ON THE FRAGILITY OF REPRESENTATIONAL SIMILARITY METRICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Representational similarity metrics are widely used tools that measure correspondences between machine learning (ML) models' learned representations. We show that these metrics can be brittle and raise important questions about whether they provide meaningful notions of model similarity. To do this, we introduce an adversarial optimization framework that manipulates the parameters and thus the representation space of an adversarial target model relative to a reference benign model. Using this framework, we show that we can inflate or suppress representational similarity metrics, such as Centered Kernel Alignment (CKA) and Singular Vector Canonical Correlation Analysis (SVCCA), in the adversarial model while keeping its parameters and task performance close to those of a benign model. We can also make the adversarial and benign models functionally unaligned but keep their representational similarity high and parameters similar. Across five datasets and eight architectures, our findings show that high functional similarity between models does not necessarily imply high representational similarity between them, and vice versa.

1 INTRODUCTION

Understanding the internal representations of machine learning models is an important goal of modern machine learning research (Samek et al., 2017; Zhang et al., 2021; Fan et al., 2021). Indeed, representation similarity analysis (Klabunde et al., 2025b) has emerged as a practical approach to comparing models' hidden representation spaces for a variety of purposes. For example, representation similarity comparisons are used to quantify how representations are preserved or altered after fine-tuning or transfer learning (Dwivedi & Roig, 2019; Phang et al., 2021; Goerttler & Obermayer, 2022). These metrics are also commonly used to identify layer-wise correspondences across models and reveal how design or optimization choices affect the learned features (Kornblith et al., 2019; Morcos et al., 2018). Finally, representation similarity has been used to study robustness (Cianfarani et al., 2022), improve model safety (Li et al., 2026; Hsiung et al., 2026), and fingerprint models (Zhang et al., 2025).

Representational similarity metrics range from methods that directly measure representation tensors, such as canonical correlation and linear alignment, to methods that compare increasingly abstracted summaries of them, including local neighborhood structure, global topology, and descriptive statistics. These approaches are helpfully summarized by Klabunde et al. (2025a). Two popular metrics are singular vector canonical correlation analysis (SVCCA) and centered kernel alignment (CKA). SVCCA applies dimensionality reduction before canonical correlation analysis, comparing directions that capture most of the activation variance (Raghu et al., 2017). CKA, the more widely used metric, compares centered Gram matrices of activations. It is invariant to orthogonal transformations and isotropic scaling, but not to arbitrary invertible linear transformations (Kornblith et al., 2019).

Fundamentally, representational similarity metrics offer a way to compare how models encode information. However, their relationship to functional similarity (*e.g.* similarity between model performance) remains an active area of research (Klabunde et al., 2025a). Ding et al. (2021) show that CKA and PWCCA, a close relative of SVCCA (Morcos et al., 2018), do not consistently track how well simple classifiers trained on those representations perform. Similarly, Davari et al. (2022) demonstrate that a model's layer-wise CKA scores can be manipulated without substantially changing its outputs. Building on these observations, *we ask whether representational similarity* ***between models*** *can be decoupled from their functional similarity, even when their parameters remain similar.*

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

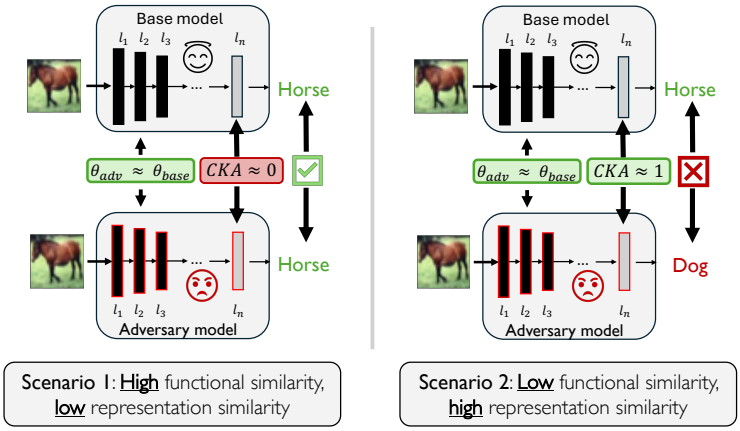


Figure 1: **Overview of our adversarial optimization framework that finds counterfactuals for representation similarity metrics.**

Our approach. To investigate this question, we introduce an adversarial optimization framework that jointly optimizes over a representational similarity metric and a task loss in the objective function (§2). Holding the model architecture and training data fixed, we select a benign reference model and optimize parameters in selected layers of a target adversarial[1] target model. The optimization process targets a desired level of benign/adversarial model representational similarity while maximizing or minimizing functional similarity between them. We also limit parameter changes in the adversarial model to determine how close in parameter space the adversarial models lie.

This setup allows us to test two key failure modes (Figure 1). **First**, we test whether two models' representational similarity can be low even when both exhibit high task performance ("functional alignment"). **Second**, we explore whether models' representational similarity can be high even as the adversarial model is optimized to be "functionally misaligned" (*i.e.* poor task performance) with the benign model. We explore these questions via a large array of experiments on six vision and one tabular model, across four vision and one tabular dataset.

Key Findings. The answer to our central question is *yes* (§3). CKA can be pushed towards extreme values, even when the adversarial and benign models are functionally aligned (§3.1). This remains true when different layers are optimized and even when parameter changes are tightly constrained. Similarly, CKA and (to a lesser extent) SVCCA can be artificially inflated even when the adversarial and benign models are functionally unaligned (§3.3). These findings imply that high representational similarity between models does not necessarily imply high functional similarity, and vice versa.

Our work suggests a need for caution in relying on representation similarity alone in analyzing model similarity (especially for sensitive applications such as model fingerprinting (Zhang et al., 2025)). It motivates a line of future work on developing more robust representation similarity metrics.

2 COUNTERFACTUALS FOR REPRESENTATION SIMILARITY METRICS

In this section, we formulate the problem of generating counterfactuals for representational similarity metrics in model parameter space. Given a benign reference model, we optimize a adversarial target model to produce a desired representational similarity score at a selected layer while controlling its functional behavior. Specifically, we seek model pairs with either (i) high functional similarity but low representational similarity, or (ii) low functional similarity but high representational similarity. We cast this search as an optimization problem over the adversarial model's parameters.

2.1 NOTATION AND SETUP

We denote the trained, benign model with good performance on the training dataset D_{train} as M_{b} with associated *fixed* parameters $\theta_b \in \mathbb{R}^m$. Here, $M_{\text{b}} : \mathbb{R}^d \to \mathcal{Y}$, where d is the dimension of the input

[1]We use the terms "adversarial" and "benign" to distinguish the model being optimized and reference model, respectively, throughout this paper. This mirrors language used in the adversarial machine learning community when one model is changed in a way intended to deceive.

space and $\mathcal{Y}$ is the label space. The optimization problem that is solved to obtain these parameters is

$$\theta_b = \arg\min_{\theta} \mathbb{E}_{(x,y)\in D_{\text{train}}} \left[\ell((x,y),\theta)\right],$$

where $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ is an appropriate loss function such as cross-entropy. We also assume that M_{b} generalizes well to test data D_{test}, *i.e.* $\mathcal{L}_{\text{b}} = \mathbb{E}_{(x,y)\in D_{\text{test}}} [\ell((x,y),\theta_{\text{b}})]$ is low. The adversarial model that is to be optimized to provide the counterfactual evidence for a given representation is denoted as M_{a}, with associated parameters θ_{a}. For simplicity, we assume that M_{a} and M_{b} come from the same parametrized hypothesis class $\mathcal{H}$, typically neural networks with a fixed architecture. This implies that $\theta_{\text{a}} \in \mathbb{R}^m$ as well. However, note this assumption is only strictly required for §2.3.

Representations and parameters: Abstractly, we consider an arbitrary representation similarity metric as $\mathsf{metric} : \mathbb{R}^{n_1 p_1} \times \mathbb{R}^{n_2 p_2} \to [0,1]$, where $\{n_i\}_{i\in\{1,2\}}$ and $\{p_i\}_{i\in\{1,2\}}$ denote the number of samples over which similarity is computed and the number of parameters respectively. Given a layer l, let $\mathbf{x}_{\text{b}}^l$ (respectively $\mathbf{x}_{\text{a}}^l$) be its vectorized feature representation. We typically compare representations resulting from the same number of samples, so we have $n = n_1 = n_2$. The parameters for a layer l are denoted as θ^l, those before a layer l as $\theta^{<l}$, and those after a layer l as $\theta^{>l}$.

Problem definition: Our main approach to demonstrating the fragility of representation similarity metrics is as follows. We search over the parameter space for a parameter vector θ_{a} such that M_{a} has either of the following properties:

1. **Functional Alignment:** $\mathsf{metric}(\mathbf{x}_{\text{a}}^l, \mathbf{x}_{\text{b}}^l)$ is low, indicating dissimilar representations, but task similarity is high (*i.e.*, both test losses are low).
2. **Functional Unalignment:** $\mathsf{metric}(\mathbf{x}_{\text{a}}^l, \mathbf{x}_{\text{b}}^l)$ is high, indicating similar representations, but task similarity is low (*i.e.*, the test loss of M_{a} is high).

If θ_{a} can be found to satisfy Property 1, then functional similarity does not imply representation similarity. If Property 2 can be satisfied, then models exist that seem similar under the representation similarity metric but are not functionally similar. These then raise the question of what one expects these metrics to show. Further implications are discussed in §5.

2.2 Unconstrained optimization over parameters

To start, we cast the problem of finding parameter vectors θ_{a} that satisfy either of the two properties above as an optimization problem with no parameter constraints. The weight-constrained version is in §2.3. To satisfy Property 1, we define the following optimization problem:

$$\min_{\theta_{\text{a}}} \mathsf{metric}(\mathbf{x}_{\text{a}}^l, \mathbf{x}_{\text{b}}^l) \quad \text{s.t. } \mathbb{E}_{(x,y)\sim D_{\text{test}}} [\ell((x,y),\theta_{\text{a}})] < t_{\text{min}}, \tag{1}$$

where t_{min} is a threshold on the loss determined using $\mathcal{L}_{\text{b}}$. Similarly, to satisfy Property 2, we define:

$$\max_{\theta_{\text{a}}} \mathsf{metric}(\mathbf{x}_{\text{a}}^l, \mathbf{x}_{\text{b}}^l) \quad \text{s.t. } \mathbb{E}_{(x,y)\sim D_{\text{test}}} [\ell((x,y),\theta_{\text{a}})] > t_{\text{max}}. \tag{2}$$

As stated, both optimizations are challenging because test data may be unavailable and the feasible region is not defined in a way that is geometrically convenient to capture. We thus consider the Lagrangian relaxations (Boyd & Vandenberghe, 2004) of each and evaluate loss over training data, assuming low excess risk (Bach, 2024), leading to the following unconstrained optimization problems:

$$\min_{\theta_{\text{a}}} \mathsf{metric}(\mathbf{x}_{\text{a}}^l, \mathbf{x}_{\text{b}}^l) + \lambda \mathbb{E}_{(x,y)\sim D_{\text{train}}} [l((x,y),\theta_{\text{a}})], \tag{3}$$

$$\max_{\theta_{\text{a}}} \mathsf{metric}(\mathbf{x}_{\text{a}}^l, \mathbf{x}_{\text{b}}^l) + \lambda \mathbb{E}_{(x,y)\sim D_{\text{train}}} [\ell((x,y),\theta_{\text{a}})]. \tag{4}$$

For ease of notation, we refer to the solution of Equation (3) (resp. Equation (4)) as $\theta_{\text{a,min}}$ ($\theta_{\text{a,max}}$), and the corresponding model as $M_{\text{a,min}}$ ($M_{\text{a,max}}$). Note that the optimization problems defined above are usually non-convex for the settings we consider (*e.g.* models under consideration are neural networks). Convergence issues for such problems are complex and out of the scope of this paper.

2.3 Constrained optimization over parameters

In the previous section, we discussed optimizing model parameters to ensure either Property 1 or 2, to demonstrate the fragility of a representation similarity metric. However, the parameters of θ_{a} were

not constrained. Given how overparametrized deep neural networks can be (Belkin et al., 2019), it would perhaps be unsurprising to find model pairs that satisfy the desired properties.

To investigate whether θ_a with the desired properties can be found *close* to the benign model θ_b, we constrain its parameters. In short, we want θ_a to be close to θ_b in an appropriately defined metric over $\mathbb{R}^p$. This leads to two constrained optimization problems for Property 1 and Property 2 respectively:

$$\min_{\theta_a} \textsf{metric}(\mathbf{x}_a^l, \mathbf{x}_b^l) + \lambda \mathbb{E}_{(x,y)\sim D_{\text{train}}}[l((x,y),\theta_a)] \quad \text{s.t. } \|\theta_a - \theta_b\|_p \leq \delta, \tag{5}$$

$$\max_{\theta_a} \textsf{metric}(\mathbf{x}_a^l, \mathbf{x}_b^l) + \lambda \mathbb{E}_{(x,y)\sim D_{\text{train}}}[l((x,y),\theta_a)] \quad \text{s.t. } \|\theta_a - \theta_b\|_p \leq \delta \tag{6}$$

We constrain using an ℓ_p norm, which is standard for regularization (Hastie et al., 2009). Since the constraints in Equation (5) and Equation (6) are geometric in nature, we can use methods like projected gradient descent to solve. The Lagrangian relaxations of the above problems are therefore:

$$\min_{\theta_a} \textsf{metric}(\mathbf{x}_a^l, \mathbf{x}_b^l) + \lambda \mathbb{E}_{(x,y)\sim D_{\text{train}}}[l((x,y),\theta_a)] + \gamma \|\theta_a - \theta_b\|_p, \tag{7}$$

$$\max_{\theta_a} \textsf{metric}(\mathbf{x}_a^l, \mathbf{x}_b^l) + \lambda \mathbb{E}_{(x,y)\sim D_{\text{train}}}[l((x,y),\theta_a)] - \gamma \|\theta_a - \theta_b\|_p, \tag{8}$$

Appendix (Table 3) summarizes all possible combinations of representation similarity, functional similarity, and parameter distance under our optimization framework. We cover all cases relevant to our counterfactuals in the main body, with discussion of the others deferred to the Appendix.

2.4 Experimental setup

To test for counterfactuals, we apply the optimization setups described in §2.2 and §2.3 to CKA with the and SVCCA. We primarily evaluate our counterfactual optimization across four image-classification datasets—MNIST (Deng, 2012), CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2010), and Tiny-ImageNet (Le et al., 2015). We train models using three ResNet architectures (18, 50, 152) (He et al., 2016) and three vision transformer architectures (Tiny/16, Small/16, Base/16) (Dosovitskiy et al., 2021) to cover a range of architectures and capacities. For tabular data, we use the CoverType dataset (Blackard & Dean, 1999) and multi-layer perceptrons (MLPs) with 3 or 5 hidden layers. Appendix Table 1 summarizes the experimental configurations and training setups.

The benign model M_b is first trained independently on the training set. We then optimize an independently initialized target model M_a using the training set according to the objectives described in Sections 2.2 and 2.3. Unless otherwise noted, for experiments reporting the final outcome of the optimization rather than its epoch wise evolution, both task accuracy and representation similarity are evaluated on held-out test data that are not used for optimization. All models are instantiated using the PyTorch Image Models (`timm`) library (Wightman et al., 2023) and trained from random initialization.

Noe that although the equations imply we apply the parameter constraint to all the layers, in our experiments we steadily increase the number of parameter tensors constrained as an ablation until all layers are constrained[2]. We report these ablation results and selected parameter tensor cutoffs in the Appendix (Table 2). To balance terms in our loss function, we assign unit weight $\lambda = 1$ to the classification and representation similarity terms, whose magnitudes are comparable in our experiments. Because the parameter distance term grows as more of the network is included in the regularizer, we set its coefficient to $\gamma = 0.1$. These coefficients were selected via grid search and were held constant throughout experiments. All experiments were conducted using NVIDIA RTX 6000 Ada GPUs.

3 Results

Using the optimization framework defined in §2, we investigate whether representational similarity can be decoupled from functional similarity. Our experiments initially focus on CKA, but we also explore SVCCA in later sections. We consider the following three research questions:

[2]For more information, refer to our anonymous github.

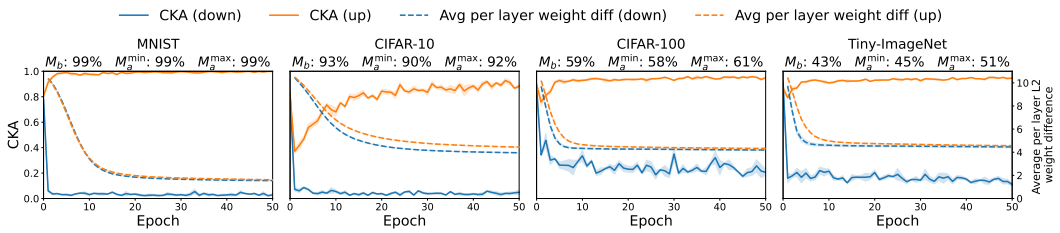


Figure 2: **Unconstrained optimization can drive CKA toward 1 (up) or 0 (down), even when both models remain functionally aligned.** *We measure CKA in the penultimate layer,* i.e. *the one just before the logit layer of the network. Results reported over* 3 *independent runs with random seeds using ResNet-18.*

- **RQ1**: can we steer CKA towards extrema in the adversarial model while maintaining task accuracy comparable to the benign model (e.g. functional alignment)? In answering this, we also explore the effect of parameter constraints and the specific layer chosen for optimization.
- **RQ2**: can CKA remain high when the adversarial model's task performance degrades relative to the benign model (e.g. functional unalignment)?
- **RQ3**: if RQ1 and RQ2 are answered affirmatively, does this result generalize to other metrics?

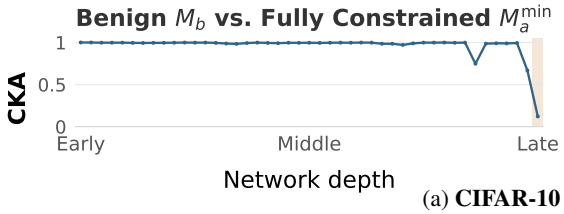
3.1 RQ1: Can we steer CKA towards extrema in the adversarial model while maintaining its functional alignment with the benign model?

To answer this, we implement the unconstrained optimization setup formalized in Section §2.2, specifically Equations 3 and 4. When adversarial model parameters are unconstrained during optimization, CKA between the benign model M_b and an optimized adversarial model M_a can be driven to either extreme. As shown in Figure 2, when CKA is measured at the penultimate layer, across three runs with different seeds on four image datasets, both M_a^min and M_a^max maintain high functional similarity to the benign model M_b. In particular, optimizing CKA toward either 0 or 1 results in only modest changes in test accuracy relative to the benign model. Our results further show that this manipulation of representational similarity is successful even when the benign and adversarial models remain functionally aligned—the accuracy of M_a still mirrors that of M_b after optimization—despite some changes to parameters.

Can we affirmatively answer RQ1, even when parameter changes are constrained? The experiments reported in Figure 2 were run with unconstrained adversarial model parameters. Thus, we next explore whether the prior result remains possible under parameter constraints (as described in §2.3 equation 7). Our goal is to find adversarial model parameters θ_a that remain close to the benign model parameters θ_b while minimizing CKA and maintaining functional alignment. Parameter distance is measured as the sum of the ℓ_2 norms of the differences between corresponding parameter tensors.

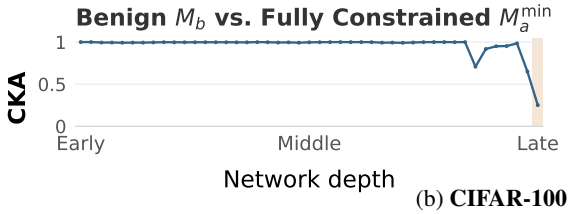
Our result stands even in this constrained setting. Figure 3 shows the result for fully constrained Resnet-18 when minimizing CKA at the penultimate layer. Across CIFAR-10, CIFAR-100, and Tiny-ImageNet, our constrained optimization framework allows us to substantially reduce the parameter space distance between M_a^{min} and M_b. The ℓ_2 distance decreases from 25373, 25728, and 34428 at Epoch 1 to 24.4, 29.7, and 27.8 at Epoch 50. Note that the ℓ_2 norm of the overall parameter vectors θ_a and θ_b remains large. Despite the models' close proximity in parameter space, CKA at the adversarialed penultimate layer is successfully driven to low (≈ 0) values. Importantly, this reduction in representation similarity does not affect the functional alignment between the models. The adversarial model's test accuracy remains within 2.5% of the benign model's. We also test the complementary setting—parameter distance increases over time, representational similarity and functional alignment remain high—and found it is achievable (see Appendix Figure 13).

Does this result persist when arbitrary layers are selected for optimization? The preceding results focused on the penultimate layer. When we constrain different layers, our result still stands. Figure 4 illustrates this, showing that CKA can be driven to either 0 or 1 by applying both optimization objectives at early, middle, and penultimate layers with all model parameters constrained.



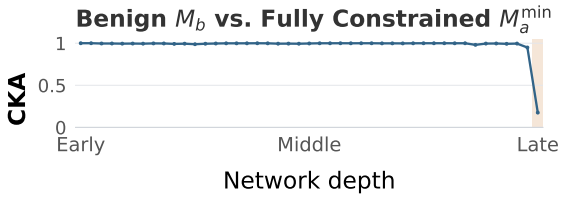
Metric	Epoch 1		Epoch 50
M_b Test Acc.			90.0%
$M_a^{\min}$ Test Acc.	52.1%	→	87.5%
$\|\theta_b - \theta_a^{\min}\|_2$	25373.98	→	24.4
$\|\theta_b\|_2$			29313.24
$\|\theta_a^{\min}\|_2$			29316.11

(a) **CIFAR-10** results



Metric	Epoch 1		Epoch 50
M_b Test Acc.			72.1%
$M_a^{\min}$ Test Acc.	13.3%	→	70.7%
$\|\theta_b - \theta_a^{\min}\|_2$	25728.93	→	29.7
$\|\theta_b\|_2$			29667.99
$\|\theta_a^{\min}\|_2$			29675.50

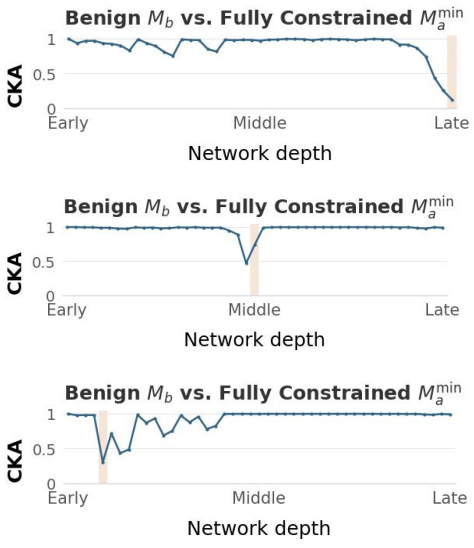
(b) **CIFAR-100** results



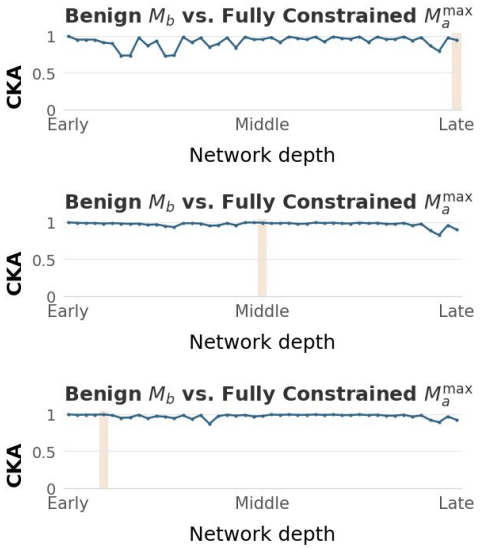
Metric	Epoch 1		Epoch 50
M_b Test Acc.			55.4%
$M_a^{\min}$ Test Acc.	12.9%	→	54.3%
$\|\theta_b - \theta_a^{\min}\|_2$	34428.4	→	27.8
$\|\theta_b\|_2$			41463.67
$\|\theta_a^{\min}\|_2$			41464.46

(c) **Tiny-ImageNet** results

Figure 3: **Even when parameter changes are constrained across all layers, an adversary can drive CKA in adversarial model M_a to 0 while maintaining functional alignment with benign model M_b.** *CKA is measured in penultimate model layer; results reported across 3 runs with random seeds using ResNet-152. M_b accuracy is reported as a fixed reference value from its independent training run.*



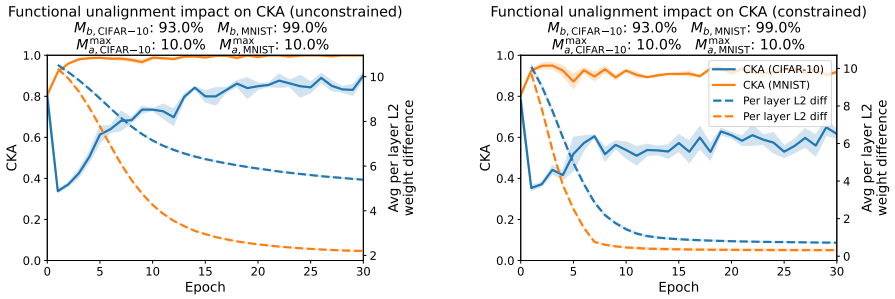
(a) **Minimizing** CKA at arbitrary layers.



(b) **Maximizing** CKA at arbitrary layers.

Figure 4: **CKA can be artificially deflated in arbitrary layers within adversarial model M_a, even when parameter changes are constrained in all layers and functional alignment is maintained between M_a and M_b.** *Results shown are for ResNet-18 models trained on CIFAR100, averaged over 3 runs.*

Answer to RQ1: Yes. CKA can be driven toward its extremes while preserving functional alignment, even when the adversarial model is constrained to remain close to the benign model in parameter space. This behavior can be induced at different target layers throughout the network.



(a) **Unconstrained** functional unalignment for MNIST and CIFAR-10 using CKA metric

(b) **Constrained** functional unalignment for MNIST and CIFAR-10 using CKA metric

Figure 5: **Our optimization framework allows us to maintain high CKA scores between M_a and M_b, even when they are functionally unaligned.** *Results averaged over 3 independent runs per setting using ResNet-50.*

3.2 RQ2: Can CKA remain high when the adversarial model M_A^{MAX} is functionally misaligned with the benign model?

To answer this, we optimize the adversarial model M_a^{max} to increase CKA while also increasing task loss In the unconstrained setting, this corresponds to using the objective of Eq. 4; to add parameter constraints, we add the regularization term from Eq. 8. Figure 5a shows that, even when M_a^{max} is functionally misaligned with M_b, CKA can remain high. Figure 5b shows that this effect persists when **all trainable layers are constrained** to remain close to the benign model. Even with this fully constrained setting keeping M_a^{max} close to M_b in parameter space while simultaneously maintaining high representational similarity and increasing task loss, CKA remains high. This again demonstrates that functional and even parameter similarity do not necessarily imply high representation similarity.

Answer to RQ2: Yes. CKA can remain high when the adversarial model becomes functionally misaligned with the benign model, even when parameters are constrained to remain close.

3.3 RQ3: Does this result extend to other similarity metrics?

Finally, we explore whether these findings generalize when we use SVCCA instead of CKA. Keeping the same experimental design and optimization framework, we ran experiments for RQ1 and RQ2 but with SVCCA in place of CKA. When we try to maximize or minimize SVCCA while keeping models functionally aligned (echoing RQ1 analysis), it barely budges (Figure 6), suggesting that SVCCA may be more robust to this optimization approach. When we repeat the functional unalignment experiments from RQ2, SVCCA declines as models become more unaligned (unlike CKA), especially when parameters are unconstrained (Figure 7(a)). With parameter constraints, SVCCA remains higher when models grow unalinged (Figure 7(b)), but we do not observe the same strong decoupling between representational similarity and functional (un)alignment that occurs with CKA. These results suggest that SVCCA is not as susceptible under a few variants of our optimization framework.

Answer to RQ3: Not entirely. Unlike CKA, we are unable to drive SVCCA toward zero while preserving functional alignment or maintain high SVCCA under severe functional unalignment, suggesting that SVCCA is less susceptible to these counterfactual optimizations.

Additional ablation studies: In §A.1, we present additional ablations on cross-architecture optimization (Figure 8) and tabular data (Figures 9 to 12). We additionally evaluate the effect of progressively constraining more parameter tensors across model architectures and datasets. In §A.2, we present results for ResNet-152 on CIFAR-10, CIFAR-100, Tiny-ImageNet, and MNIST; in §A.3, we present results for ResNet-50 on CIFAR-100 and Tiny-ImageNet; and in §A.4, we present results for ResNet-18 on CIFAR-10. Finally, we extend these experiments to vision transformers: §A.5 reports ViT-Base results on CIFAR-100 and Tiny-ImageNet, while §A.6 and §A.7 report results for ViT-Small and ViT-Tiny, respectively. Collectively, these results show that our ability to manipulate representational similarity is not specific to a single architecture, dataset, or degree of parameter constraint.

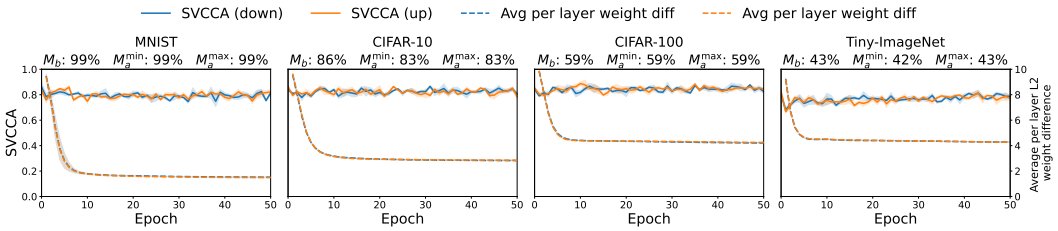
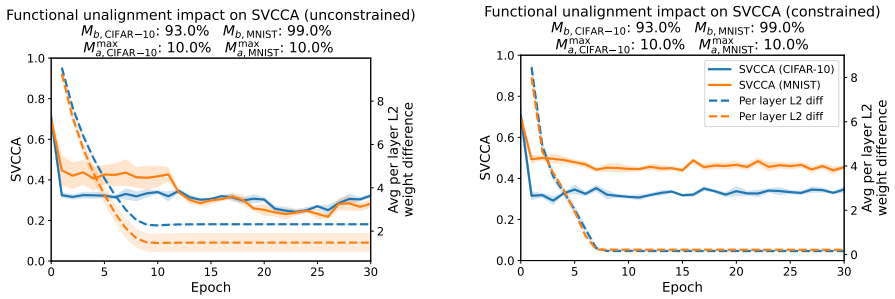


Figure 6: **Unconstrained optimization has a minimal effect on SVCCA between two functionally aligned models, effecting little change from baseline values.** *SVCCA measured in the penultimate model layer; results reported over 3 independent runs with random using ResNet-18.*



(a) **Unconstrained** functional unalignment for MNIST and CIFAR10 using SVCCA.

(b) **Constrained** functional unalignment for MNIST and CIFAR10 using SVCCA.

Figure 7: **In the unconstrained setting (left side), SVCCA decreases as the performance of $M_{\mathbf{a}}^{\max}$ degrades. Under parameter constraints (right side), SVCCA remains more stable as task performance decreases.** *Results are reported over $n = 3$ independent runs with different random seeds using ResNet-50.*

4 Related Work

Uses of representation similarity metrics: One popular application of representation similarity metrics is model stitching (Bansal et al., 2021), in which these metrics help determine whether two neural networks can spliced together while preserving task performance. More recently, Huh et al. (2024) used CKA and other metrics to show that language and vision models have aligned representations. High representational similarity scores have been linked to the transferability of adversarial examples and jailbreak attacks between ML models (Li et al., 2025). Angell et al. (2025) further showed that LLMs with higher representational similarity are more vulnerable to the same jailbreaks. Finally, Zhang et al. (2025) propose using CKA to "fingerprint" LLMs and detect potentially distilled variants via their representational similarity to the original model.

Reliability of representation similarity metrics: One prior work has probed the reliability of representation similarity metrics when comparing layers *of the same model*. Davari et al. (2022) identify several weaknesses in linear CKA that allow substantial changes to the similarity score without much effect on model performance. First, they show that CKA is sensitive to outliers—even a single deviation can cause the CKA score between two representations to drop sharply. Second, they find that CKA can be sensitive to transformations that preserve linear separability, so two representations can have near-identical linear decision boundaries but low CKA after a translation or related modification. Finally, they report counterintuitive empirical behavior, including high measured CKA similarity between early layers of trained vs. untrained networks. These suggest that CKA does not provide reliable evidence of meaningful representational correspondence between layers of the same model. More broadly, Gröger et al. (2026) identify that increasing model depth or width can inflate representational similarity scores, calling into question the use of uncalibrated similarity scores as evidence for the Platonic Representation Hypothesis of (Huh et al., 2024).

Vulnerabilities in model interpretation: Nanfack et al. (2024) propose adversarial attacks on interpretation using activation maximization. They fine-tune a model so that attacker-chosen decoy inputs become the highest-activating examples for a selected layer or channel, while keeping task performance essentially unchanged. Although their objective targets interpretation within a single

model rather than comparative similarity across models, it similarly explores how representation analyses can be steered via optimization without notable effects on model performance. Beyond adversarial attacks, Smith et al. (2025) examine model stitching as a functional alignment method for comparing representations and show that "stitchability" does not necessarily imply that two models encode similar information. Models with substantially different biases, including those trained on different tasks including random noise, can still be stitched into trained models. Their results suggest that even when an alignment-based comparison succeeds without adversarial optimization, the resulting notion of similarity can still be misleading.

Reliability of between-model similarity metrics: Collectively, this literature shows that representational similarity metrics, while sometimes useful, can behave in surprising ways and do not always track functional similarity. For example, Ciernik et al. (2025) find that the relationship between representational similarity and downstream task performance depends on the evaluation dataset. Klabunde et al. (2025b) argue that representational similarity has no single ground truth, and propose a benchmark that evaluates similarity measures across architectures, datasets, and domains. We build on these observations by explicitly demonstrating metrics' fragility.

5 Discussion

The ability to steer representational similarity metrics may lie in the geometry of overparametrized models and their corresponding loss surfaces. Near a good solution, there may still be many functionally similar solutions because overparameterization allows the training process flexibility to select among a large set of high-performing fits (Belkin, 2021). In fact, even in local minima that appear sharp (Dinh et al., 2017), there can exist directions that preserve model utility, but still leave enough room for our adversarial optimization to succeed, even under parameter constraints.

Overall, our results suggest that high representational similarity between two models does not necessarily indicate model similarity from a task perspective and *vice versa*. This is consistent with Klabunde et al. (2025b), who found that across many tests no similarity metric consistently performed best. Such metrics should thus be supplemented with measurements of other model characteristics (*e.g.* performance, generalization, etc.) to give deeper assurance of similarity. Reducing something as complex as "model similarity" down to one number is, not unsurprisingly, insufficient and unreliable.

Limitations: Although both CKA and SVCCA can be steered via optimization, we do not establish theoretically why this is possible or why their susceptibility differs. For CKA, perhaps the perturbations found during optimization reshape a layer's feature space—*e.g.* by anisotropically rescaling feature directions or redistributing activation variance—in a way that affects CKA but not task accuracy. One possible explanation is that since CKA is not invariant to arbitrary invertible linear transformations, it is possible that our optimization exploits this property. Given that our approach works across complex architectures and for arbitrary layers, however, this is likely not the sole reason for its success. In contrast, we believe SVCCA could be less susceptible by construction due to its use of singular value decomposition (SVD). SVD effectively performs dimension reduction, discarding lower variance directions, and this perhaps makes SVCCA less vulnerable to small perturbations introduced during optimization. However, our paper does not provide any definitive analysis supporting these speculations, and different optimization approaches may succeed on SVCCA.

Future work: Our framework can also be directly used to simultaneously optimize the representational similarity of several layers at once as well as to only *partially* align or unalign models. It can also be extended to other differentiable metrics such as CKNNA (Huh et al., 2024). Further work is also needed to investigate the theoretical aspects of the adversarial vulnerabilities in the metrics we study, so we can understand more precisely when misleading conclusions can arise. Our work only considered image and tabular datasets, so future work could consider other modalities and model architectures. Finally, future work could construct new representation similarity metrics that build on strengths of existing metrics and mitigate issues found in this paper.

AI USE STATEMENT

In this work, we did not use generative AI tools for any tasks with required disclosure (such as theoretical modeling, proof writing, method implementation, synthetic data generation, or result interpretation). Additionally, we used generative AI tools (LLMs) for tasks with recommended disclosure: identifying relevant literature to expand our related work survey, and generating boilerplate visualization scripts (such as plots and heatmaps).

We have thoroughly reviewed all AI-assisted work. All papers suggested by LLMs were manually verified, downloaded, and read by the authors to confirm their relevance and avoid hallucinated citations. All visualization and debugging code was manually verified and tested for accuracy before running experiments. We take full responsibility for the final content of this work, including all text, code, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This paper concerns adversarial modifications to models which may be used to elide model fingerprinting techniques. However, this is in line with the open-sourcing of both attacks and defenses against ML models in the past Carlini & Wagner (2017), and thus does not raise any serious ethical concerns.

REPRODUCIBILITY STATEMENT

Our code has been made available here: https://anonymous.4open.science/r/frag_repr_similarity-B1B7/README.md. All hyperparameters are public, and multiple runs were carried out for statistical validity of results.

REFERENCES

Rico Angell, Jannik Brinkmann, and He He. Jailbreak transferability emerges from shared representations, 2025. URL https://arxiv.org/abs/2506.12913.

Francis Bach. *Learning theory from first principles*, volume 2. MIT press Cambridge, 2024.

Yamini Bansal, Preetum Nakkiran, and Boaz Barak. Revisiting model stitching to compare neural representations. *Advances in Neural Information Processing Systems*, 2021.

Mikhail Belkin. Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation. *Acta Numerica*, 30:203–248, 2021.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

Jock A. Blackard and Denis J. Dean. Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Computers and Electronics in Agriculture*, 24(3):131–151, 1999.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (sp)*, pp. 39–57. Ieee, 2017.

Christian Cianfarani, Arjun Nitin Bhagoji, Vikash Sehwag, Ben Zhao, Haitao Zheng, and Prateek Mittal. Understanding robust learning through the lens of representation similarities. *Advances in Neural Information Processing Systems*, 35:34912–34925, 2022.

Laure Ciernik, Lorenz Linhardt, Marco Morik, Jonas Dippel, Simon Kornblith, and Lukas Muttenthaler. Objective drives the consistency of representational similarity across datasets, 2025. URL https://arxiv.org/abs/2411.05561.

MohammadReza Davari, Stefan Horoi, Amine Natik, Guillaume Lajoie, Guy Wolf, and Eugene Belilovsky. Reliability of cka as a similarity measure in deep learning. In *The Eleventh International Conference on Learning Representations*, 2022.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Frances Ding, Jean-Stanislas Denain, and Jacob Steinhardt. Grounding representation similarity through statistical testing. *Advances in Neural Information Processing Systems*, 34:1556–1568, 2021.

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, pp. 1019–1028. PMLR, 2017.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Kshitij Dwivedi and Gemma Roig. Representation similarity analysis for efficient task taxonomy & transfer learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12387–12396, 2019.

Feng-Lei Fan, Jinjun Xiong, Mengzhou Li, and Ge Wang. On interpretability of artificial neural networks: A survey. *IEEE Transactions on Radiation and Plasma Medical Sciences*, 5(6):741–760, 2021.

Thomas Goerttler and Klaus Obermayer. Similarity of pre-trained and fine-tuned representations, 2022. URL https://arxiv.org/abs/2207.09225.

Fabian Gröger, Shuo Wen, and Maria Brbić. Revisiting the platonic representation hypothesis: An aristotelian view, 2026. URL https://arxiv.org/abs/2602.14486.

Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Lei Hsiung, Tianyu Pang, Yung-Chen Tang, Linyue Song, Tsung-Yi Ho, Pin-Yu Chen, and Yaoqing Yang. Why llm safety guardrails collapse after fine-tuning: A similarity analysis between alignment and fine-tuning datasets. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16601–16619, 2026.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation hypothesis, 2024. URL https://arxiv.org/abs/2405.07987.

Max Klabunde, Tobias Schumacher, Markus Strohmaier, and Florian Lemmerich. Similarity of neural network models: A survey of functional and representational measures. *ACM Computing Surveys*, 57(9):1–52, 2025a.

Max Klabunde, Tassilo Wald, Tobias Schumacher, Klaus Maier-Hein, Markus Strohmaier, and Florian Lemmerich. Resi: A comprehensive benchmark for representational similarity measures, 2025b. URL https://arxiv.org/abs/2408.00531.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pp. 3519–3529. PMlR, 2019.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). *URL http://www. cs. toronto. edu/kriz/cifar. html*, 5(4):1, 2010.

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

Lingyu Li, Yan Teng, Yingchun Wang, and Xia Hu. Representational alignment yields generalizable safety in language models, 2026. URL https://arxiv.org/abs/2609.04022.

Zeyu Michael Li, Hung Anh Vu, Damilola Awofisayo, and Emily Wenger. Causes and consequences of representational similarity in machine learning models, 2025. URL https://arxiv.org/abs/2505.13899.

Ari Morcos, Maithra Raghu, and Samy Bengio. Insights on representational similarity in neural networks with canonical correlation. *Advances in Neural Information Processing Systems*, 31, 2018.

Geraldin Nanfack, Alexander Fulleringer, Jonathan Marty, Michael Eickenberg, and Eugene Belilovsky. Adversarial attacks on the interpretation of neuron activation maximization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 4315–4324, 2024.

Jason Phang, Haokun Liu, and Samuel Bowman. Fine-tuned transformers show clusters of similar representations across layers. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pp. 529–538, 2021.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *Advances in Neural Information Processing Systems*, 30, 2017.

Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296*, 2017.

Damian Smith, Harvey Mannering, and Antonia Marcu. Functional alignment can mislead: Examining model stitching. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=glLqTK9En3.

Ross Wightman, Nathan Raw, Alexander Soare, Aman Arora, Chris Ha, Christoph Reich, Fredo Guan, Jakub Kaczmarzyk, mrT23, Mike, SeeFun, contrastive, Mohammed Rizin, Hyeongchan Kim, Csaba Kertész, Dushyant Mehta, Guillem Cucurull, Kushajveer Singh, hankyul, Yuki Tatsunami, Andrew Lavin, Juntang Zhuang, Matthijs Hollemans, Mohamed Rashad, Sepehr Sameni, Vyacheslav Shults, Lucain, Xiao Wang, Yonghye Kwon, and Yusuke Uchida. rwightman/pytorch-image-models: v0.8.10dev0 release, February 2023. URL https://doi.org/10.5281/zenodo.7618837.

Jie Zhang, Dongrui Liu, Chen Qian, Linfeng Zhang, Yong Liu, Yu Qiao, and Jing Shao. Reef: Representation encoding fingerprints for large language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

Yu Zhang, Peter Tiňo, Aleš Leonardis, and Ke Tang. A survey on neural network interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 5(5):726–742, 2021.

A APPENDIX

The appendix is organized as follows:

1. Table 1, Table 2, and Table 3 summarize the experimental setup, parameter cutoffs, and all possible scenarios under our constrained optimization framework, respectively.
2. §A.1 presents additional ablations and extensions of our optimization framework, cross-architecture optimization shown in Figure 8, experiments on tabular data shown in Figures 9 to 12, and the complementary parameter-distance setting shown in Figure 13.
3. §A.2: results for ResNet-152 on CIFAR-10, CIFAR-100, Tiny-ImageNet, and MNIST.
4. §A.3: results for ResNet-50 on CIFAR-100 and Tiny-ImageNet.
5. §A.4: results for ResNet-18 on CIFAR-10.
6. §A.5, §A.6, §A.7: results for vision transformers ViT-Base, ViT-Small, and ViT-Tiny respectively on CIFAR-100 and Tiny-Imagenet.

Table 1: **Summary of our experiment setup.** *Models are trained for 50 epochs using the AdamW optimizer and cross-entropy loss, 3 random seeds per experiment.*

Setup element	Used in our experiments
Datasets	MNIST, CIFAR-10, CIFAR-100, Tiny-ImageNet, CoverType
ResNet architectures	ResNet-18, ResNet-50, ResNet-152
ViT architectures	ViT-Tiny/16, ViT-Small/16, ViT-Base/16
MLP architectures	3-layer and 5-layer with ReLU activations

Table 2: **Layers' parameters cutoffs.** *Each index i includes tensors* 1 *through i, inclusive.*

Architecture	Parameter Tensor Index
ResNet-18	$\{1, 10, 20, 30, 40, 50, 60, 62\}$
ResNet-50	$\{1, 28, 56, 84, 112, 130, 158, 161\}$
ResNet-152	$\{1, 80, 160, 240, 320, 400, 467\}$
ViT architectures	$\{1, 28, 56, 84, 112, 130, 152\}$

Table 3: **All possible scenarios under our constrained optimization framework**. *Highlighted settings correspond to Equations 7 and 8, our primary focus in this paper.*

Case	CKA	CE Loss	Parameter Distance	Optimization Objective	Interpretation
1	Low	Low	Close	$\min_{\theta_a} metric + \lambda\mathcal{L} + \gamma\theta$	Dissimilar representations, aligned function, close parameters
2	Low	Low	Far	$\min_{\theta_a} metric + \lambda\mathcal{L} - \gamma\theta$	Dissimilar representations, aligned function, distant parameters
3	Low	High	Close	$\min_{\theta_a} metric - \lambda\mathcal{L} + \gamma\theta$	Dissimilar representations, unaligned function, close parameters
4	Low	High	Far	$\min_{\theta_a} metric - \lambda\mathcal{L} - \gamma\theta$	Dissimilar representations, unaligned function, distant parameters
5	High	Low	Close	$\max_{\theta_a} metric - \lambda\mathcal{L} - \gamma\theta$	Similar representations, aligned function, close parameters
6	High	Low	Far	$\max_{\theta_a} metric - \lambda\mathcal{L} + \gamma\theta$	Similar representations, aligned function, distant parameters
7	High	High	Close	$\max_{\theta_a} metric + \lambda\mathcal{L} - \gamma\theta$	Similar representations, unaligned function, close parameters
8	High	High	Far	$\max_{\theta_a} metric + \lambda\mathcal{L} + \gamma\theta$	Similar representations, unaligned function, distant parameters

As shown in Table 3, our primary focus is on Cases 1 and 7, which form our key representational similarity counterfactuals. Cases 2 and 8 effectively capture the same counterfactuals without parameter constraints, and largely represent our unconstrained setting. Cases 3-6 are not relevant counterfactuals for our analysis. Cases 3 and 6 capture counterfactual relationships in parameter space, while cases 4 and 5 represent settings where representational similarity, functional similarity, and parameter distance are mutually consistent.

A.1 ABLATIONS

Different architectures for benign and adversarial models: Figure 8 tests whether our counterfactual optimization requires the benign and adversarial models to share the same architecture. We train a ResNet-152 as the benign model M_b and optimize a ResNet-18 as the target model M_a on CIFAR-100, showing that CKA can still be driven toward low and high values while maintaining comparable task performance.

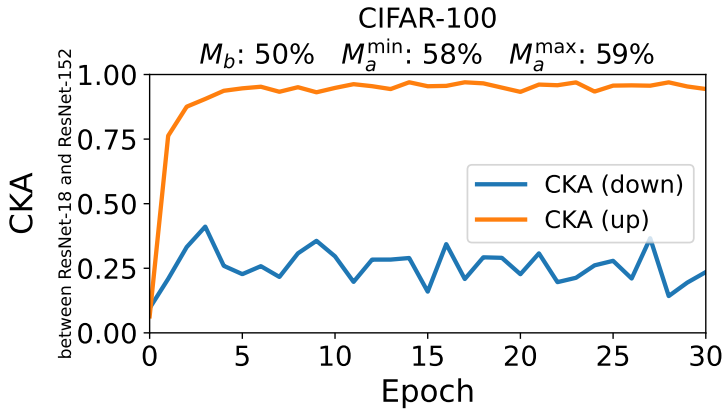


Figure 8: **Unconstrained optimization using different architectures for reference and target models.** We repeat the unconstrained representation optimization experiment using a ResNet-152 benign model (M_b) and a ResNet-18 adversarial model (M_a) on CIFAR-100. Despite the architectural difference between the models, optimization can drive CKA toward both low (down) and high (up) values while maintaining comparable task performance.

Tabular data: The following figures presents a similar set of experiments on tabular data with 3- and 5-layers MLPs with ReLU activations. This allows us to probe for the behavior of this optimization in a low parameter setting. We conduct our experiments on the tabular Covertype dataset. Figures 9 and 10 show that the *min* direction optimization is effective for different layers of the 5-layer MLP, and Figures 11 and 12 show the same for the 3-layer MLP. Accuracies for all models across both architectures (benign and adversarial) lie within 94 and 96%.

Parameter distance counterfactual: Figure 13 shows that parameter distance can increase over time while representational similarity and functional alignment remain high.

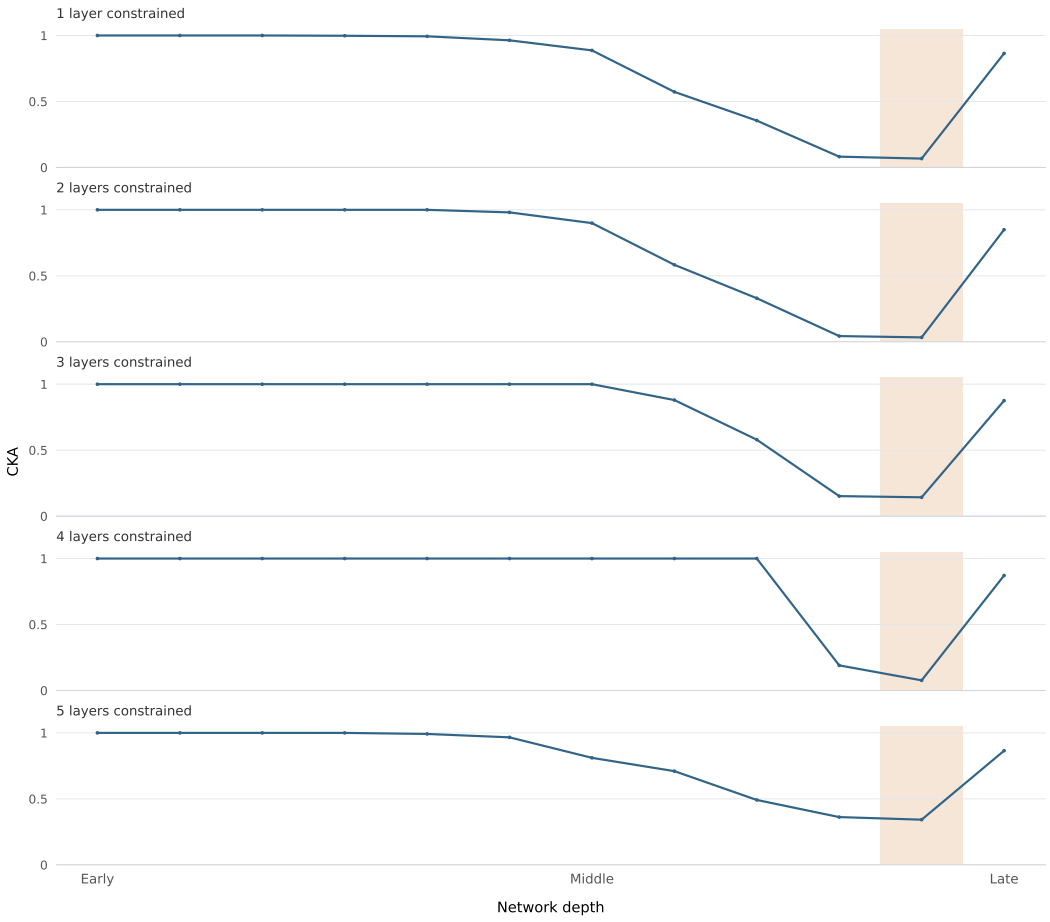


Figure 9: Results for increasing layer constraints in **min** direction for 5 hidden layer MLP on Covertype dataset targeting the penultimate layer.

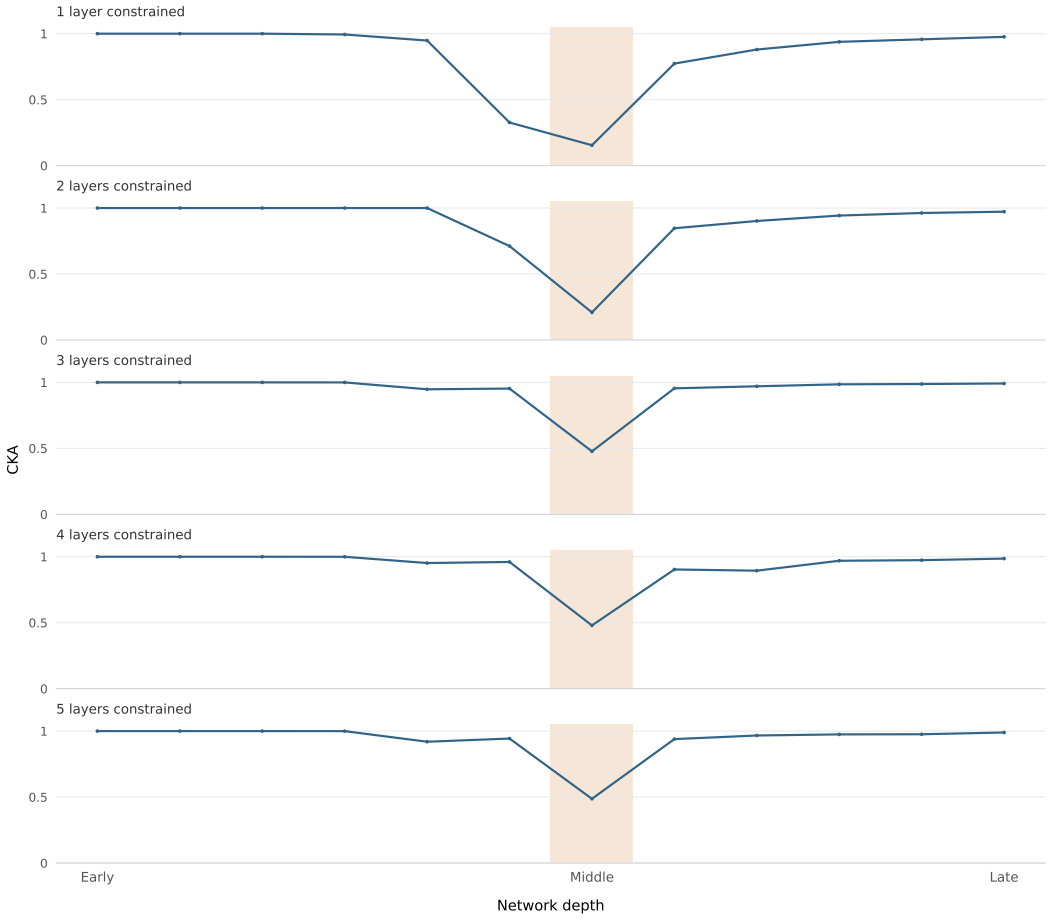


Figure 10: Results for increasing layer constraints in **min** direction for 5 hidden layer MLP on Covertype dataset targeting the middle layer.

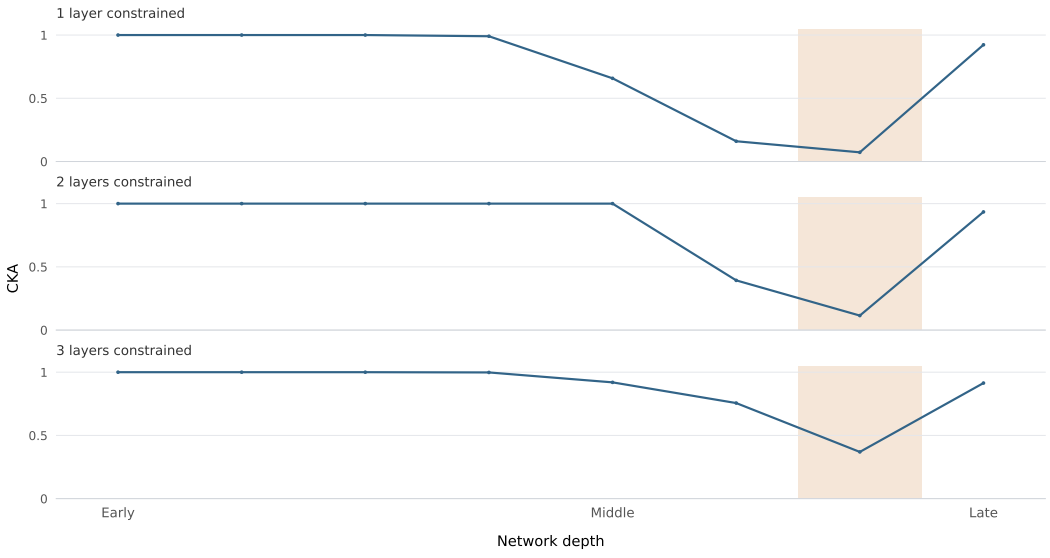


Figure 11: Results for increasing layer constraints in **min** direction for 3 hidden layer MLP on Covertype dataset targeting the penultimate layer.

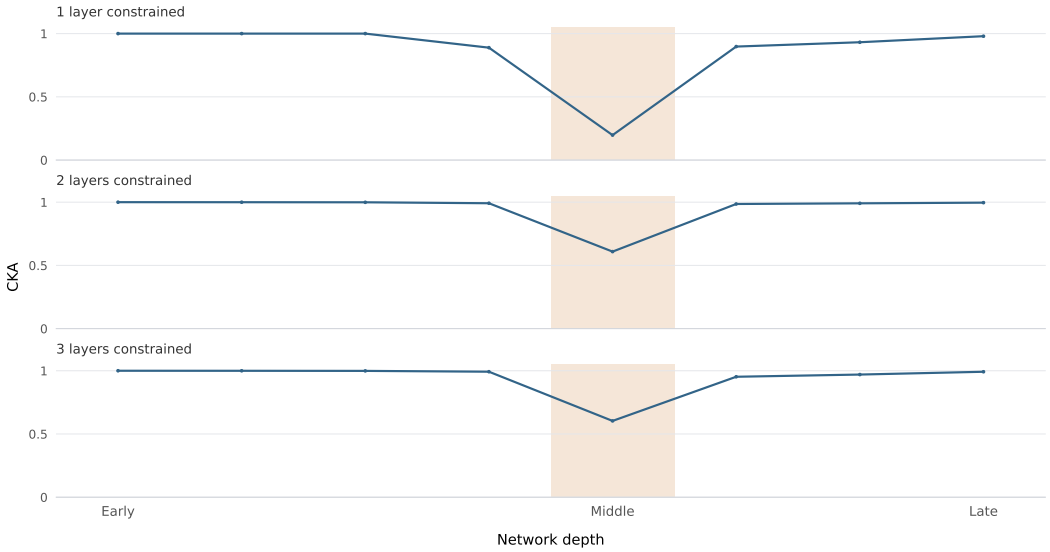
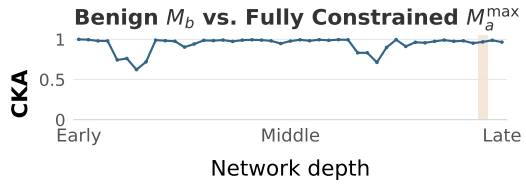
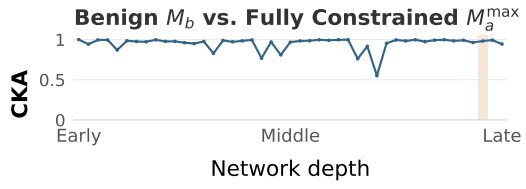


Figure 12: Results for increasing layer constraints in **min** direction for 3 hidden layer MLP on Covertype dataset targeting the middle layer.



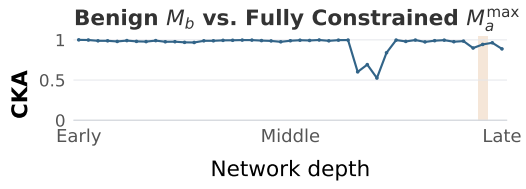
Metric	Epoch 1		Epoch 50
M_b Test Acc.			92.5%
$M_a^{\min}$ Test Acc.	44.9%	$\rightarrow$	92.1%
$\|M_b - M_a^{\max}\|_2$	28618.64	$\rightarrow$	32354.76

(a) **CIFAR-10**



Metric	Epoch 1		Epoch 50
M_b Test Acc.			72.1%
$M_a^{\min}$ Test Acc.	15.7%	$\rightarrow$	72.0%
$\|M_b - M_a^{\max}\|_2$	29016.85	$\rightarrow$	81757.31

(b) **CIFAR-100**



Metric	Epoch 1		Epoch 50
M_b Test Acc.			62.1%
$M_a^{\min}$ Test Acc.	10.0%	$\rightarrow$	55.8%
$\|M_b - M_a^{\max}\|_2$	40619.50	$\rightarrow$	44407.13

(c) **Tiny-ImageNet**

Figure 13: **A fully unconstrained adversary can have high representation similarity at the penultimate layer with the benign model while maintaining functional alignment.** This figure shows results for the setting described in Row 6 of Table 3 using ResNet-152. M_b accuracy is reported as a fixed reference value from its independent training run.

In the subsequent appendices, we present results for ResNet and ViT architectures across four datasets and increasing levels of parameter constraint while maintaining functional alignment. We omit the accuracy values here, but the models remain functionally aligned or unaligned per the setup of each experiment. The following summarizes each figure's result:

- Figure 14 shows that CKA min works for ResNet-152 on CIFAR-100 at the penultimate layer.
- Figure 15 shows that CKA max works for ResNet-152 on CIFAR-100 at the penultimate layer.
- Figure 16 shows that CKA min works for ResNet-152 on CIFAR-100 at the middle layer.
- Figure 17 shows that CKA max works for ResNet-152 on CIFAR-100 at the middle layer.
- Figure 18 shows that CKA min works for ResNet-152 on Tiny-ImageNet at the penultimate layer.
- Figure 19 shows that CKA max works for ResNet-152 on Tiny-ImageNet at the penultimate layer.
- Figure 20 shows that CKA min works for ResNet-152 on CIFAR-10 at the penultimate layer.
- Figure 21 shows that CKA max works for ResNet-152 on CIFAR-10 at the penultimate layer.
- Figure 22 shows that CKA min works for ResNet-152 on MNIST at the penultimate layer.
- Figure 23 shows that CKA max works for ResNet-152 on MNIST at the penultimate layer.
- Figure 24 shows that CKA min works for ResNet-50 on CIFAR-100 at the penultimate layer.
- Figure 25 shows that CKA min works for ResNet-50 on Tiny-ImageNet at the penultimate layer.
- Figure 26 shows that CKA min works for ResNet-18 on CIFAR-10 at the penultimate layer.
- Figure 27 shows that CKA min works for ViT-Base on CIFAR-100 at the penultimate layer.
- Figure 28 shows that CKA max works for ViT-Base on CIFAR-100 at the penultimate layer.
- Figure 29 shows that CKA min works for ViT-Base on Tiny-ImageNet at the penultimate layer.
- Figure 30 shows that CKA max works for ViT-Base on Tiny-ImageNet at the penultimate layer.
- Figure 31 shows that CKA min works for ViT-Small on Tiny-ImageNet at the penultimate layer.
- Figure 32 shows that CKA min works for ViT-Tiny on CIFAR-100 at the penultimate layer.
- Figure 33 shows that CKA min works for ViT-Tiny on CIFAR-100 at the middle layer.

A.2 RESNET-152

A.2.1 CIFAR-100



Figure 14: Results for increasing layer constraints in **min** direction for ResNet-152 on CIFAR-100 targeting the penultimate layer.

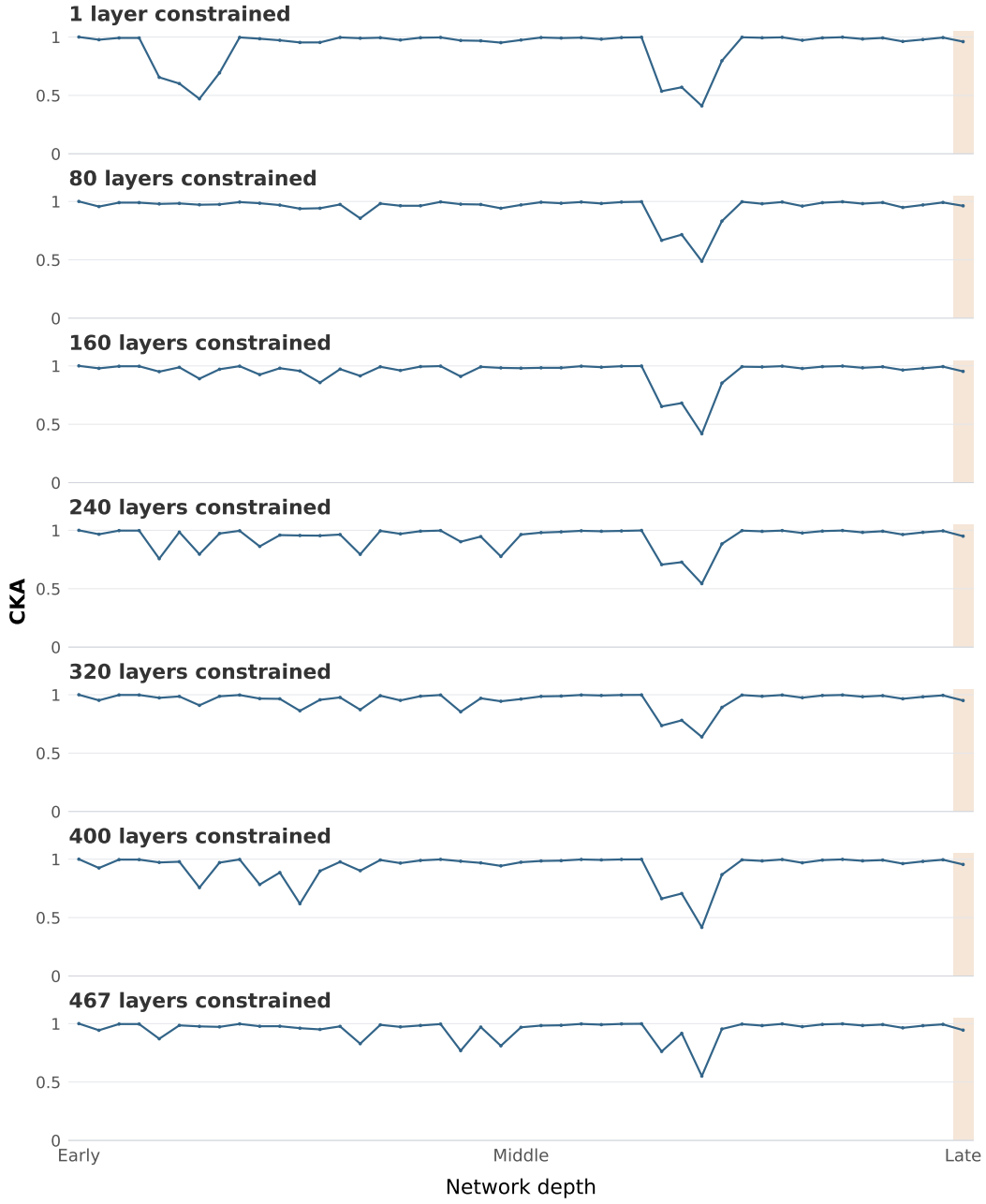


Figure 15: Results for increasing layer constraints in **max** direction for ResNet-152 on CIFAR-100 targeting the penultimate layer.



Figure 16: Results for increasing layer constraints in **min** direction for ResNet-152 on CIFAR-100 targeting a middle layer.



Figure 17: Results for increasing layer constraints in **max** direction for ResNet-152 on CIFAR-100 targeting a middle layer.

A.2.2 TINY-IMAGENET

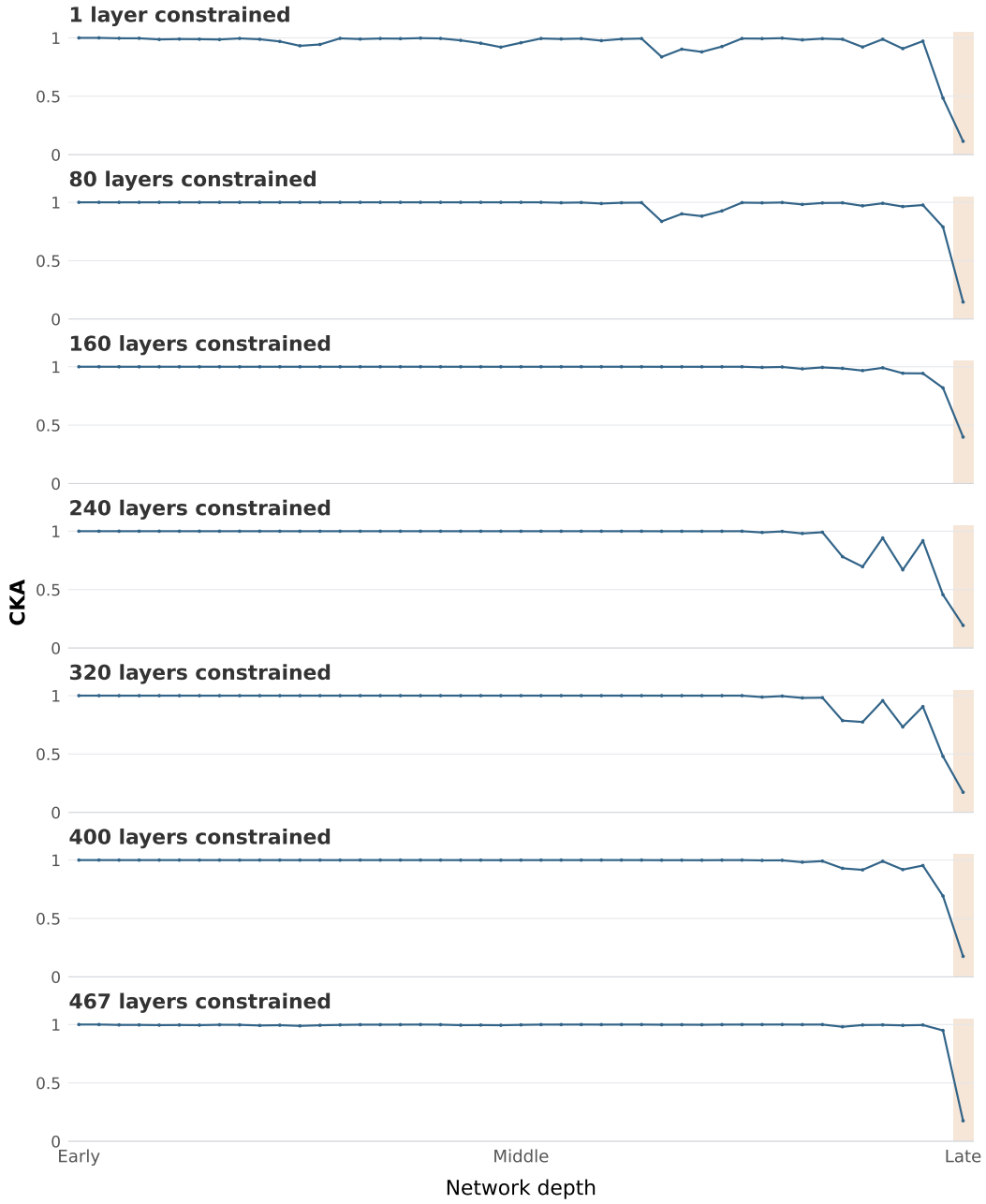


Figure 18: Results for increasing layer constraints in **min** direction for ResNet-152 on Tiny-Imagenet targeting the penultimate layer.

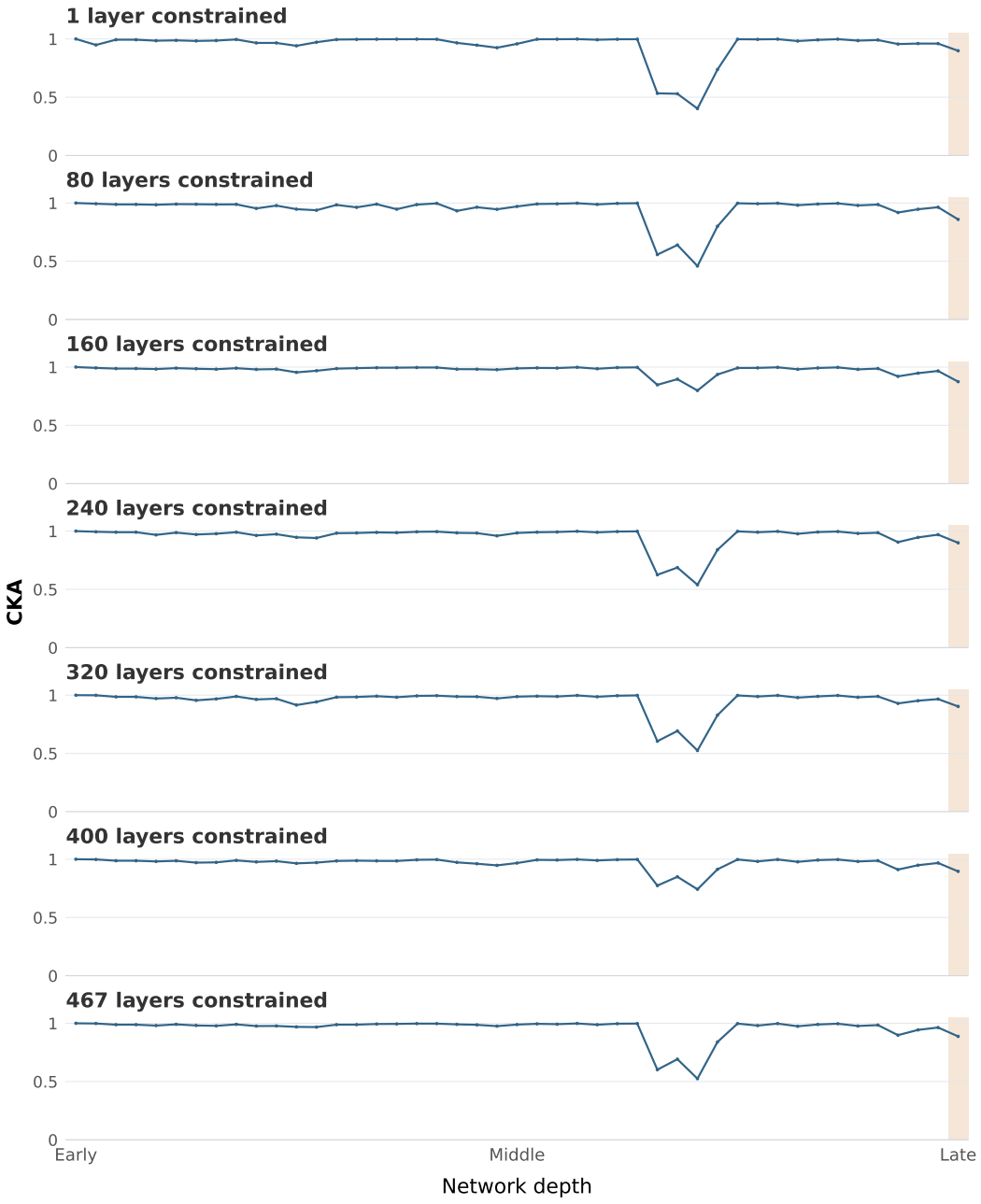


Figure 19: Results for increasing layer constraints in **max** direction for ResNet-152 on Tiny-Imagenet targeting the penultimate layer.

A.2.3 CIFAR-10

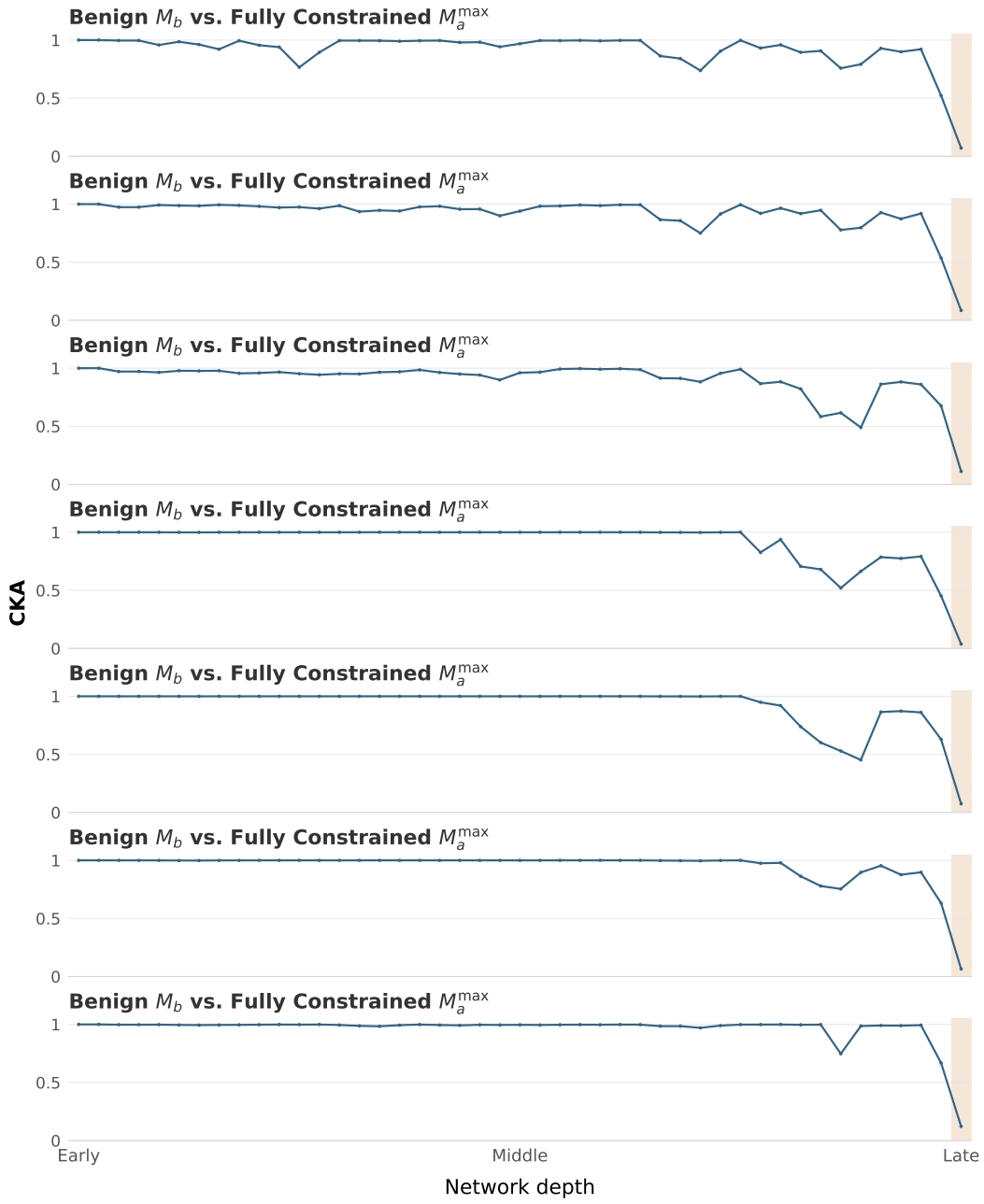


Figure 20: Results for increasing layer constraints in **min** direction for ResNet-152 on CIFAR-10 targeting the penultimate layer.

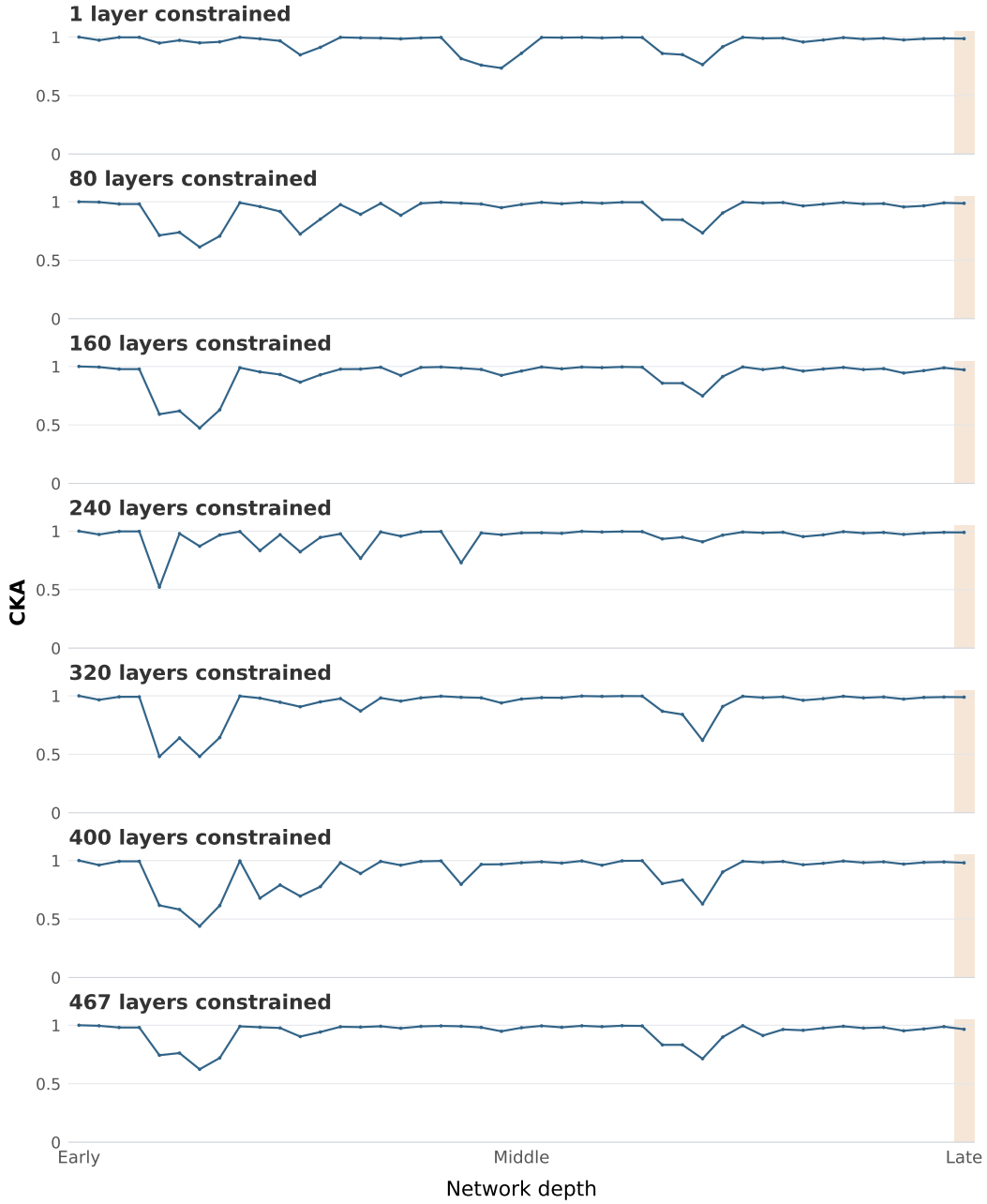


Figure 21: Results for increasing layer constraints in **max** direction for ResNet-152 on CIFAR-10 targeting the penultimate layer.

A.2.4 MNIST

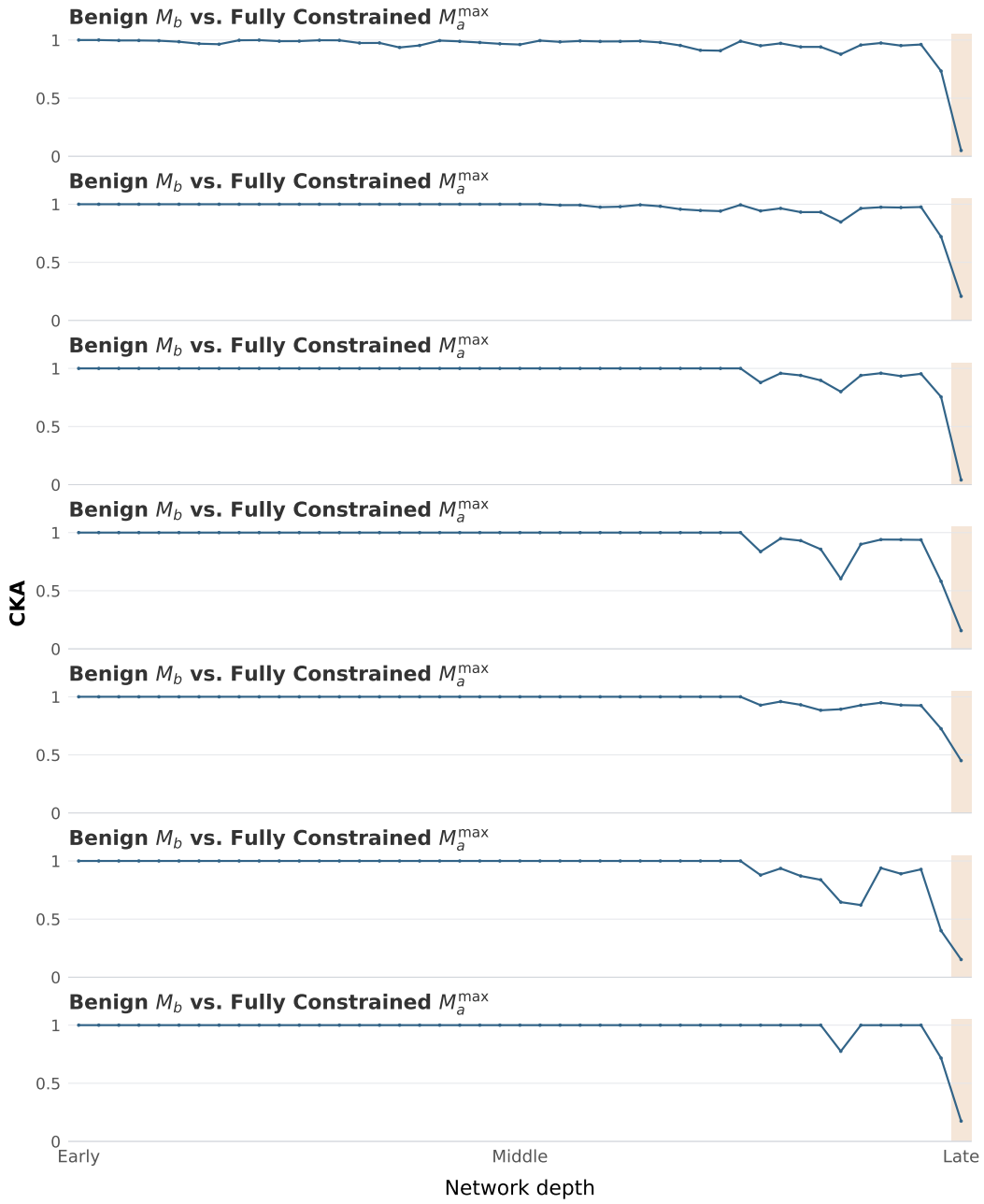


Figure 22: Results for increasing layer constraints in **min** direction for ResNet-152 on MNIST targeting the penultimate layer.

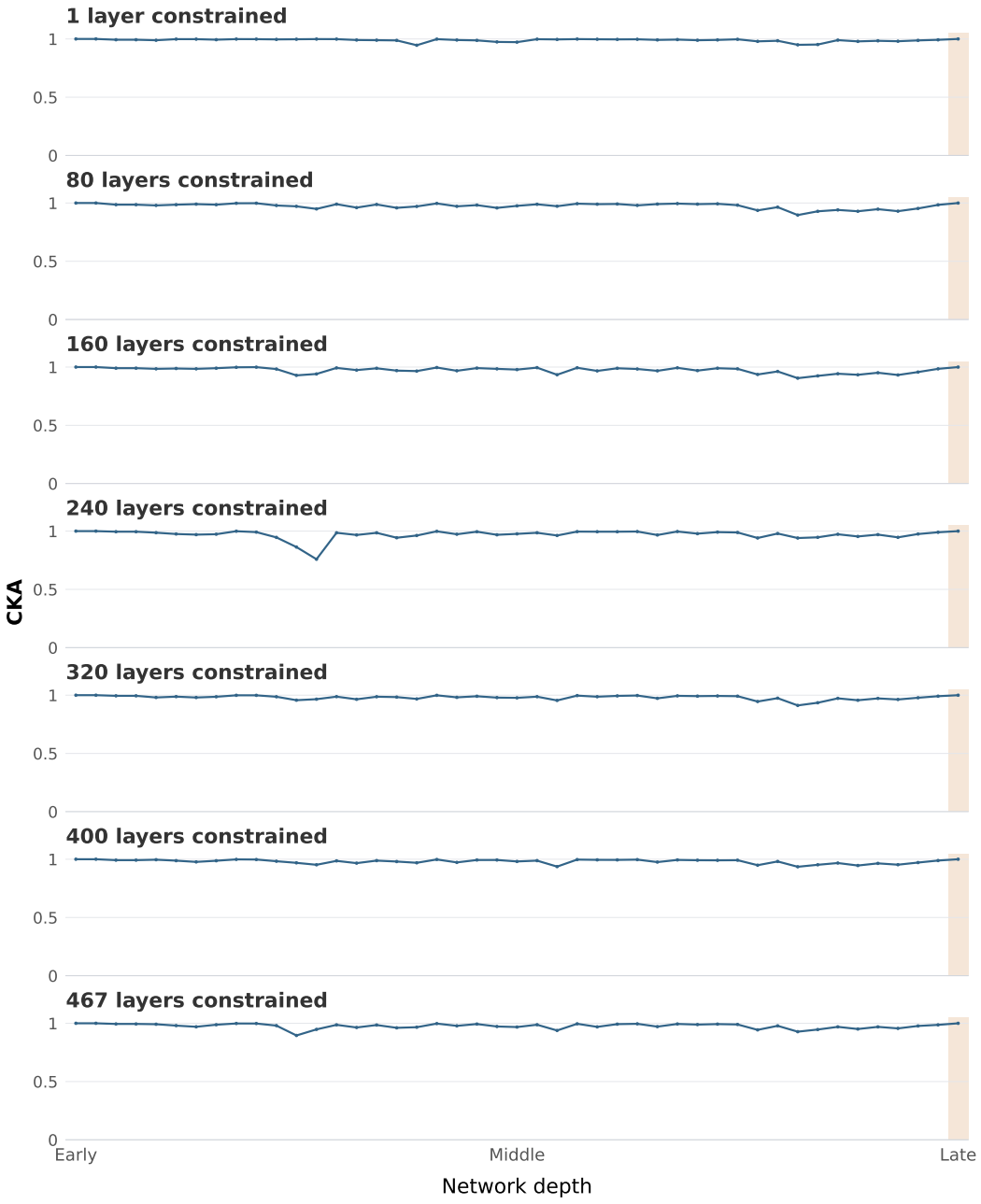


Figure 23: Results for increasing layer constraints in **max** direction for ResNet-152 on MNIST targeting the penultimate layer.

A.3 ResNet-50

A.3.1 CIFAR-100

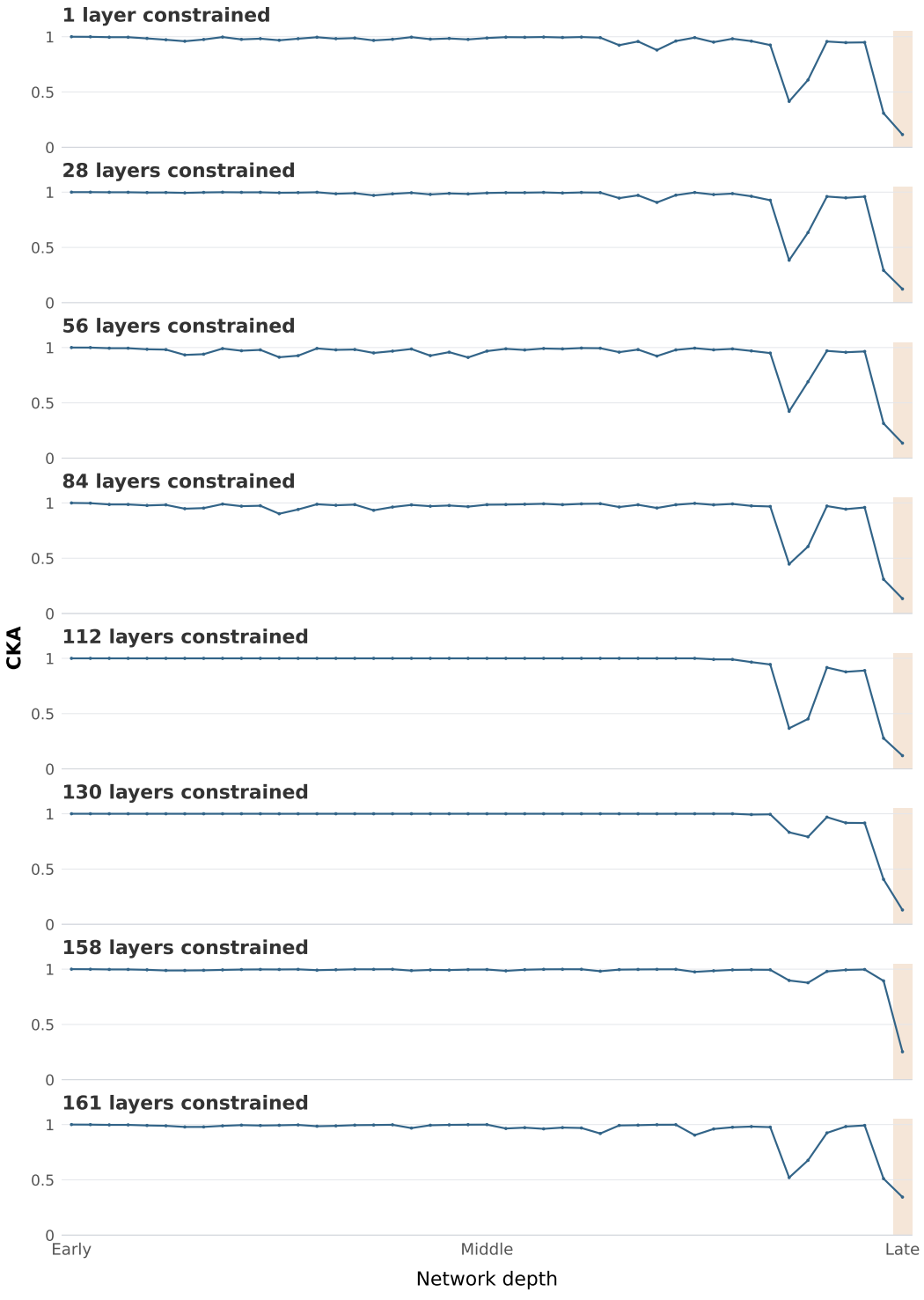


Figure 24: Results for increasing layer constraints in **min** direction for ResNet-50 on CIFAR-100 targeting the penultimate layer.

A.3.2 TINY-IMAGENET

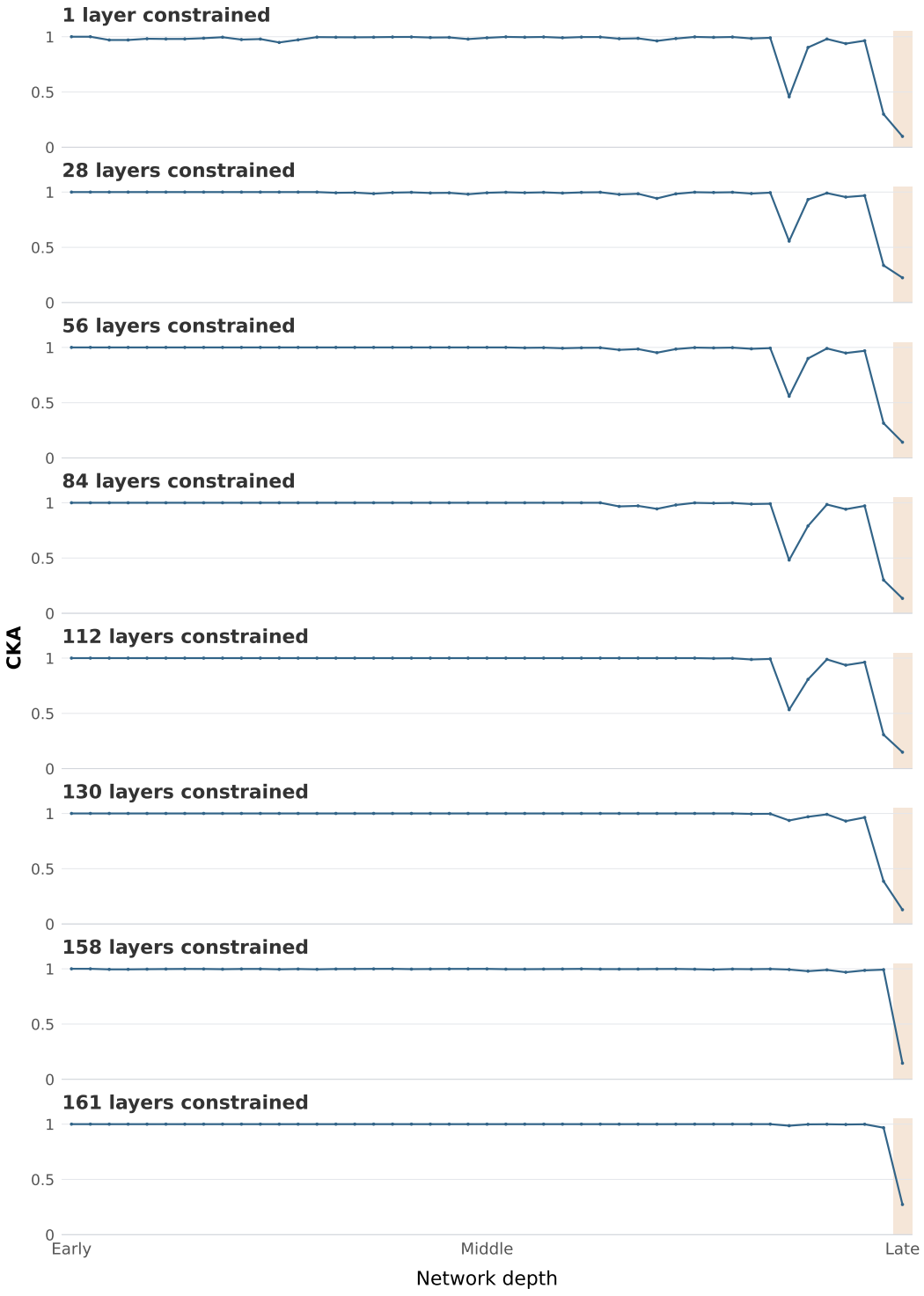


Figure 25: Results for increasing layer constraints in **min** direction for ResNet-50 on Tiny-Imagenet targeting the penultimate layer.

A.4 RESNET-18

A.4.1 CIFAR-10

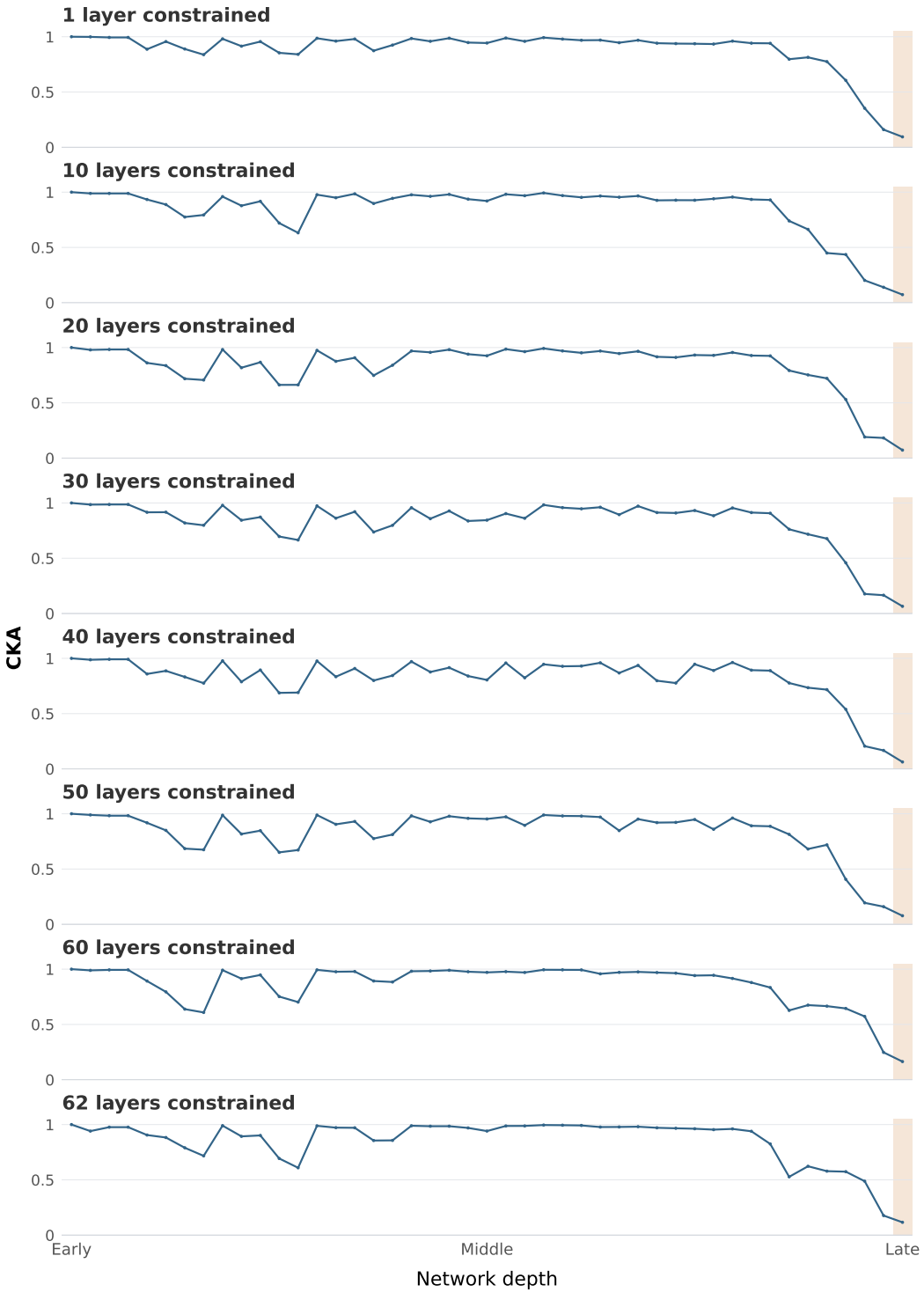


Figure 26: Results for increasing layer constraints in **min** direction for ResNet-18 on CIFAR-10 targeting the penultimate layer.

A.5 VIT-BASE-PATCH16-224

A.5.1 CIFAR-100

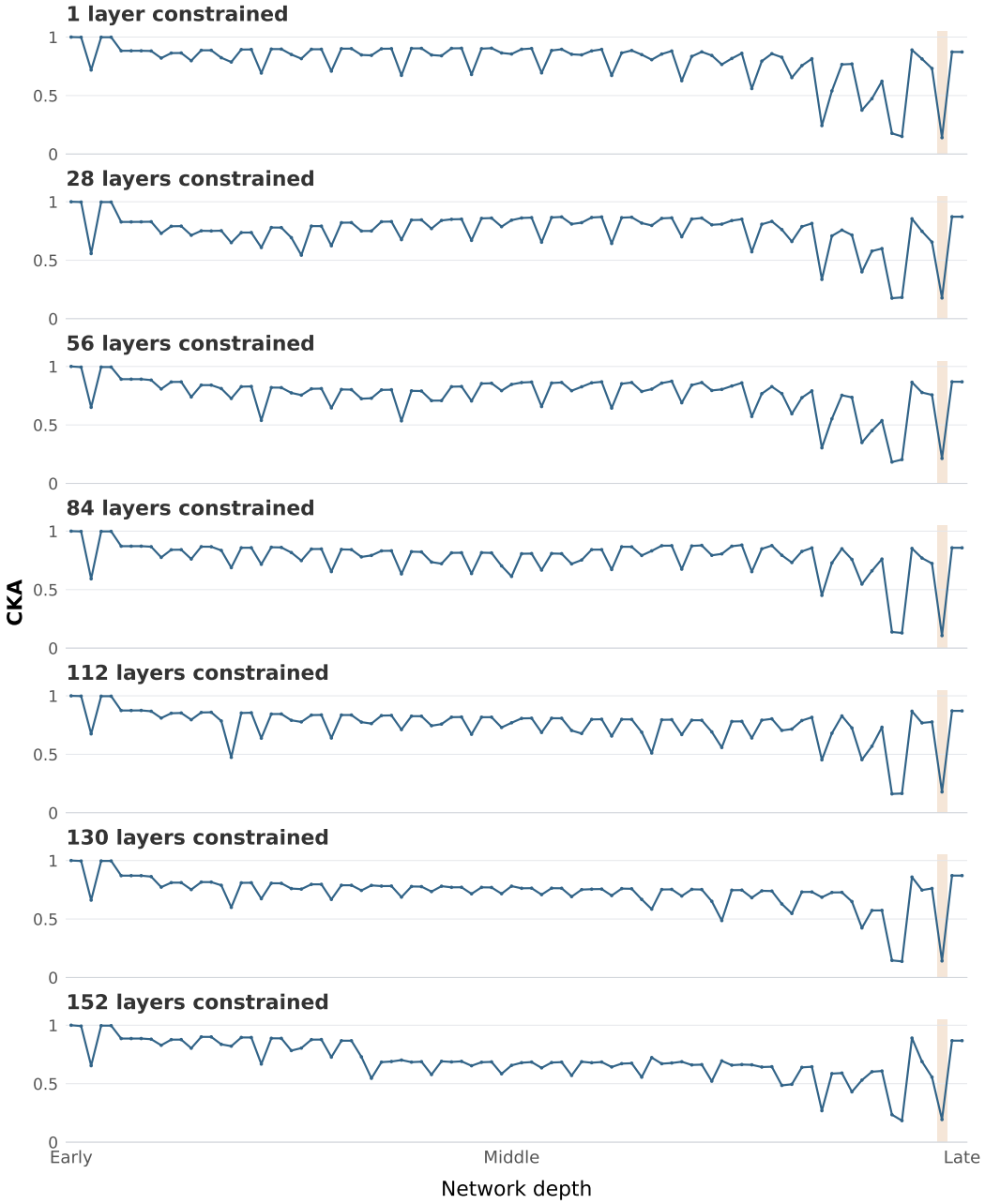


Figure 27: Results for increasing layer constraints in **min** direction for vit-base-patch16-224 on CIFAR-100 targeting the penultimate layer.

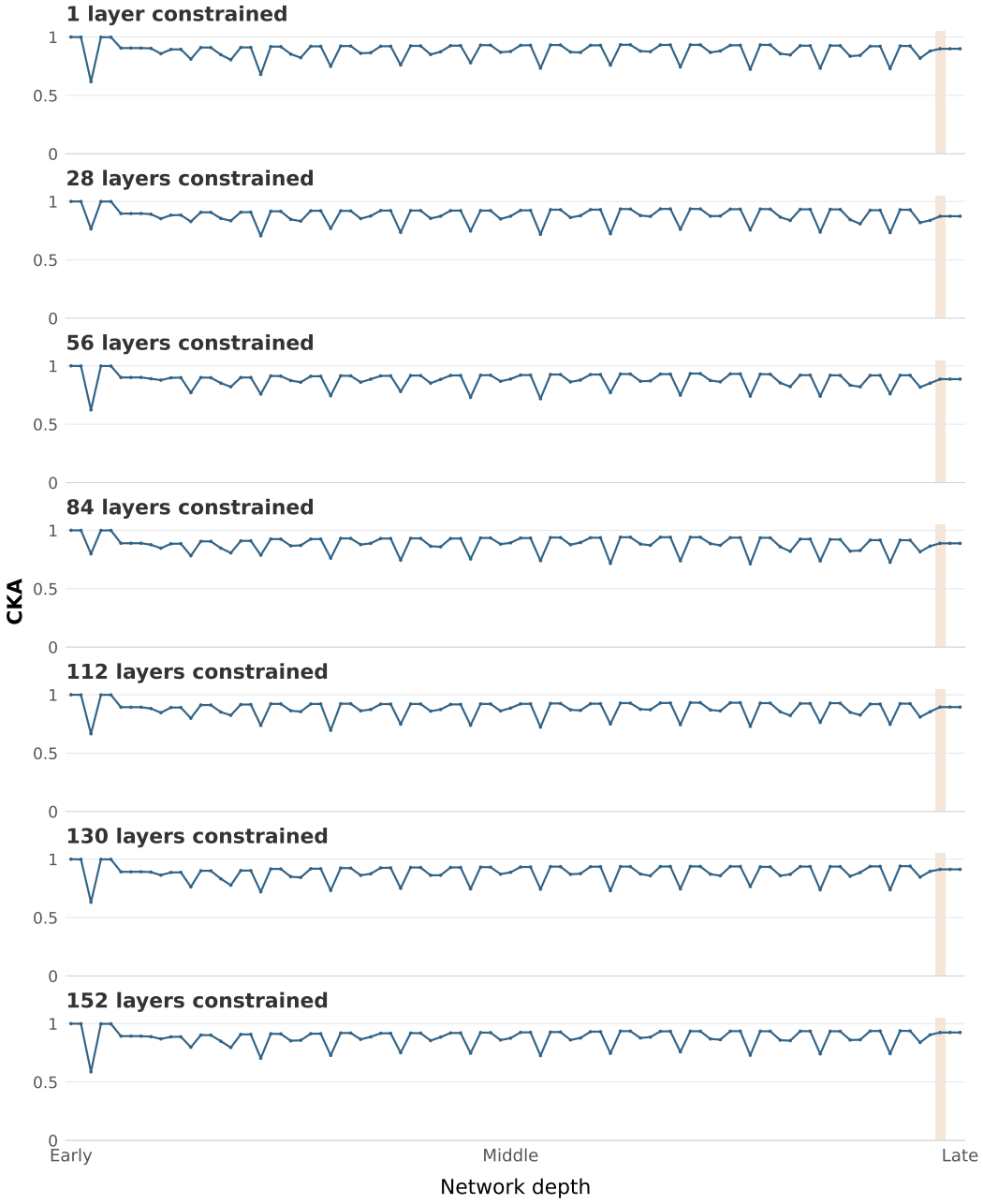


Figure 28: Results for increasing layer constraints in **max** direction for vit-base-patch16-224 on CIFAR-100 targeting the penultimate layer.

A.5.2 TINY-IMAGENET

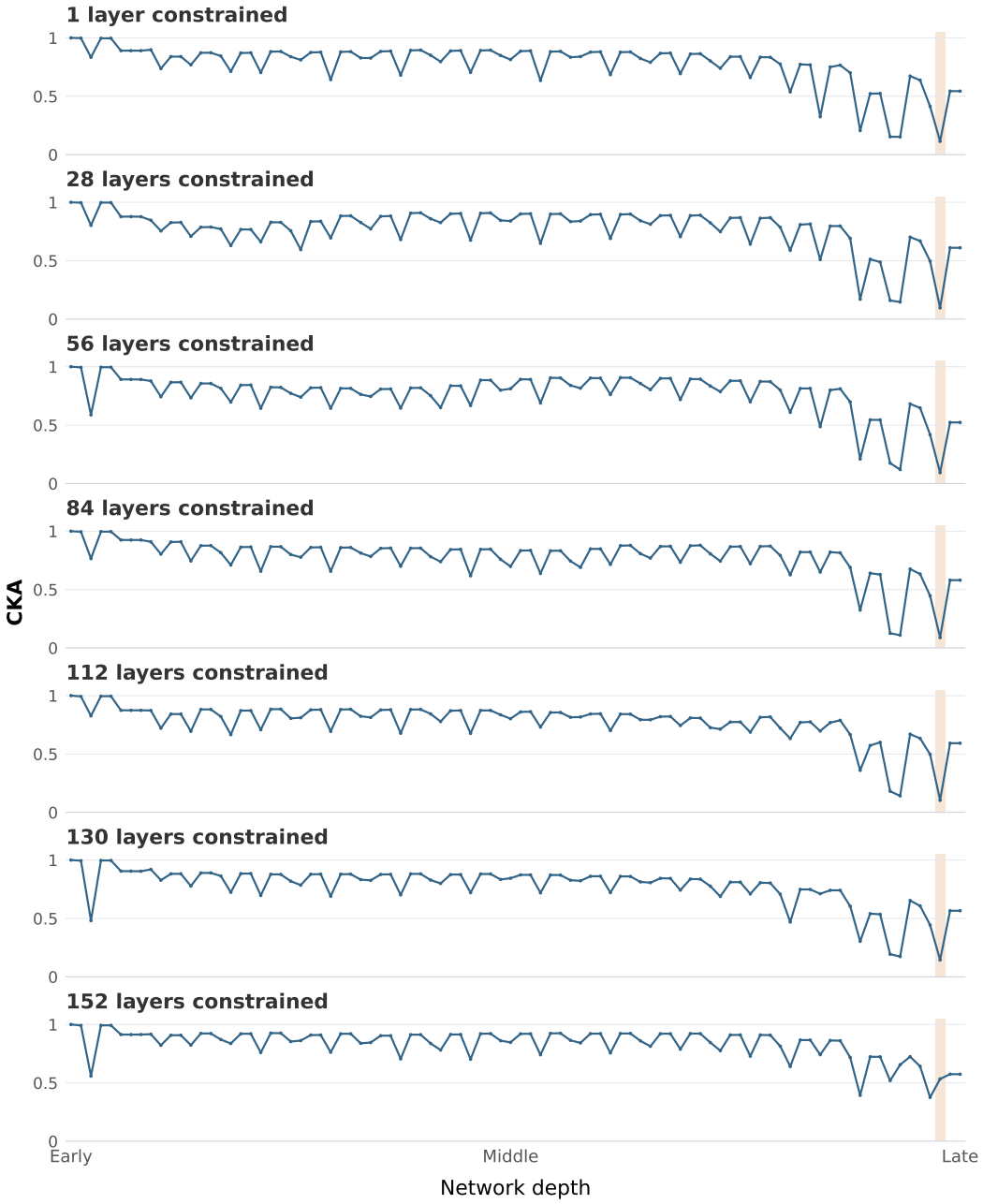


Figure 29: Results for increasing layer constraints in **min** direction for vit-base-patch16-224 on Tiny-Imagenet targeting the penultimate layer.

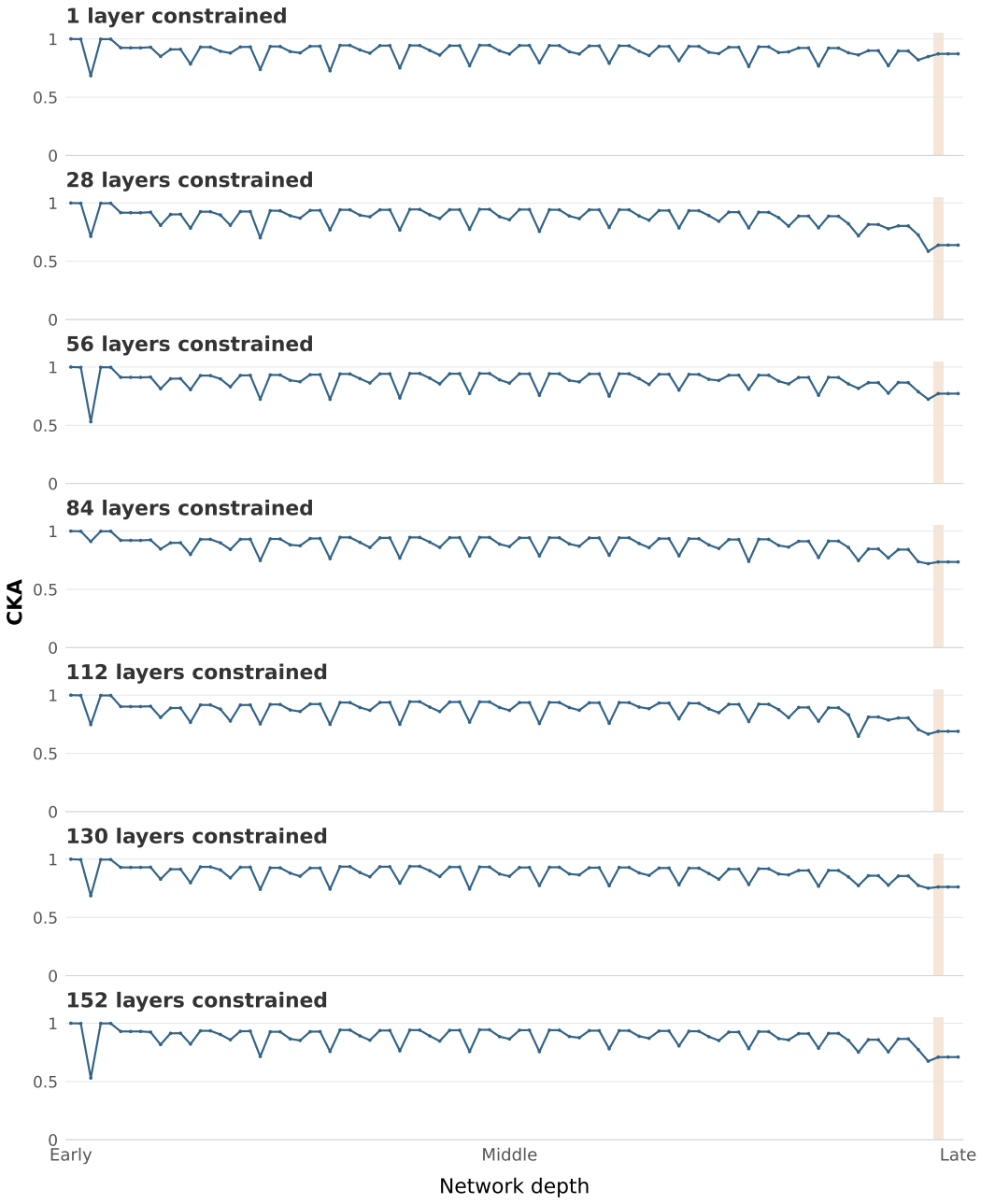


Figure 30: Results for increasing layer constraints in **max** direction for vit-base-patch16-224 on Tiny-Imagenet targeting the penultimate layer.

A.6 VIT-SMALL-PATCH16-224

A.6.1 TINY-IMAGENET

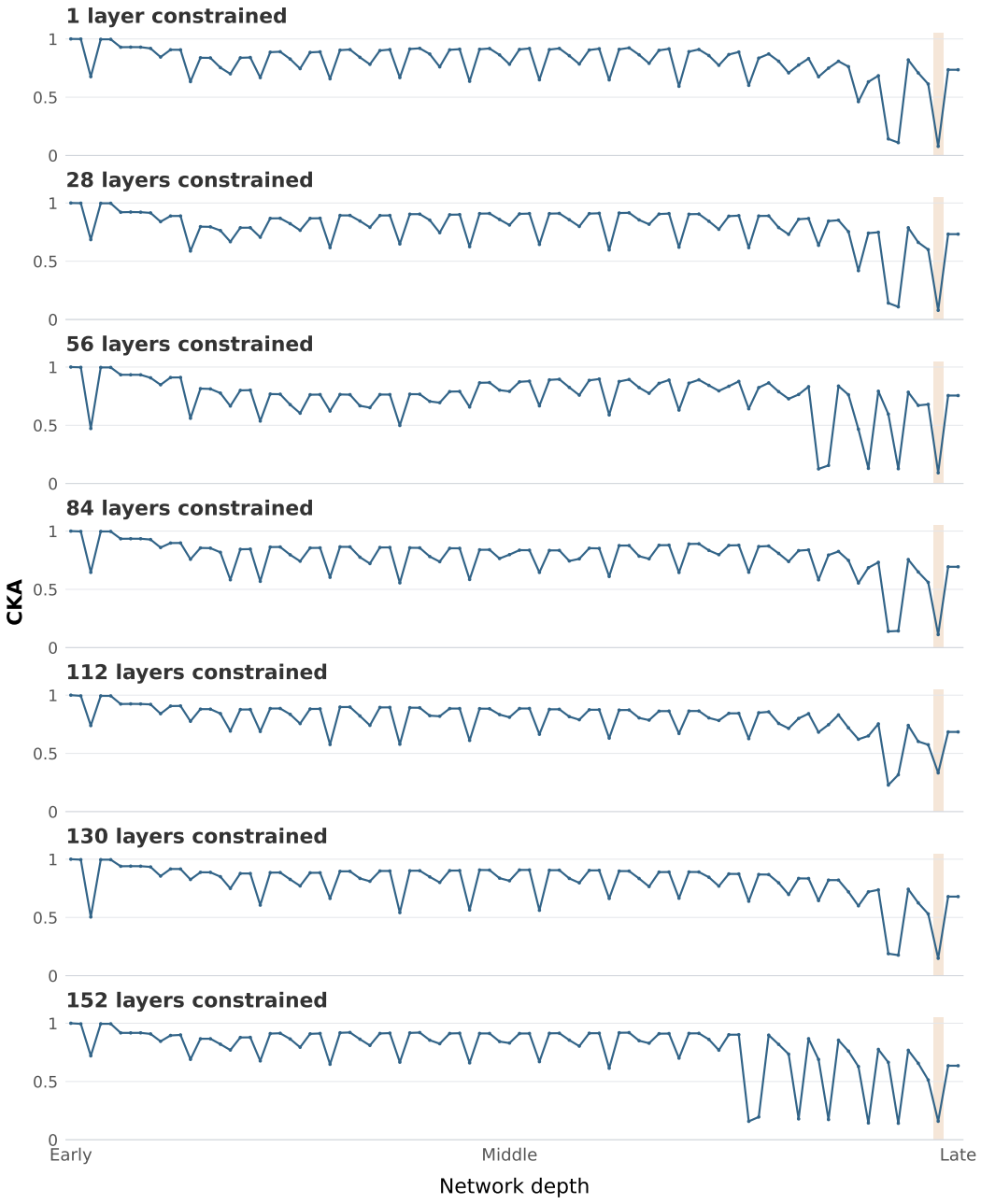


Figure 31: Results for increasing layer constraints in **min** direction for vit-small-patch16-224 on Tiny-Imagenet targeting the penultimate layer.

A.7 VIT-TINY-PATCH16-224

A.7.1 CIFAR-100

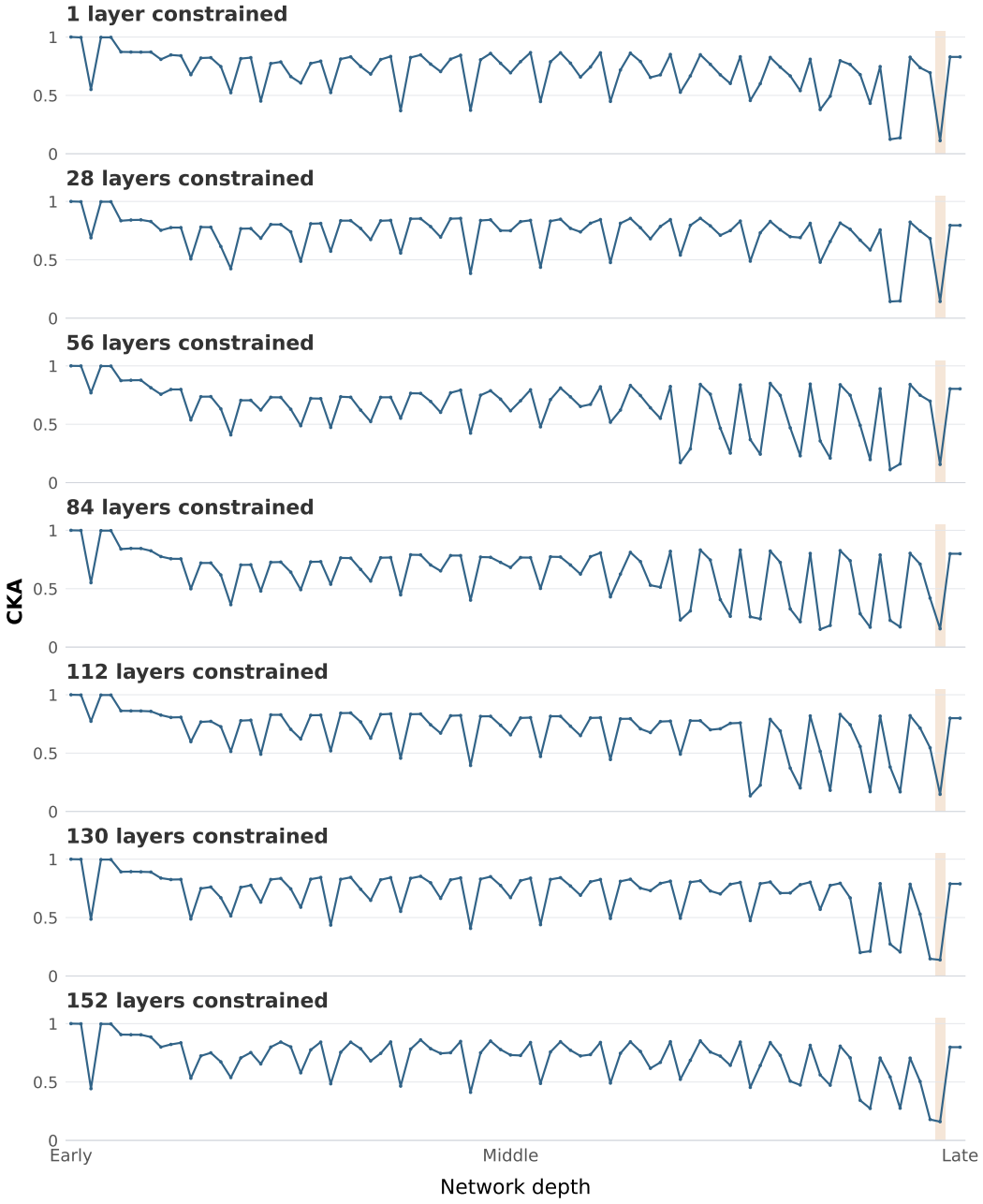


Figure 32: Results for increasing layer constraints in **min** direction for vit-tiny-patch16-224 on CIFAR-100 targeting the penultimate layer.

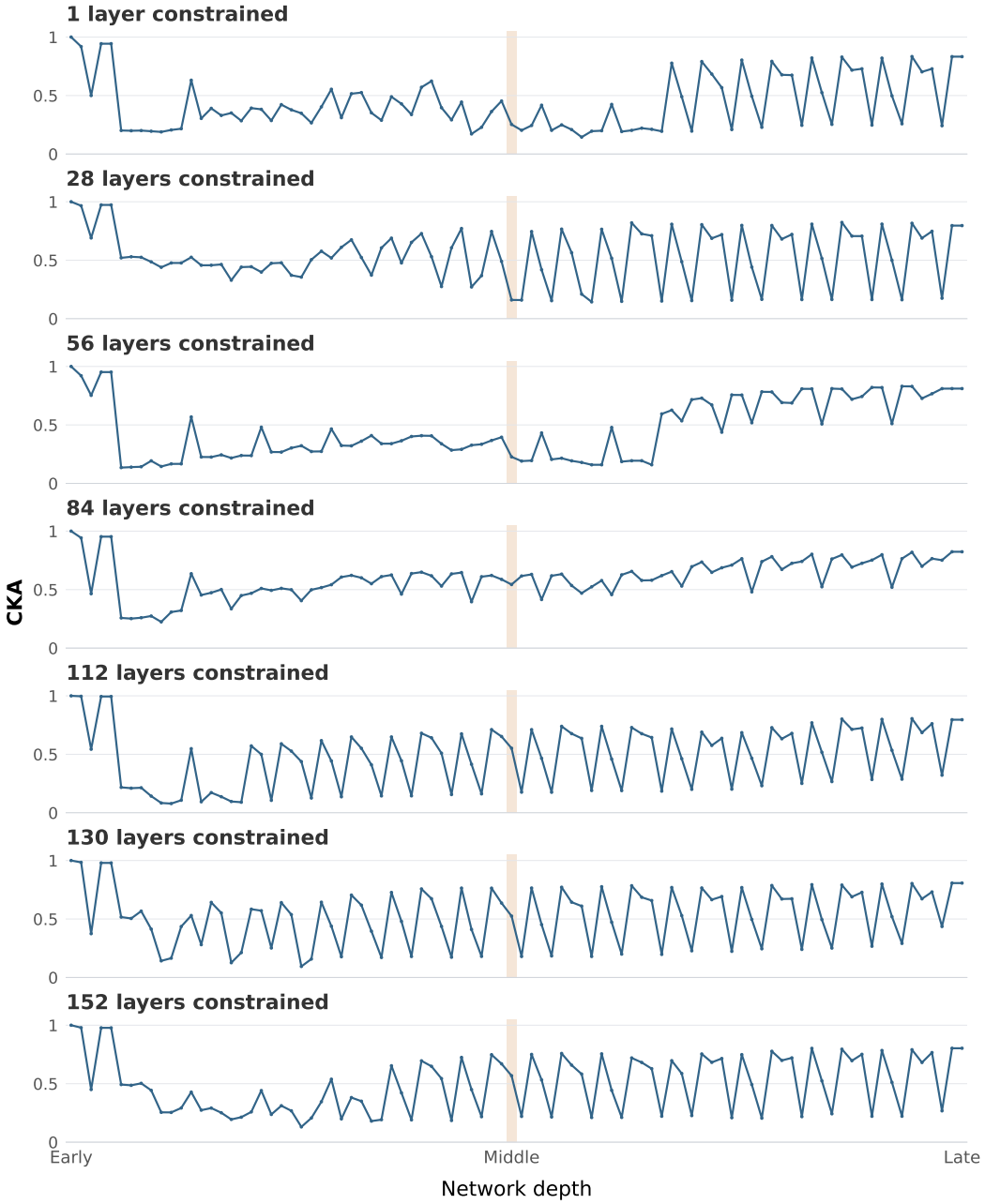


Figure 33: Results for increasing layer constraints in **min** direction for vit-tiny-patch16-224 on CIFAR-100 targeting the middle layer.