when depth or width dominates is scarce. What is missing is a diagnostic that holds all three (parameter budget, optimiser protocol, and noise scale) fixed simultaneously, isolates the two failure modes at opposite ends of the depth-width spectrum, and uses targeted interventions (residual pathways, off-budget widening) to attribute each failure mode to a specific mechanism.

We present a controlled empirical diagnostic of depth-width trade-offs under iso-parameter constraints. The goal is not to claim that intermediate depth is surprising in itself, but to isolate whether the deep-plain penalty remains after the main budget and optimiser confounds are removed. We fix the total parameter count at $\sim$5M and sweep a 4×4 grid of depths and widths, solving numerically for the width that hits the budget at each depth. Three hypotheses are tested directly:

1. **(H1, fixed-budget frontier)** At a fixed parameter budget, depth is not monotonically beneficial for plain CNNs. An intermediate depth-width allocation can outperform both shallow-wide and deep-narrow extremes in this controlled grid.
2. **(H2, trainability)** The degradation of deep plain CNNs is primarily a trainability effect: if skip connections reduce the deep-model gap at matched budget, optimisation dynamics rather than representational ca-

pacity are implicated.

3. **(H3, shallow-wide compensation)** The shallow-wide endpoint is not primarily trainability-limited. If its weakness is lower parameter efficiency, additional width should recover accuracy only at a substantially larger parameter cost.

A matched-depth ResNet control acts as a diagnostic for H2. If introducing skip connections reduces the performance gap between deep plain and deep residual networks at matched parameter budget, a trainability mechanism is implicated. If the gap persists, representational or schedule-limited explanations remain plausible. To isolate structural effects from optimiser-interaction effects, the main experiments use a fixed protocol in which learning rate, batch size, momentum, schedule, and data budget are held constant across the plain-CNN grid. This also fixes the Smith and Le SGD noise scale, allowing us to test whether the observed width/depth differences reduce to optimiser noise.

Our contribution is therefore a controlled empirical diagnosis rather than a new architecture. Specifically, we provide:

- An iso-parameter 4×4 depth-width grid on CIFAR-10/100 with deterministic budget control within 0.8% (Fig. 1).
- A fixed-protocol accuracy and train/test gap analysis showing a non-monotonic iso-parameter frontier, with $D = 14$ best and $D = 26$ degraded on both datasets.
- Layer-wise gradient-norm and convergence diagnostics that quantify the optimisation penalty paid by deeper plain networks.
- A matched-budget ResNet and shallow-wide compensation diagnostics showing that the two extremes fail differently. The deep plain penalty is recoverable under residual pathways consistent with trainability, while the shallow endpoint recovers only at substantially higher parameter cost.
- An SGD noise-scale check demonstrating that the architecture-correlated generalisation differences are not reducible to g alone in the fixed protocol.

2. Related Work

Depth and width in CNN theory. Plain VGG-style CNNs [12] make depth and width easy to vary without shortcut pathways, which is useful when studying optimisation rather than architectural state of the art. Wide residual networks [16] showed that width can compensate for depth at equal parameter counts, while ResNet [6] introduced residual connections to make very deep networks trainable. Later analyses characterise the initialisation and signal-propagation conditions under which very deep plain networks can be optimised [10, 5]. Our work does not claim novelty in depth-width trade-offs or residual trainability gains. It re-examines them while jointly controlling parameter count and SGD noise scale.

Architecture scaling and design spaces. Resource-aware architecture studies compare accuracy, parameter count, operations, and latency across families [2]. Compound-scaling and design-space studies then vary depth, width, resolution, and related dimensions to obtain stronger accuracy-efficiency trade-offs [14, 11]. Revisiting ResNets, Bello et al. [1] further showed that the preferred depth/width scaling strategy can depend on the training regime, reinforcing that architecture and optimiser protocol are hard to disentangle. Canziani et al. [2] also show that parameter count and accuracy trade off differently across architecture families, motivating our use of parameter efficiency when interpreting the shallow-wide endpoint's recovery cost. Our goal is instead diagnostic: we restrict the family to plain VGG-style CNNs, fix the parameter budget and optimiser protocol, and use residual pathways only as a trainability intervention.

Positioning of this study. The closest question to ours is not whether modern residual or compound-scaled models are stronger in absolute terms, but whether an apparent depth-width advantage survives when the architecture family, parameter budget, data budget, and optimiser state are held fixed. We therefore treat depth and width as controlled variables inside one deliberately simple model family, and treat residual pathways and extra shallow width as mechanism interventions. This positions the project as a study of learning behaviour under controlled trade-offs rather than a benchmark-oriented architecture search.

Sharpness and flat minima. Flat-minimum accounts link optimisation geometry to generalisation [7, 9], but sharpness is sensitive to reparameterisation [3]. We therefore treat flatness as an alternative explanation and focus on directly observed trainability diagnostics.

SGD noise scale. Smith and Le [13] and Jastrzebski et al. [8] formalised the approximate noise scale $g \approx \eta N/(B(1-\beta))$ governing SGD's implicit regularisation. Our fixed protocol and explicit reporting of g (Fig. 6) are designed specifically to prevent the architecture comparison from being explained by different protocol-level noise scales.

Fairness of empirical comparisons. Depth-width studies can be distorted by several moving targets: parameter count,

training length, augmentation, learning-rate tuning, and implicit regularisation can all change with architecture. Dogo et al. [4] study the interaction between optimiser choice and CNN depth/width variation across multiple datasets, but do not control parameter budget or SGD noise scale across conditions, leaving structural and optimiser effects entangled. We therefore use a fixed optimiser protocol rather than searching separately for each cell's best hyperparameters, sacrificing claims about absolute best accuracy but making the mechanism sharper: frontier changes cannot be attributed to different learning rates, batch sizes, data budgets, or noise-scale values.

Trainability of deep plain networks. The observation that plain networks degrade with depth while residual ones do not was the central motivation for ResNet [6]. Follow-up work on Fixup [17] and dynamical-isometry initialisations [15] argued that much of this degradation is optimiser-side rather than representational. Our ResNet control formalises this argument as a diagnostic on our own grid: where skip connections close the gap, the effect is consistent with trainability. Where they do not, a representational account remains plausible.

3. Methods

3.1. Iso-parameter grid construction

Both architecture families share a three-stage VGG-style layout with channel multipliers $(w, 2w, 4w)$, 3×3 convolutions, BatchNorm, ReLU, and max-pooling between stages. *Depth* D counts convolutional layers and is distributed across stages as evenly as possible. *Width* w is the base channel count. This simple family exposes the depth-width allocation problem without additional architectural mechanisms.

Given depth D and target count $P^\star$, we solve for the integer $w^\star$ minimising $|P(D,w) - P^\star|$ by bisection on the monotone function $P(D,w)$. Here $P^\star = 5\times10^6$, and the achieved error across $D \in \{8, 14, 20, 26\}$ is below 0.8% for both plain and residual families (Fig. 1). This makes the iso row a budget-allocation comparison rather than a raw parameter-count comparison.

At each depth we additionally train widths at ratios $\{0.5, 0.75, 1.5\}$ of $w^\star$ to contextualise the local parameter/accuracy trade-off. These off-budget cells do not answer the main fixed-budget question.

3.2. Hypothesis-to-measure mapping

The three hypotheses use different comparison sets. H1 is evaluated on the $1.0\times$ iso-parameter column, where depth changes but the parameter budget is matched, and neighbouring width ratios only check whether this conclusion is consistent with the local accuracy surface. H2 is evaluated with optimisation diagnostics and a matched-budget residual intervention: convergence epoch and gradient-ratio trends identify symptoms, while residual recovery tests whether changing the optimisation pathway specifically helps the deep endpoint. H3 is evaluated with a deliberately off-budget shallow-wide run, asking whether the shallow endpoint can recover accuracy and how expensive that recovery is. This mapping keeps the main fixed-budget claim separate from the mechanism checks.

3.3. Training protocols

All plain-CNN runs use SGD with momentum 0.9, weight decay 5×10^{-4}, cosine learning-rate annealing, learning rate $\eta = 0.1$, batch size 128, and standard CIFAR augmentation. We intentionally do not tune a separate learning rate, batch size, or training length for each architecture: doing so could improve absolute accuracy but would obscure the mechanism by giving each cell a different optimiser state.

All plain-CNN cells are trained for 150 epochs. The original CIFAR-100 ResNet diagnostic used 200 epochs. We keep it as a recoverability check and add CIFAR-100 residual reruns for seeds 1 and 2 at the matched 150-epoch schedule. Under the fixed optimiser protocol the Smith and Le noise scale $g = \frac{\eta N}{B(1-\beta)}$ is constant at 390.625, so differences across depth and width cannot be attributed to a different SGD noise scale.

3.4. Trainability instrumentation

For every run we report test accuracy and a train/test gap proxy, computed as online augmented train accuracy minus clean test accuracy. Because the training side uses augmentation and train-mode BatchNorm, we interpret this as a consistent optimisation/generalisation proxy rather than a calibrated clean-train generalisation estimate.

To test H2 we log each convolutional weight tensor's gradient norm on a fixed probe batch. We derive (i) the first-to-last layer grad-norm ratio, a proxy for gradient attenuation or amplification across depth, and (ii) the convergence epoch at which online train accuracy first reaches 99% of its final value. The former tests a gradient-flow symptom. The latter tests whether global optimisation slows down.

3.5. ResNet diagnostic control

To separate trainability (H2) from representation, we train a matched ResNet control at the same iso-parameter budget. It shares the layout and width solver. The intervention is changing the optimisation pathway by wrapping pairs of 3×3 convolutions in residual blocks with identity shortcuts (1×1 projections where channels change). If deep-plain degradation is reduced, H2 is supported. If it

DLCC #47

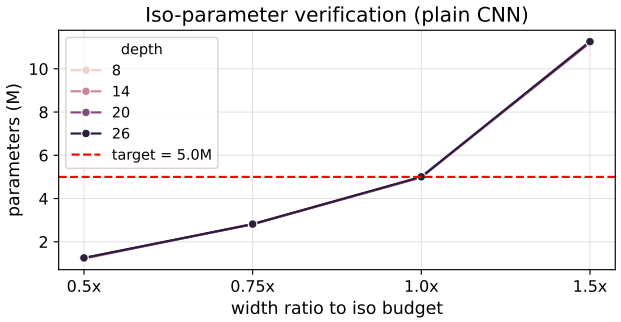

Figure 1. Iso-parameter verification. Parameter count for every cell of the plain-CNN grid. The dashed line marks the 5×10^6 target. Iso-parameter cells hit the budget within 0.55%.

persists, representational or schedule-limited explanations remain plausible.

3.6. Shallow-wide compensation diagnostic

To test H3 without changing the main fixed-budget frontier, we add a post-hoc CIFAR-100 diagnostic for the shallow endpoint: $D = 8$, $w = 208$, seeds 0, 1, and 2, trained for the same 150 epochs under the fixed protocol. This is $2.0\times$ the $D = 8$ iso width and is compared against the existing $D = 8$ and $D = 14$ $1.0\times$/$1.5\times$ rows.

4. Experiments

4.1. Grid setup

We sweep depths $D \in \{8, 14, 20, 26\}$ and, for each depth, the iso-parameter width plus three off-budget neighbours. All iso cells hit 5×10^6 parameters within 0.8% for both architecture families (Fig. 1), so trends along the $1.0\times$ column are allocation effects rather than raw parameter-count effects. The main grid comprises 3 seeds $\times$ 2 datasets $\times$ 16 cells = 96 plain-CNN runs. The ResNet diagnostic grid adds one seed at each iso depth on both datasets, plus two additional CIFAR-100 seeds at the matched 150-epoch schedule.

4.2. H1: the fixed-budget frontier is non-monotonic

Result. At the iso-parameter widths, performance peaks at the intermediate depth $D = 14$ on both datasets (Fig. 2, Table 2). On CIFAR-10, $D = 14$ reaches 95.07% test accuracy with a 4.92 percentage-point gap proxy, while the deepest plain model $D = 26$ drops to 93.90% with a 6.03 point proxy. We note that on CIFAR-10 the $D = 14$ vs. $D = 20$ gap (95.07 vs. 94.86) is within one seed-σ, so the CIFAR-10 frontier is better described as a D=14/20 plateau above D=8 and well above D=26. The pattern is sharper on CIFAR-100: $D = 14$ reaches 77.18%, but $D = 26$ falls to 73.66%, increasing the gap proxy from 22.79 to 26.24 points. On CIFAR-100 the $D = 14$ row also dominates every other depth at all four width ratios (cf. Fig. 2), so the iso-row peak is not an artifact of the $1.0\times$ slice. Off-budget widening usually improves the local frontier, especially on CIFAR-100 where the $1.5\times$ width is best at every depth.

Interpretation. These results support H1 conditionally: deeper is not automatically better once the parameter budget is fixed. Moving right within a row usually helps because the model receives more channels and parameters, but moving down the iso column trades width for depth at roughly fixed capacity. The $D = 14$ to $D = 26$ drop therefore indicates that extra plain depth carries an optimisation cost large enough to offset any representational benefit in this regime. The $D = 8$ to $D = 14$ gap is analysed separately in §4.5 because the shallow endpoint fits quickly rather than showing the same trainability symptoms.

4.3. H2: trainability degrades with depth

Result. The convergence epoch rises steadily as depth increases. Averaged across CIFAR-10/100 iso-parameter runs, convergence moves from epoch 111.5 at $D = 8$ to 129.0 at $D = 26$ (Table 2). The gradient-ratio proxy is not monotonic across the full grid (Fig. 3). We interpret this proxy qualitatively, while the main accuracy and gap uncertainty is reported as seed standard deviations in Table 2.

Interpretation. The clearest positive signal is slower global optimisation: the deepest plain model takes longer to reach its own final training accuracy despite having no larger budget. The gradient-ratio panel rules out a simple first-layer vanishing-gradient collapse, leaving a broader trainability explanation supported by the convergence trend and by the matched-budget ResNet control below. We note one caveat: convergence is defined relative to each run's *own* final online train accuracy (99% of final), so a deeper model with a slightly lower final train accuracy uses a slightly lower threshold. If anything, this understates the deep-side trainability gap, making the monotonic increase to $D = 26$ conservative.

4.4. ResNet diagnostic control

Result. The ResNet control shows that the deepest plain-model gap is recoverable under residual pathways. On CIFAR-10, the $D = 26$ residual model improves test accuracy from 93.90% to 95.07%, bringing its gap proxy down from 6.03 to 4.92 points. On CIFAR-100, additional matched-schedule 150-epoch residual reruns for seeds 1 and 2 show that the recovery is already present at the plain-grid endpoint: at $D = 26$, the residual model reaches 78.00% and reduces the gap proxy from 26.24 to 21.98 points. The original seed-0 200-epoch residual diagnostic reaches

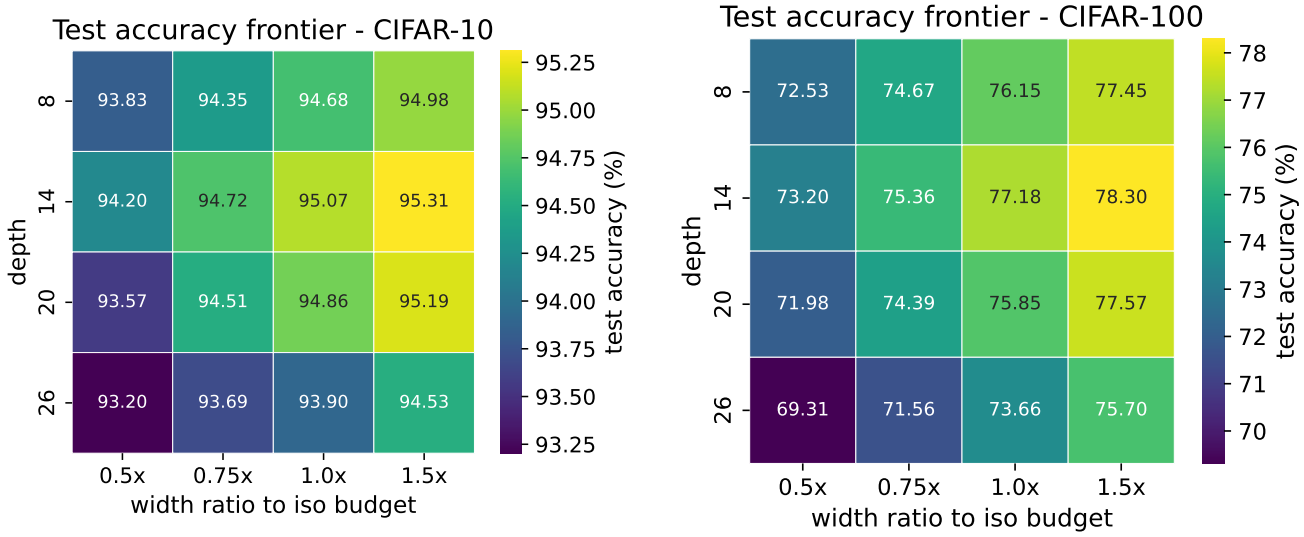

Figure 2. Test-accuracy frontier under the fixed protocol. Rows are depth and columns are local width ratio relative to the iso-parameter width. Cells show 3-seed mean test accuracy in percent. Iso-row standard deviations are reported in Table 2.

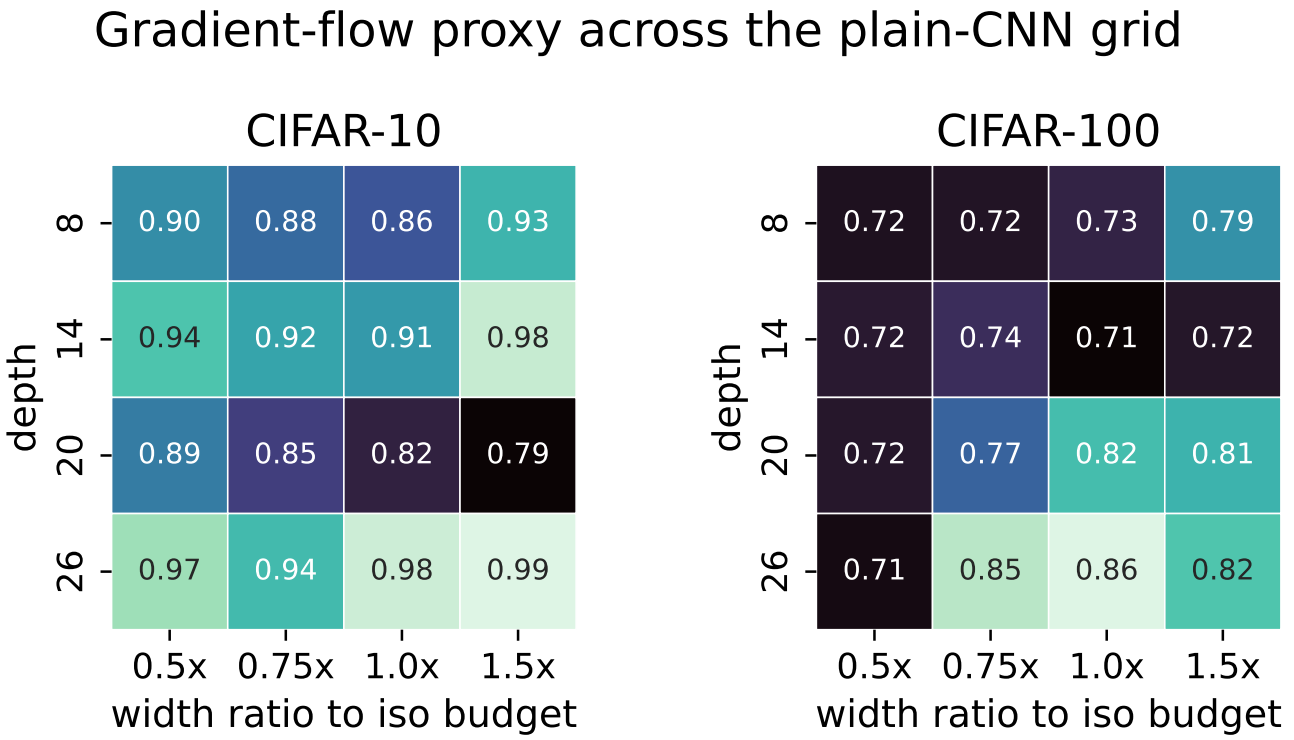

Figure 3. First-to-last gradient-norm ratio across the plain-CNN grid. Cells show seed means. The proxy does not collapse monotonically with depth, suggesting that the deep-model penalty is broader than a single vanishing-gradient symptom.

78.53%, so the longer schedule confirms recoverability but is no longer the only evidence for the deep residual gain.

Interpretation. Because the residual control keeps depth scale and parameter budget while changing the optimisation pathway, recovery at $D = 20$ and $D = 26$ supports H2. Crucially, the gain is depth-selective: on CIFAR-100 the matched rerun is slightly below plain at $D = 8$, nearly tied at $D = 14$, and clearly better at $D = 20$ and $D = 26$. This fits trainability better than a uniform "ResNet is better"

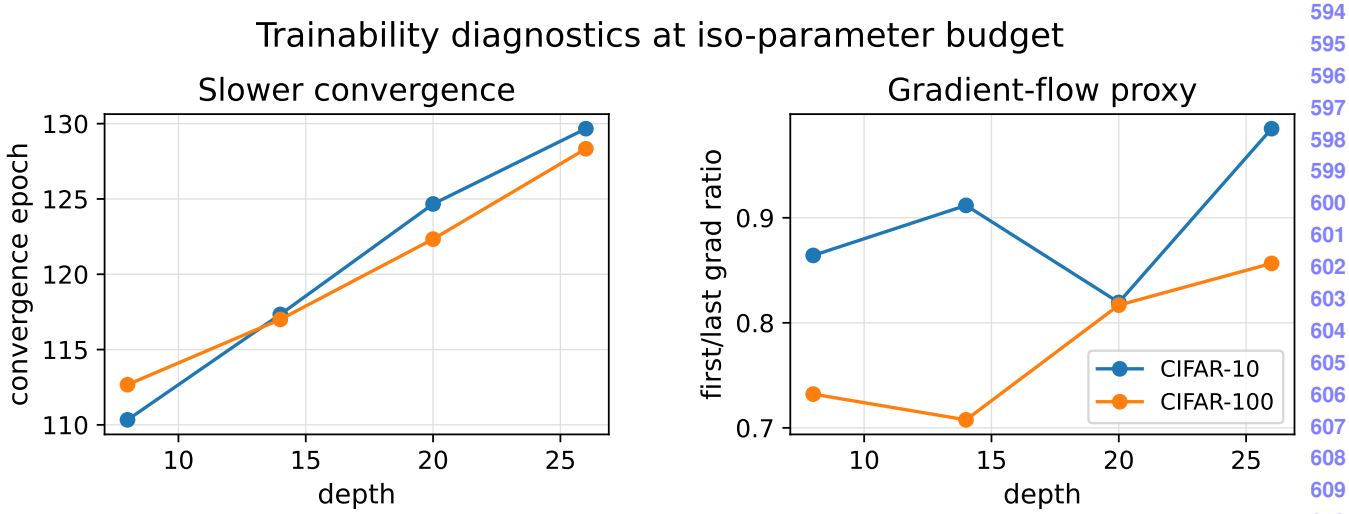

Figure 4. Iso-parameter trainability diagnostics for the plain CNN. Convergence epoch increases monotonically with depth, while the first/last gradient-ratio proxy shows that the deep-model penalty is not a simple one-layer vanishing-gradient collapse.

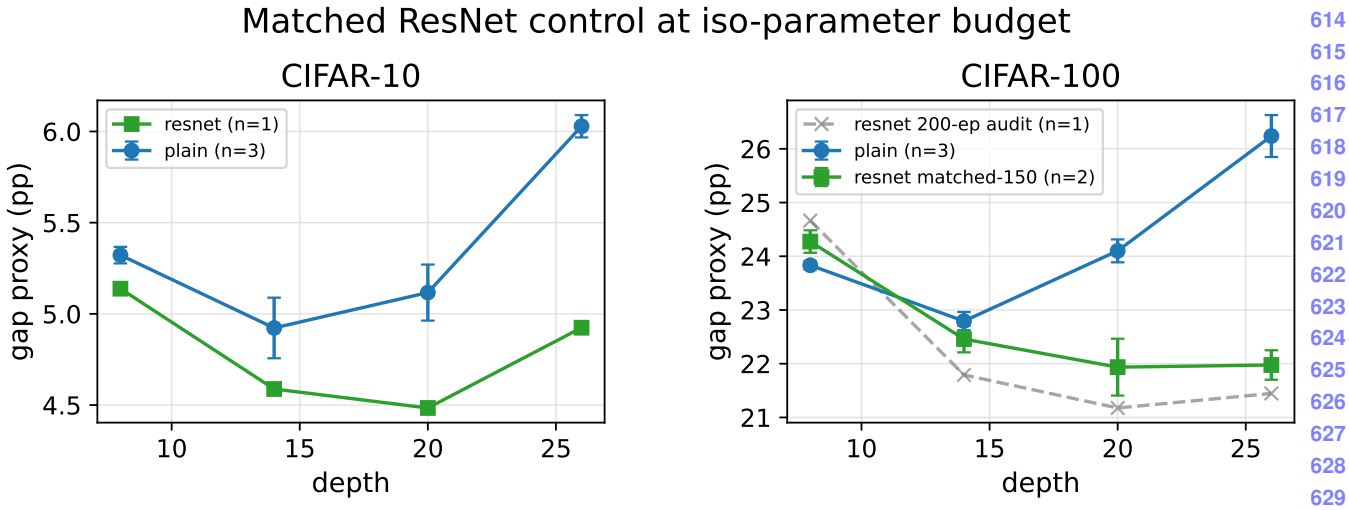

Figure 5. Train/test gap proxy (in percentage points) for plain and residual architectures at matched iso-parameter budgets. CIFAR-10 uses one residual seed at 150 epochs. CIFAR-100 shows the matched 150-epoch reruns (seeds 1 and 2, headline solid line) alongside the original 200-epoch seed-0 audit (dashed). Per-panel y-ranges differ. CIFAR-10 gaps are smaller in absolute pp because the task is easier.

Table 1. CIFAR-100 ResNet schedule diagnostic. Values are test accuracy (%). Here s0 denotes seed 0, and Δ is rerun 150 minus plain 150.

D	plain 150 mean	audit 150 s0	rerun 150 mean	final 200 s0	Δ rerun − plain
8	76.15	71.50	75.70	75.31	-0.45
14	77.18	72.89	77.51	78.19	+0.33
20	75.85	73.64	78.04	78.80	+2.19
26	73.66	74.21	78.00	78.53	+4.34

explanation. Table 1 also separates the seed-0 audit from a 200-epoch cosine run from the direct 150-epoch reruns, which are the fairer matched-schedule endpoint.

4.5. H3: shallow-wide width compensation

Result. The shallow-wide endpoint does not show the same optimisation symptoms as the deep endpoint: on CIFAR-100, $D = 8$ reaches 99.98% final online train accuracy and converges faster than $D = 14$ at both the 1.0× and 1.5× widths. Increasing width can compensate for its accuracy deficit. The post-hoc $D = 8, w = 208$ diagnostic reaches 78.38% test accuracy with 19.96M parameters, compared with 76.15% at $D = 8, w = 104$ and 77.45% at $D = 8, w = 156$. It exceeds the fixed-budget $D = 14, w = 77$ row by 1.20 points, but only roughly matches the wider $D = 14, w = 116$ row (78.30%, 11.32M

parameters) while using 1.76× more parameters. Its peak-to-final drop is only 0.13 points.

Interpretation. This supports H3: shallow-wide underperformance is better supported as parameter inefficiency than as trainability failure under these diagnostics. Width can recover the shallow endpoint, so $D = 8$ is not representationally hopeless. However, the recovery is expensive relative to the intermediate-depth frontier. Together with H2, this suggests asymmetric failure modes: shallow-wide models are fast to fit but parameter-expensive, whereas deep-narrow plain models are more trainability-limited.

4.6. Decoupling width from SGD noise scale

Result. In the fixed protocol, all plain-grid runs use the same η, B, β, and data size, so they share $g = 390.625$. Despite this, the CIFAR-100 gap proxy ranges from 21.68 to 28.04 points across cells, and the CIFAR-10 gap proxy ranges from 4.69 to 6.49 points across cells. Width and depth therefore continue to explain substantial variation at identical noise scale.

Interpretation. This rules out variation in the Smith and Le noise-scale value as the source of the observed frontier under the fixed protocol. Figure 6 is a confound check, not a claim that noise scale never matters: substantial gap variation remains at the same g, so the pattern must involve architecture, optimisation dynamics, or their interaction beyond this scalar noise-scale value. The downward trend with increasing width is consistent with width changing capacity or implicit regularisation, but our diagnostics do not isolate which channel is responsible. The key point for the fixed-protocol comparison is narrower: depth and width still correlate with the gap after the scalar noise-scale value has been held constant.

Full grid summary. Table 2 consolidates the iso-parameter plain-CNN rows used for the main fixed-budget comparison.

5. Discussion

How the figures fit together. Figure 1 establishes budget control, Fig. 2 shows the non-monotonic frontier, Figs. 3 and 4 probe the deep side for trainability symptoms, Fig. 5 and Table 1 intervene with residual connections, the shallow-wide diagnostic addresses the opposite endpoint, and Fig. 6 rules out fixed-protocol noise-scale variation as a confound.

Scope of the contribution. The main contribution is a controlled empirical analysis showing that the two extremes fail differently, which is a result that only becomes visible once parameter budget, optimiser protocol, and SGD noise scale are jointly controlled.

Implications for fixed-budget comparisons. A fixed parameter count is necessary but not sufficient for a fair architectural comparison: here, the same budget hides a residual-recoverable deep endpoint and a parameter-expensive shallow endpoint. Thus fixed-budget studies should report the budget solver, optimiser protocol, and at least one diagnostic intervention before attributing a frontier to depth or width alone.

Interpreting the shallow-wide endpoint. Under our diagnostics, the $D = 8$ endpoint is fast to fit and does not show the same trainability limitation as the deep endpoint. The 2.0× diagnostic reaches 78.38% on CIFAR-100, but uses 19.96M parameters: nearly 4× the $D = 14$ iso-budget and 1.76× the 11.32M-parameter $D = 14$ 1.5× model, which reaches 78.30%. Its small peak-to-final drop (0.13 points) also weakens a late-overfitting or schedule-mismatch explanation. Thus width compensates for shallow depth, but inefficiently in this setting.

Interpreting the ResNet control. The residual control is the strongest evidence for trainability, but it should be read as a diagnostic intervention rather than a fully powered architecture comparison. At $D = 26$, residual connections recover 1.17 points on CIFAR-10 and 4.34 points on CIFAR-100 without increasing the target budget. The CIFAR-100 residual gain is depth-selective (-0.45, $+0.33$, $+2.19$, $+4.34$ for $D = \{8, 14, 20, 26\}$), so the control is not merely showing that ResNets are uniformly better. Shortcuts help most where the plain architecture is most trainability-limited. Because CIFAR-10 uses one residual seed and the matched CIFAR-100 residual rows use two seeds, the result supports a trainability interpretation but does not by itself estimate the full distribution of residual-model performance.

Remaining uncertainty. The H3 diagnostic narrows but does not fully solve the shallow-side mechanism. It is consistent with lower parameter efficiency, but does not distinguish hierarchy, feature reuse, or regularisation as the underlying cause. Likewise, H2 is supported by convergence and residual recovery, not by a complete optimisation-theory proof. We therefore interpret effects by size, direction, and consistency across datasets rather than by formal significance testing. The plain grid is replicated over three seeds, while the residual diagnostics are intentionally lower-powered checks.

DLCC #47

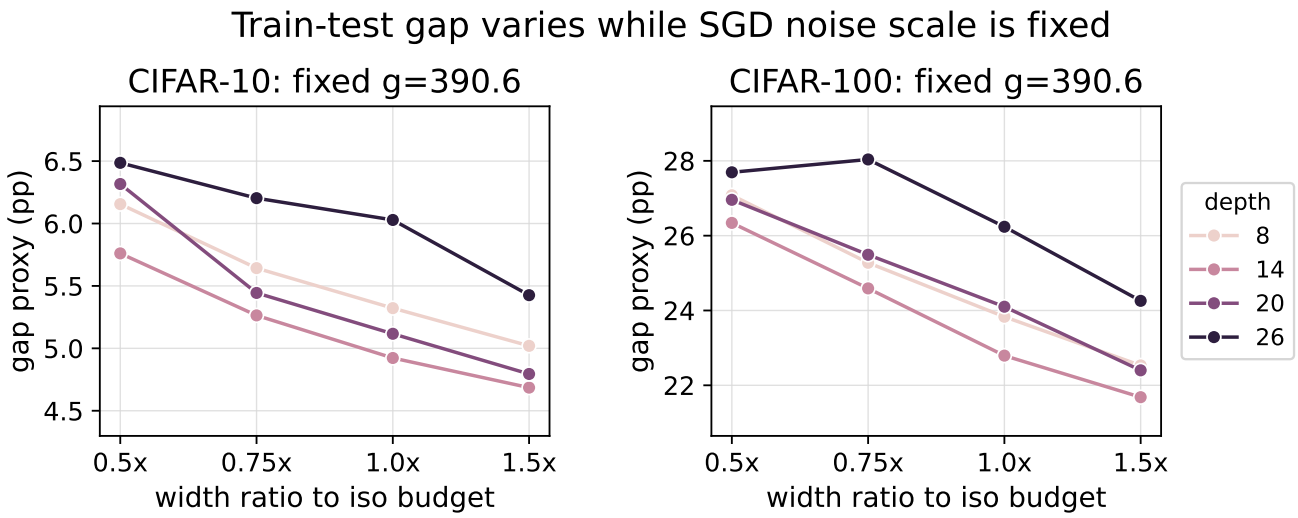

Figure 6. The train/test gap proxy (in percentage points) varies across width and depth while the fixed protocol holds SGD noise scale at $g = 390.625$ for every plain-grid run. Panels use separate y-axis ranges so the smaller CIFAR-10 variation remains visible. Absolute gap magnitudes should be compared using the axes.

Table 2. Iso-parameter fixed-protocol summary for the plain CNN. Accuracy and gap are percentages. Parentheses are 3-seed standard deviations. Conv. and grad columns average both datasets.

(depth, width)	params (M)	CIFAR-10 acc	CIFAR-10 gap	CIFAR-100 acc	CIFAR-100 gap	conv. epoch	grad ratio
(8, 104)	4.99	94.68 (0.05)	5.32 (0.05)	76.15 (0.08)	23.83 (0.08)	111.5	0.80
(14, 77)	4.99	95.07 (0.17)	4.92 (0.17)	77.18 (0.18)	22.79 (0.17)	117.2	0.81
(20, 64)	5.00	94.86 (0.15)	5.12 (0.15)	75.85 (0.21)	24.10 (0.21)	123.5	0.82
(26, 56)	5.02	93.90 (0.05)	6.03 (0.06)	73.66 (0.39)	26.24 (0.39)	129.0	0.92

Limitations. Our study uses one main parameter budget (5×10^6), one VGG-style architecture family, and CIFAR-10/100 classification only. The 19.96M shallow-wide run is a targeted diagnostic, not a full second-budget grid. The ResNet control is less powered than the plain grid: CIFAR-10 uses one residual seed, and matched CIFAR-100 reruns use seeds 1 and 2. We also match parameter count rather than FLOPs or wall-clock cost, so the result is a budget-allocation diagnosis rather than a deployment-efficiency claim. Finally, our gap is an online train/test proxy rather than a clean train-set evaluation, and we do not report final Hessian sharpness. These limits bound the claim to this controlled trade-off regime, not a universal ranking of depth and width.

6. Conclusion

We presented a fixed-protocol empirical study of depth-width allocation at an iso-parameter CNN budget. The main result is not that depth or width wins universally, nor that intermediate depth is inherently novel, but that the frontier is conditional: in our 5M-parameter grid, the intermediate $D = 14$ plain CNN is best. The two extremes fail differently. The deepest $D = 26$ plain model degrades despite the same budget and optimiser noise-scale value, and is partly repaired by residual pathways. The shallow $D = 8$ endpoint is fast to optimise and can recover with enough width, but doing so is parameter-expensive. Together, these findings support a trade-off view in which intermediate depth balances parameter efficiency against trainability, and show why fixed-budget architectural comparisons must control both parameter count and optimiser state.

Practically, this means that a fixed parameter count should be treated as a starting control rather than a complete fairness guarantee: budget solvers, optimiser protocols, and trainability diagnostics all affect how a depth-width frontier should be read. Future work should repeat the diagnostic across multiple parameter budgets, stronger architecture families, and compute-matched rather than parameter-matched settings. The broader goal is not to choose a universal depth or width rule, but to make architectural scaling claims conditional on the failure mode they actually diagnose.

References

[1] I. Bello, W. Fedus, X. Du, E. D. Cubuk, A. Srinivas, T.-Y. Lin, J. Shlens, and B. Zoph. Revisiting ResNets: Improved training and scaling strategies. In *NeurIPS*, 2021. 2

[2] A. Canziani, A. Paszke, and E. Culurciello. An analysis of deep neural network models for practical applications. *arXiv:1605.07678*, 2016. 2

[3] L. Dinh, R. Pascanu, S. Bengio, and Y. Bengio. Sharp minima can generalize for deep nets. In *ICML*, 2017. 2

[4] E. M. Dogo, O. J. Afolabi, and B. Twala. On the relative impact of optimizers on convolutional neural networks with varying depth and width for image classification. *Applied Sciences*, 12(23):11976, 2022. 3

[5] B. Hanin and D. Rolnick. How to start training: The effect of initialization and architecture. In *NeurIPS*, 2018. 2

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 2, 3

[7] S. Hochreiter and J. Schmidhuber. Flat minima. *Neural Computation*, 9(1):1–42, 1997. 2

[8] S. Jastrzebski, Z. Kenton, D. Arpit, N. Ballas, A. Fischer, Y. Bengio, and A. Storkey. Three factors influencing minima in SGD. *arXiv:1711.04623*, 2017. 1, 2

[9] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *ICLR*, 2017. 2

[10] Q. Nguyen and M. Hein. Optimization landscape and expressivity of deep CNNs. In *ICML*, 2018. 2

[11] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollar. Designing network design spaces. In *CVPR*, 2020. 2

[12] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 2

[13] S. L. Smith and Q. V. Le. A bayesian perspective on generalization and stochastic gradient descent. In *ICLR*, 2018. 1, 2

[14] M. Tan and Q. V. Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019. 2

[15] L. Xiao, Y. Bahri, J. Sohl-Dickstein, S. S. Schoenholz, and J. Pennington. Dynamical isometry and a mean field theory of CNNs: How to train 10,000-layer vanilla convolutional neural networks. In *ICML*, 2018. 3

[16] S. Zagoruyko and N. Komodakis. Wide residual networks. In *BMVC*, 2016. 1, 2

[17] H. Zhang, Y. N. Dauphin, and T. Ma. Fixup initialization: Residual learning without normalization. In *ICLR*, 2019. 3

A. Supplementary details

Training details. All experiments use PyTorch 2.x. Each run is launched by `src/train/train.py` with a merged YAML configuration. The resolved configuration, git commit SHA, parameter count, and final metrics are snapshotted into the run directory. The committed `results/summary.csv` is the source for the aggregate numbers in the paper, and `scripts/make_figures.py` regenerates all figures from that summary.

Iso-parameter solver. Given a depth D and target parameter count $P^\star$, bisection on $P(D, w)$ yields an integer width w that minimises $|P(D, w) - P^\star|$. For the plain VGG-style network the resulting iso widths are $(D, w) \in \{(8, 104), (14, 77), (20, 64), (26, 56)\}$. For the residual control, projection shortcuts alter the count slightly, producing matched-budget widths $(8, 93), (14, 72), (20, 61), (26, 54)$ with achieved parameter counts 5.036M, 4.988M, 4.993M, 5.020M respectively (all within 0.72% of the 5×10^6 target).

Extended grid observations. The off-budget cells are used to contextualise the iso-parameter frontier. On both CIFAR-10 and CIFAR-100, increasing width from $0.5\times$ to $1.5\times$ the iso width reduces the train/test gap proxy at every tested depth. This supports the local widening trend, but these off-budget cells answer a different question from how to allocate a fixed parameter budget.

Reproducibility checks. The final artifact was checked with the iso-parameter verifier, unit tests, the ResNet schedule audit script, and a clean PDF rebuild.