Towards an Optimal Control Foundation of Early Exits – Turnpikes and ResNets

Jens Püttschneider*
Institute of Control Systems
Hamburg University of Technology
Hamburg, Germany
jens.puettschneider@tuhh.de

Simon Heilig*
Faculty of Computer Science
Ruhr University Bochum
Bochum, Germany
simon.heilig@rub.de

Asja Fischer
Faculty of Computer Science
Ruhr University Bochum
Bochum, Germany

Timm Faulwasser
Institute of Control Systems
Hamburg University of Technology
Hamburg, Germany

Abstract

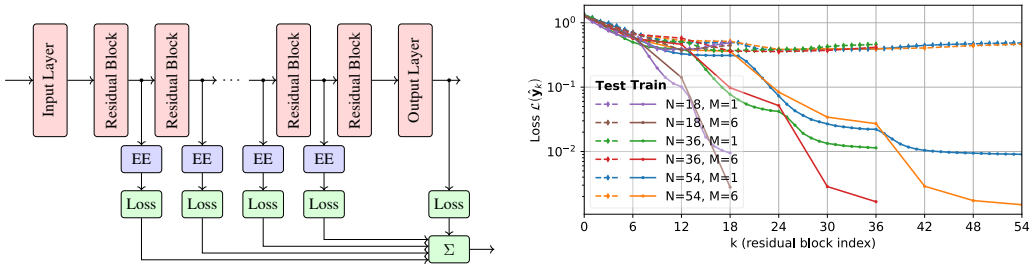
The feed-forward propagation through neural networks is crucial for training and inference. To trade off efficiency and accuracy, it has been proposed to consider intermediate outputs of hidden states as data-dependent regularization during training and, ultimately, as early-exits during inference. In this work, we analyze the joint optimization of loss functions computed for the output layer and intermediate layers using optimal control theory. We prove that ResNets trained in this joint fashion exhibit the so-called turnpike phenomenon, this implies that the number of layers required to reach a prescribed loss level is bounded from above independently of the overall network depth. This induces a similarity property across the internal representations of early-exiting networks of varying depths. We connect the turnpike to the overthinking phenomenon, and provide theoretical insights into the success of early-exit criteria derived from softmax probabilities, specifically entropy and max-probability. In a range of experiments, we demonstrate the existence and effect of turnpikes and the resulting similarity in the early-exit performance for ResNets of varying depths.

1 Introduction

Training *deep* neural networks is arguably the tipping point of today's success of deep learning models. Key techniques are residual connections and intermediate output heads informing the network's hidden layers about their current performance [Szegedy et al., 2015, He et al., 2016b]. What was initially used only as training scheme [Lee et al., 2015], quickly evolved as inference method to save computation on instances handled equally well by shallower parts of the network [Teerapittayanon et al., 2016] and has since been applied to early-exiting language models and vision transformers [Scardapane et al., 2020, Zhou et al., 2020, Schuster et al., 2022, Chen et al., 2023]. Without early-exiting, a key failure mode of deep neural networks is *overthinking* [Kaya et al., 2019]: samples correctly classified by earlier layers are propagated further, incurring unnecessary computation and potentially degrading predictions at later stages. At the core of training these early-exit networks lies an optimization problem considering a combined loss of all auxiliary and final output heads, as illustrated in Figure 1a. As we show in this work, it is precisely this joint training formulation that gives rise to the turnpike phenomenon from optimal control, offering a principled explanation

*Equal contribution.

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



(a) ResNet with early-exits and joint training.

(b) Loss trajectories.

Figure 1: Illustration of a ResNet with early-exits (EE), the joint training formulation induces the turnpike phenomenon from optimal control, which implies that early-exiting ResNets of different depths have similar trajectories of the early-exit losses along the network depth.

for why early-exit classifiers perform well and why overthinking can be avoided. With the turnpike phenomenon at hand, we are able to show that the early-exit depth required to reach a prescribed loss level is independent of the overall backbone depth. Consequently, the early-exit losses of networks of different depths are similar, as illustrated in Figure 1b.

Prior theoretical work has analyzed early-exit training either under restrictive assumptions such as strong convexity [Lee et al., 2015] or focused on layer-wise supervision regimes that lack the upstream gradient signal of joint optimization [Sakamoto and Sato, 2024]. Consequently, a fundamental understanding of how joint early-exit training behaves and why one can expect intermediate classifiers to perform well has remained elusive; a gap this work addresses through the lens of optimal control theory.

Motivated by residual connection-based neural networks (ResNets) and their connection to neural ODEs [He et al., 2016b,a, Chen et al., 2018], research started using tools from optimal control to investigate these deep learning models, where trainable parameters act as control inputs that steer hidden states towards a target representation determined by the loss function [Esteve et al., 2020, Esteve-Yagüe and Geshkovski, 2023, Ruiz-Balet and Zuazua, 2023]. A central technique is to use a *stage cost* over the full forward trajectory accumulating (loss-) penalties along the trajectory the data takes through the network. This can be understood as shaping the transient loss behavior, which enables stronger convergence guarantees and explicit depth bounds [Esteve et al., 2020, Faulwasser et al., 2024]. A commonality of these works is the turnpike phenomenon: hidden representations tend to cluster around the loss minimizer for the majority of layers, independent of network depth. However, while existing analyses focus on continuous space-time networks [Esteve et al., 2020], the application of this framework to standard ResNets architectures remains an open problem. In addition, our novel optimal control perspective on early-exiting enables new avenues towards formal analysis of training formulations.

In this work, we exploit the close relation between stage cost terms in discrete-time optimal control problems and the joint optimization formulation of early-exit neural networks. This allows us to show under which conditions this approach leads to the turnpike phenomenon. In turn this helps to uncover the fundamental mechanisms of joint training of early-exit neural networks. In short, our paper makes the following contributions:

- A turnpike-based foundation of early-exit-based ResNet training and inference.
- Novel asymptotic upper bounds on the cross-entropy loss during forward propagation revealing the loss turnpike phenomenon.
- Theoretical bounds on guaranteeing correct classification and bounded early-exiting depths, derived for entropy and maximum softmax probability criteria.
- Empirical support of the turnpike analysis and a principled approach to obtain the intermediate outputs in standard ResNets.

2 Preliminaries

We begin with a review of the concepts of early-exiting neural networks and optimal control theory.

2.1 Early-Exiting ResNets

Early-exiting, as proposed by Teerapittayanon et al. [2016], augments a neural network with intermediate classifiers that provide predictions before the final layer. The inference process is interrupted as soon as an intermediate prediction is deemed sufficiently confident, thereby reducing computational cost and preventing the model from overthinking [Kaya et al., 2019].

In this work, we consider standard ResNets [He et al., 2016a, Dosovitskiy et al., 2021] with N residual blocks (RBs) which for a given input sample $x^{(i)}$ produce the output $\hat{y}^{(i)}$ by

$$x^{(i)} = \mathcal{V}\left(x^{(i)}, \mathbf{w}_{\mathcal{V}}\right), \quad x_{k+1}^{(i)} = \mathcal{S}_k\left(x_k^{(i)}, \mathbf{w}_{\mathcal{S},k}\right) + \mathcal{F}_k\left(x_k^{(i)}, \mathbf{w}_{\mathcal{F},k}\right), \quad \hat{y}^{(i)} = \mathcal{O}(x_N^{(i)}, \mathbf{w}_{\mathcal{O}}). \tag{1}$$

This architecture includes the input block $\mathcal{V}$ with parameters $\mathbf{w}_{\mathcal{V}}$, N residual blocks with skip connection $\mathcal{S}_k$ and residual term $\mathcal{F}_k$ with parameters $\mathbf{w}_{\mathcal{S},k}$ and $\mathbf{w}_{\mathcal{F},k}$ (summarized as $\mathbf{w}_k = [\mathbf{w}_{\mathcal{S},k}, \mathbf{w}_{\mathcal{F},k}]$), as well as the output head $\mathcal{O}$ with parameters $\mathbf{w}_{\mathcal{O}}$. We collect all parameters of the ResNet into $\mathbf{w} = [\mathbf{w}_{\mathcal{V}}, \mathbf{w}_0, \ldots, \mathbf{w}_{N-1}, \mathbf{w}_{\mathcal{O}}]$. Our subsequent analysis applies to both networks with varying state dimensions, such as convolutional ResNets [He et al., 2016a], or ResNets with a constant state dimension. For details on the ResNet implementation and our control-inspired notation, we defer to Appendix B.

In the classical case, the output head produces the *final prediction* $\hat{y}^{(i)}$, while early-exit neural networks additionally have intermediate output heads $\mathcal{O}_k$ with parameters $\mathbf{w}_{\mathcal{O},k}$ attached to the k-th hidden layer (where $k \in \mathcal{K}_{\mathcal{O}} \subseteq \mathbb{N}_{[0,N-1]}$) to obtain *intermediate predictions*: $\hat{y}_k^{(i)} = \mathcal{O}_k\left(x_k^{(i)}, \mathbf{w}_{\mathcal{O},k}\right)$. During inference, common decision rules are based on the indication of high confidence such as by low entropy $\mathcal{H}(\hat{y}_k^{(i)})$ or high maximum softmax probability $\hat{p}_{\max}(\hat{y}_k^{(i)})$ [Huang et al., 2017, Passalis et al., 2020, Liu et al., 2022b].

To train the backbone network and the early-exits, different strategies have been proposed [Rahmath P et al., 2024, Kubaty et al., 2025b, Regol et al., 2024]. In this work, we set out to study the joint training formulation through its intrinsic connection to optimal control theory, considering a supervised classification problem with labeled training data $\mathbb{D} = \left\{\left(x^{(1)}, y^{(1)}\right), ..., \left(x^{(D)}, y^{(D)}\right)\right\}$, where $\mathbf{x} = \left[x^{(1)}, x^{(2)}, \ldots, x^{(D)}\right]$ denotes the stacked features (boldface symbols used for their propagation through the network), $\hat{\mathbf{y}}$ the network output over the entire dataset, and $\mathcal{L}(\hat{\mathbf{y}})$ the corresponding average cross-entropy loss[2]. The main training approaches are either sequential—first optimizing the backbone and then keeping it frozen while training the early-exits—or joint, where backbone and early-exits are trained simultaneously. This is achieved by minimizing the *joint training objective*, a weighted sum of intermediate and final losses,

$$\sum_{k \in \mathcal{K}_{\mathcal{O}}} \gamma \mathcal{L}(\hat{\mathbf{y}}_k) + \mathcal{L}(\hat{\mathbf{y}}),$$

where the scalar hyperparameter $\gamma \in (0, 1]$ controls the trade-off between the intermediate and final losses. Larger values of γ emphasize the intermediate losses, while smaller values prioritize the final loss.

2.2 The Turnpike Phenomenon in Optimal Control

Control theory aims to determine input signals or feedback policies for dynamic systems to induce a desired trajectory behavior. Optimal control, in particular, determines these control inputs by solving an optimization problem.

Consider the dynamic system $x_{k+1} = f(x_k, w_k)$, with the initial condition $x_0 = x^0$. The evolution of the state $x_k \in \mathbb{R}^n$ over the discrete time index k is steered by the control inputs $w_k \in \mathbb{R}^m$. The goal of control is to determine the inputs w_k as to achieve a desired behavior of the state trajectories

[2]For simplicity of exposition, we omit the ground-truth label as an additional argument of the loss function.

x_k. This is achieved by optimizing over the possible trajectories of this dynamic system. A simple optimal control problem (OCP) with finite horizon $N \in \mathbb{N}$ reads

$$\min_{w_0,\dots,w_{N-1}} \sum_{k=0}^{N-1} \ell(x_k, w_k) + J_{\mathrm{f}}(x_N) \quad \text{s.t. } x_{k+1} = f(x_k, w_k),\ x_0 = x^0,\ \forall k = 0, \dots, N-1. \tag{2}$$

The objective entails the stage cost $\ell : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, which penalizes the states and control inputs along the trajectory and the terminal cost $J_{\mathrm{f}} : \mathbb{R}^n \to \mathbb{R}$ which considers the terminal state x_N. The structure of the OCP depends on the system dynamics f and the choice of the objective. It can range from linear systems with quadratic costs to nonlinear systems with non-convex costs (or it may include inequality constraints which we neglect here). Such problems can be solved using various methods, ranging from direct solution using numerical optimization to applying dynamic programming/reinforcement learning to compute/approximate optimal value function [Bertsekas, 2005]. Figure 2 shows solutions of the simple OCP (2) with $x_{k+1} = 2x_k + w_k$, $x^0 = 0$, $\ell(x_k, w_k) = (x_k - 2)^2 + w_k^2$, and $J_{\mathrm{f}}(x_N) = 0$ for varying horizons $N = 5, 10, \dots, 40$. In the middle of the horizon, all trajectories cluster around $x_s^\star = 1$ and $w_s^\star = -1$, while they depart towards the end of the horizon. One may observe that $x_s^\star = 1$ and $w_s^\star = -1$ is the optimal steady state pair of the OCP (2), i.e., the steady state pair that minimizes the stage cost $(x_s^\star, w_s^\star) = \operatorname{argmin}_{(x_s, w_s)} \ell(x_s, w_s)$, s.t. $x_s = f(x_s, w_s)$. However, near the end of the finite horizon, the objective is minimized by departing from the optimal equilibrium, a phenomenon known as a *leaving arc*.

This characteristic clustering of optimal trajectories close to the optimal steady state is known as the *turnpike property*, early observations of which can be traced back to von Neumann [1938], predating the fundamental works of Bellman [1954] and Boltyanskii et al. [1956]. The term *turnpike* has been coined in the book by Dorfman et al. [1958] and it has received widespread attention in control, cf. McKenzie [1976], Trélat and Zuazua [2015]. Indeed, the turnpike property is a similarity property of the OCP solutions, i.e., the input and state trajectories remain close to the optimal steady state for most of the time independent of the initial conditions x_0 and horizon lengths N. A formal definition of the phenomenon is as follows.

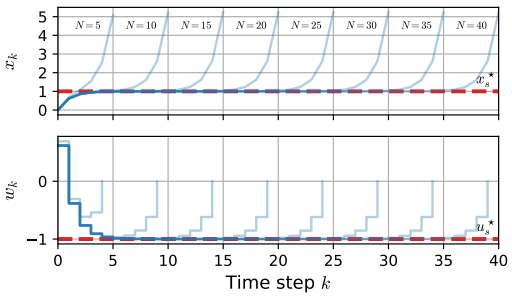


Figure 2: Example of optimal trajectories of the OCP (2) for varying horizons $N = 5, 10, \dots, 40$. The red dashed lines indicate the turnpike $(x_s^\star, w_s^\star) = (1, -1)$.

Definition 2.1 (Turnpike phenomenon)**.** The OCP (2) is said to exhibit the turnpike phenomenon with respect to $(x_s^\star, w_s^\star) = const$ if for every threshold $\varepsilon > 0$, there exists a constant $\kappa < \infty$ such that the number of timesteps for which the optimal states and input lie outside of an ε-neighborhood of $(x_s^\star, w_s^\star)$ satisfies the uniform bound $\#\{k = 0, ..., N \mid \|(x_k, w_k) - (x_s^\star, w_s^\star)\| > \varepsilon\} \leq \kappa$ independent of the horizon N and the initial condition x^0.

Indeed, there exist different approaches on how turnpikes can be analyzed, we refer to Trélat and Zuazua [2015] and Faulwasser and Grüne [2022] for recent overviews.

3 Optimal Control Theory for Early-Exiting

We now formalize the joint training formulations for early-exiting ResNets from an optimal control perspective. We begin by considering ResNets as dynamic systems and view early exit training as optimal control problem in Section 3.1. Our main contributions are twofold: in Section 3.2, we introduce a novel turnpike notion, and in Section 3.3, we extend this analysis to early-exit criteria.

3.1 Training Early-Exit ResNets as Optimal Control

ResNets with constant state dimension can be regarded as a *discrete-time dynamic system* $\mathbf{x}_{k+1} = f_k(\mathbf{x}_k, \mathbf{w}_k) = \mathbf{x}_k + \mathcal{F}_k(\mathbf{x}_k, \mathbf{w}_k), \quad \mathbf{x}_0 = \mathbf{x}$, on the finite horizon $k = 0, ..., N-1$ corresponding to the number of residual blocks. The hidden state $\mathbf{x}_k$ corresponds to the *state of the system* while the

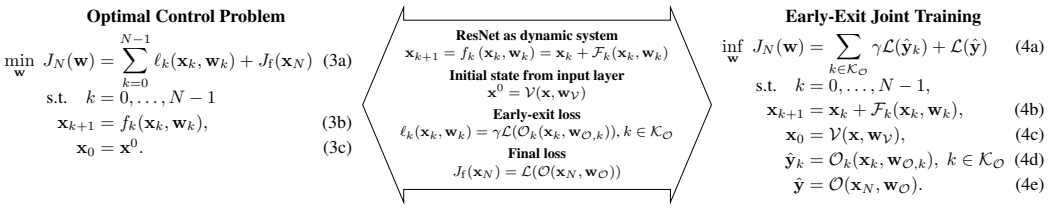


Figure 3: The fundamental connection between early-exiting joint training formulation and training as an optimal control problem.

trainable parameters $\mathbf{w}_k$ act as *control inputs*, see also Li et al. [2017] and Ruiz-Balet and Zuazua [2023]. An important property of ResNets is that zero parameters reduce a residual block to an identity mapping via the skip connection $\mathbf{x}_{k+1} = f_k(\mathbf{x}_k, 0) = \mathbf{x}_k + \mathcal{F}_k(\mathbf{x}_k, 0) = \mathbf{x}_k$, implying that under convergence of the training, the performance does not degrade with increasing depth, since additional residual blocks can always be bypassed He et al. [2016a]. From the optimal control perspective, this implies that each state can be rendered a steady state by $\mathbf{w}_k = 0 = const.$

Next, we phrase the early-exit joint training formulation for ResNets as an OCP. We begin with the joint training formulation for an early-exiting neural network based on ResNet [He et al., 2016a]. Our insights are summarized in Figure 3. We use a constant weight $0 < \gamma \leq 1$ to trade off the loss function of the intermediate outputs and the final output.

In contrast to classical OCPs, where the state dimension of the dynamical system is fixed, in standard ResNets the hidden-state dimension varies with depth [He et al., 2016a]. In principle, these ResNets can still strictly be considered as a dynamic system with constant state dimension by embedding the entire network dynamic into a fixed state space corresponding to the maximum dimension in the network. Here, however, we allow the state dimension to vary and instead focus our analysis on the outputs, which have a fixed dimension and are used to obtain predictions and evaluate the loss function. In classification tasks, this dimension typically equals the number of classes over which softmax probabilities are computed.

It remains to cast the early-exit joint training problem (4) as an OCP in the form of (3), cf. Figure 3. To this end, we define the stage cost as the loss function for the intermediate outputs, i.e., $\ell_k(\mathbf{x}_k, \mathbf{w}_k) = \gamma \mathcal{L}(\mathcal{O}_k(\mathbf{x}_k, \mathbf{w}_{\mathcal{O},k}))$ for the RBs with early-exits ($k \in \mathcal{K}_\mathcal{O}$), where with slight abuse of notation, $\mathbf{w}_k$ contains both the residual block parameters and the intermediate output parameters, and $\ell_k(\mathbf{x}_k, \mathbf{w}_k) = 0$ for the RBs without ($k \notin \mathcal{K}_\mathcal{O}$). The loss function of the final output is used as terminal penalty, i.e., $J_\mathrm{f}(\mathbf{x}_N) = \mathcal{L}(\mathcal{O}(\mathbf{x}_N, \mathbf{w}_\mathcal{O}))$. In contrast to the fixed initial state in the OCP, the initial state for the first residual block comes from the trainable input layer, $\mathbf{x}_0 = \mathcal{V}(\mathbf{x}, \mathbf{w}_\mathcal{V})$. With this, we obtain a direct equivalence between the joint training formulation for early-exiting ResNets (4) and the OCP (3) as is illustrated in Figure 3.

3.2 Turnpikes in Early-Exiting ResNets

With the optimal control formulation of the joint training for early-exiting ResNets at hand, we analyze the implications of this perspective. Since the joint training (4) with the cross-entropy loss might not attain its minimum, we analyze δ suboptimal solution with $J_N(\mathbf{w}) \leq \inf_{\mathbf{w}'} J_N(\mathbf{w}') + \delta$, for a fixed $0 \leq \delta < \infty$. Crucially, our analytic findings derived this way are supported by the numerical experiments presented in Section 4. We begin by making an assumption on the loss decay of intermediate outputs as the network depth tends to infinity.

Assumption 3.1 (Exponential loss decay)**.** For the given training dataset $\mathbb{D}$, there exist constants $\rho \in [0, 1)$ and $\beta > 0$, and an infinite-depth ResNet with early-exits attached to each residual block, with parameters $\tilde{\mathbf{w}}_\mathcal{V}$ for the input block, $\tilde{\mathbf{w}}_k$ for the infinite sequence of residual blocks $k \in \mathbb{N}_{[0,\infty)}$, and $\tilde{\mathbf{w}}_{\mathcal{O},k}$ for the corresponding intermediate outputs, such that the training loss of the intermediate outputs $\tilde{\mathbf{y}}_k$ satisfies $\mathcal{L}(\tilde{\mathbf{y}}_k) \leq \beta \rho^k$, for $k \in \mathbb{N}_{[0,\infty)}$.

Intuitively, this assumption reflects the idea that, with increasing depth the achievable loss improves. In addition, the constants β and ρ capture the representational capacity of the network relative to the complexity of the training data: for a more challenging learning problem or a network with lower representational capacity per residual block, they become larger, reflecting that a deeper network is

required to reach a given loss level. For loss functions with finite minimizers, such as the ℓ_2 loss, this decay assumption can be verified using well-known memorization results [Hardt and Ma, 2017, Nguyen and Hein, 2018, Kim et al., 2023]. However, for the cross-entropy loss this decay requires the logits to diverge which we show in Theorem E.5 for fully connected ResNets with ReLU activations by extending memorization results of Hardt and Ma [2017]. Hence this assumption is not overly restrictive. Consequently, in the following loss turnpike results, in contrast to classical turnpike results in optimal control, the outputs do not become stationary; rather, they diverge to further reduce the cross-entropy loss. Under this assumption we obtain the following result.

Theorem 3.2 (Loss turnpike in ResNet training)**.** *Suppose that Assumption 3.1 holds. Let a ResNet (1) with depth N be trained with stage cost loss weighted by $0 < \gamma \leq 1$ using the optimal control training formulation (4) with the early-exit placement $\mathcal{K}_\mathcal{O}$, and let its parameters $\mathbf{w}$ be δ-suboptimal for some fixed $0 \leq \delta < \infty$, i.e., $J_N(\mathbf{w}) \leq \inf_{\mathbf{w}'} J_N(\mathbf{w}') + \delta$, then the training objective is bounded independent of the depth N and the early-exit placement $\mathcal{K}_\mathcal{O}$ by*

$$J_N(\mathbf{w}) \leq \bar{J} = \frac{\gamma\beta}{1-\rho} + \beta + \delta,$$

and the ResNet exhibits a loss turnpike on the training data. More precisely, for a loss level $\varepsilon > 0$, consider the set of intermediate outputs with training losses exceeding ε, $\hat{\mathcal{Q}}_\varepsilon \doteq \{k \in \mathcal{K}_\mathcal{O} \cup \{N\} \,|\, \mathcal{L}(\hat{\mathbf{y}}_k) > \varepsilon\}$. Then, for every loss level $\varepsilon > 0$ there exists a constant $\kappa(\varepsilon) < \infty$, independent of the network depth N and the early-exit placement $\mathcal{K}_\mathcal{O}$, such that the cardinality of $\hat{\mathcal{Q}}_\varepsilon$ is bounded $\#\hat{\mathcal{Q}}_\varepsilon \leq \kappa(\varepsilon) = \frac{\bar{J}}{\gamma\varepsilon}$.

Proof sketch. By Assumption 3.1, there exists an infinite-depth network with early-exits at every residual block whose training losses are bounded by the convergent sequence $\beta\rho^k$. For any early-exit placement $\mathcal{K}_\mathcal{O} \subseteq \mathbb{N}_{[0,N-1]}$, taking the first N blocks and the early-exits in $\mathcal{K}_\mathcal{O}$ gives a feasible solution to the training OCP, whose objective, by $\mathcal{K}_\mathcal{O} \subseteq \mathbb{N}_{[0,\infty)}$ and non-negativity of the loss, is bounded by the corresponding geometric series. Hence, the limit of corresponding geometric series yields a constant upper bound $\bar{J}$ on the joint training objective, independent of the network depth $J_N(\mathbf{w}) \leq \bar{J}$. By construction, each intermediate output $k \in \hat{\mathcal{Q}}_\varepsilon$ contributes at least $\gamma\varepsilon$ to the joint training objective. Since the total objective is bounded by $\bar{J} = \frac{\gamma\beta}{1-\rho} + \beta + \delta$, it follows that the number of such intermediate outputs is finite, i.e., $\#\hat{\mathcal{Q}}_\varepsilon \leq \kappa(\varepsilon) \doteq \frac{\bar{J}}{\gamma\varepsilon}$. For the full proof we refer to Appendix E. □

This loss turnpike ensures that, for any prescribed loss level, only a limited forward propagation steps are needed in a jointly trained early-exit ResNet. Consequently, each sample is correctly classified after a finite depth implying that the turnpike not only occurs in the loss but also in the accuracy, which we formalize in Theorem E.7. This result provides a theoretical explanation for the overthinking phenomenon in deep networks by showing that the depth at which a training sample enters the stable classification regime is bounded independently of the total network depth [Kaya et al., 2019].

3.3 Early-Exiting the Turnpike

We now investigate the implications for early-exiting inference, where inference is terminated once an uncertainty measure falls below a threshold [Huang et al., 2017, Passalis et al., 2020, Liu et al., 2022b]. At a block $k \in \mathcal{K}_\mathcal{O}$, an early-exit is taken if the uncertainty measure satisfies $\phi(\hat{y}_k^{(i)}) \leq T$. Commonly used uncertainty measures include the entropy $\phi(\hat{y}_k^{(i)}) = \mathcal{H}(\hat{y}_k^{(i)})$ or the maximum softmax probability complement $\phi(\hat{y}_k^{(i)}) = 1 - \hat{p}_{\max}(\hat{y}_k^{(i)})$. Since the loss turnpike bounds the depth required to reach any loss level $\varepsilon > 0$, the softmax probabilities lie in an ε-neighborhood of the one-hot label, and hence any derived uncertainty measure is similarly bounded. Based on this, we show that a jointly trained ResNet exhibiting a loss turnpike also exhibits turnpike behavior in the entropy (Theorem E.10) and maximum softmax probability complement (Theorem E.12) on the training data. Consequently, using these quantities as exit criteria yields that for any exiting threshold $T > 0$, the corresponding exit depth on the training data is bounded independent of the network depth. In addition, we show that there exists an uncertainty threshold $T > 0$ that guarantees correct classification of early-exited samples. Together, these results guarantee that both uncertainty criteria can be used to correctly exit the turnpike, avoiding overthinking on the training data without sacrificing accuracy as formulated in the following result.

Theorem 3.3 (Bounded and correct early-exiting with turnpikes)**.** *Let Assumption 3.1 hold and consider a ResNet of depth N and early-exits attached residual blocks $\mathcal{K}_\mathcal{O}$ trained by (4) with stage cost weight $0 < \gamma \leq 1$, and let its parameters $\mathbf{w}$ be δ-suboptimal for some fixed $0 \leq \delta < \infty$, i.e., $J_N(\mathbf{w}) \leq \inf_{\mathbf{w}'} J_N(\mathbf{w}') + \delta$. Then, there exists an early-exiting threshold $T > 0$ and an exit depth κ for the early-exit criteria entropy and max softmax probability complement, such that every training sample $i = 1, ..., D$ is exited at latest at the $(\kappa+1)$-th exit $\# \left\{ k \in \mathcal{K}_\mathcal{O} \cup \{N\} \,|\phi(\hat{y}_k^{(i)}) > T \right\} \leq \kappa$, independent of the depth N and the early-exit placement. Consequently, the early-exit depth in terms of the number of residual blocks is bounded by*

$$k^{(i)} = \min\left(\left\{k \in \mathcal{K}_\mathcal{O} | \phi(\hat{y}_k^{(i)}) \leq T\right\} \cup \{N\}\right) \leq \operatorname{seq}\left(\mathcal{K}_\mathcal{O} \cup \{N\}\right)_{\kappa+1}.$$

Additionally, if a sample exits before the final output, i.e., $\phi(\hat{y}_k^{(i)}) \leq T$ for $k \in \mathcal{K}_\mathcal{O}$, it is classified correctly $\hat{c}(\hat{y}_k^{(i)}) = y^{(i)}$.

We prove this result separately for each criterion in Theorem E.15 (entropy) and Theorem E.16 (max softmax probability complement), respectively. Note that if these early-exits are placed with fixed spacing $M \geq 1$, i.e., $\mathcal{K}_\mathcal{O} = \{0, M, 2M, \ldots, N - M\}$, then the early-exiting depth of each sample $i = 1, \ldots, D$ is bounded by $k^{(i)} = \min\left(\left\{k \in \mathcal{K}_\mathcal{O} | \phi(\hat{y}_k^{(i)}) \leq T\right\} \cup \{N\}\right) \leq \kappa \cdot M$, independent of the depth N. For more discussion on the asymptotic exit placement, see Appendix D.

4 Experiments

To underscore our turnpike analysis for early-exit networks, we conduct experiments[3] on standard ResNet architectures of varying depths, following the deeply supervised net of Lee et al. [2015] supervising every hidden state and the partially supervised approach of Kaya et al. [2019]. We detail the early-exit architectures used in Section 4.1. Then, in Section 4.2 we examine whether the turnpike similarity predicted by Theorem 3.2 and Theorem E.7 is observable in practice for ResNets of different depth trained via gradient-based optimization. In Section 4.3 we analyze how the turnpike behavior empirically relates to overthinking [Kaya et al., 2019] and how it translates to the inference time and demonstrate that the turnpike similarity from Theorem 3.3 in the early-exit inference. Finally, Section 4.4 evaluates the influence of the stage cost loss weight and its influence of the turnpike bound from Theorem 3.2.

4.1 Early-Exit Architecture

Instead of having independent exit paths, we design the intermediate outputs $\mathcal{O}_k$ for the hidden state $x_k^{(i)}$ by following the computational graph through the following skip connections and finally the output layer of the neural networks

$$\hat{y}_k^{(i)} = \mathcal{O}_k(x_k^{(i)}, \mathbf{w}_{\mathcal{O},k}) = \mathcal{O}(\mathcal{S}_{N-1}(\cdots \mathcal{S}_k(x_k^{(i)}, \mathbf{w}_{\mathcal{S},k}), \cdots, \mathbf{w}_{\mathcal{S},N-1}), \mathbf{w}_\mathcal{O}).$$

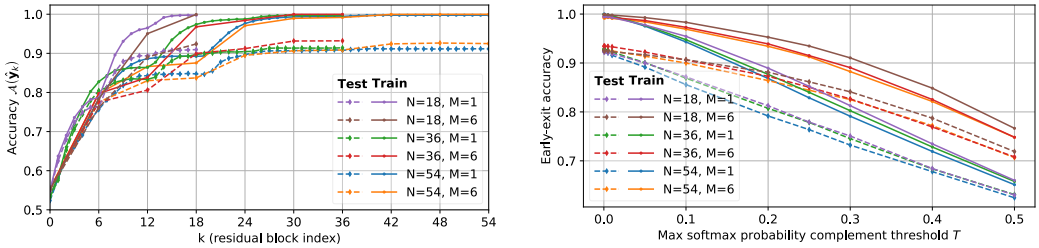
This architecture of exits ensures that if from some layer $k < N$ onward the parameters of all residual blocks are set to zero, the intermediate output at layer k and all subsequent intermediate outputs coincide with the final network output. Leveraging this property, we prove in Theorem E.2 that the training loss of the final output exceeds the loss of any intermediate output by at most the suboptimality δ. Therefore, the leaving arc of the turnpike is bounded by the suboptimality of the training and vanishes for optimal training.

Since most skip connections are identity mappings and only at stage transitions a 1×1 convolution is employed [He et al., 2016a], this design introduces only marginal overhead. Additionally, this design naturally enables parameter sharing between the backbone and the early-exits, which we examine experimentally in Appendix A.4.

4.2 Can we Observe the Turnpike?

To examine the turnpike phenomenon for varying depths, we conduct experiments on CIFAR-10, CIFAR-100, and TinyImagenet (Appendix A.2) [Krizhevsky, 2009, Deng et al., 2009] using ResNet-110 (N=54) and two shallower variants with $N = 18$ and $N = 36$ residual blocks as backbone

[3] Code to reproduce our results is available here https://github.com/OptCon/early-exit-turnpike.



(a) Accuracy trajectories.

(b) Early-exit accuracy over exiting thresholds.

Figure 4: Accuracies on CIFAR-10 for ResNets with residual blocks $N \in \{18, 36, 54\}$ with two levels of early-exit supervision: every residual block ($M = 1$) and every sixth ($M = 6$).

architectures. In addition, we vary the early-exit placement by attaching exits either to every residual block ($M = 1$) or to every $M = 6$ residual blocks, following Lee et al. [2015] and Kaya et al. [2019]. The experimental setup is further outlined in Appendix A.1 and we plan to release experimental code publicly for the camera-ready version.

In Figure 1, we show the loss trajectories obtained at the respective early-exits of the trained networks and averaged over all samples. The training losses of all networks decrease very sharply with depth and converge to near-zero values, indicating that the networks enter the interpolation regime and thereby verifying the exponential loss decrease from Assumption 3.1. Moreover, the training loss improves with the backbone depth and over the depth of the early-exits for all networks. The qualitatively similar training-loss trajectories observed for ResNets of different depths and early-exit placements are consistent with the loss turnpike behavior predicted by Theorem 3.2. The monotonic decrease of the loss implies that the corresponding trajectories do not exhibit a leaving arc, which is expected for this choice of intermediate outputs and a near optimal training by Theorem E.2. In addition, due to the different stages, the rate at which the loss decays increase for the deeper stages, which possess more trainable parameters per residual block, leading to kinks in the plot. For a detailed discussion of the influence of the architecture, we refer to Appendix A.2.

Figure 4a shows the corresponding accuracy trajectories. Within numerical precision, all models reach almost $100\,\%$ accuracy at their terminal output, confirming a perfect fit to the CIFAR-10 training data. Across all backbone depths and early-exit placements, this is preceded by a steep initial rise in the first early-exits, with the accuracy turnpike reached within at most 30 residual blocks, consistent with the accuracy turnpike of Theorem E.7. Reaching this turnpike enables correct early-exit predictions; however, remaining on the turnpike without exiting constitutes wasteful overthinking.

4.3 Turnpikes during Test-Time and Early Exiting?

To answer this questions, we take the test splits of CIFAR-10, CIFAR-100, and TinyImageNet (in Appendix A.2) as in-distribution samples. Since our theoretical guarantees only hold for the training data, we investigate empirically how they transfer to the test data. The test accuracies in Figure 4a show a similar convergence behavior as the training accuracies, albeit with the well-reported generalization gap of the ResNets of about $7\,\%$ points on CIFAR-10 [He et al., 2016a]. In line with the empirical findings of Huang et al. [2017], providing the model with fewer (every $M = 6$) early-exits leads to slight improvement of the test accuracy.

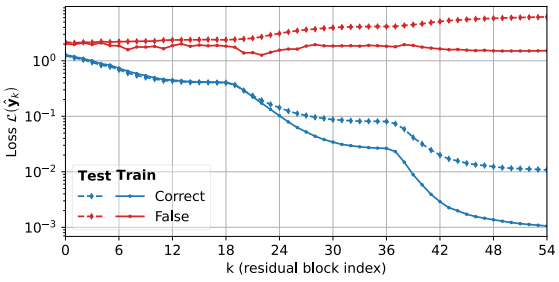


Figure 5: Training and test loss trajectories across network depth, conditioned on the final prediction outcome on CIFAR-10 with ResNet $N = 54$ and early exits placed at every block ($M = 1$).

While the test accuracy trajectories behave similarly to their training counterparts, the test loss shows a different pattern: it initially decreases but then increases again, and none of the networks attain their minimum test loss at the final layer. As shown in Figure 5, falsely classified test samples become increasingly overconfident in their incorrect predictions, incurring much larger loss than correctly classified samples, which saturate near zero. Hence, test samples exhibit similar turnpike behavior as their training counterparts, but for misclassified samples the turnpike drives predictions toward the wrong class with growing confidence. This overconfidence of misclassified samples may help explain why destructive overthinking occurs primarily at the first exits [Kaya et al., 2019].

Building on the turnpike analysis, we now investigate its implications for early-exit inference, using the max softmax probability complement as the uncertainty-based exiting criterion. Figure 4b shows the early-exiting accuracy over the thresholds for varying ResNet depths and early-exit placements. As the threshold increases, and exiting is allowed at lower confidences, the accuracies decrease for all networks, strikingly, the accuracies are nearly identical across the network depths N, reflecting the turnpike similarity property in the exiting behavior from Theorem 3.3. In addition, the decrease is more pronounced for exits at every block $M = 1$ than for $M = 6$, since the greater number of exit opportunities for $M = 1$ favors shallower exits. We further analyze the turnpike behavior on test data and during early-exit inference, considering the exit depth and entropy as the exit criterion, in Appendix A.3.

4.4 Effect of Stage-Cost Weight on the Turnpike Behavior?

The stage-cost loss weight $\gamma > 0$ is an additional hyperparameter in the joint training objective relative to standard training. It controls the trade-off between the final loss and the early-exit losses: larger values of γ place greater emphasis on intermediate outputs, whereas smaller values focus training on the final output, yielding better final-output performance. This relationship is also reflected in the turnpike bound from Theorem 3.2. In particular, the number of non-turnpike residual blocks for which the cross-entropy loss exceeds a loss threshold $\varepsilon > 0$ is bounded by $\#\hat{\mathcal{Q}}_\varepsilon \leq \frac{1}{\gamma\varepsilon}\left[\frac{\gamma\beta}{1-\rho} + \beta + \delta\right] = \frac{1}{\varepsilon}\left[\frac{A}{\gamma} + B\right]$, where A and B absorb the model and training-dependent constants β, ρ and δ. Hence, when the stage cost loss weight becomes small ($\gamma \to 0$) the number of intermediate outputs with higher loss values increases.

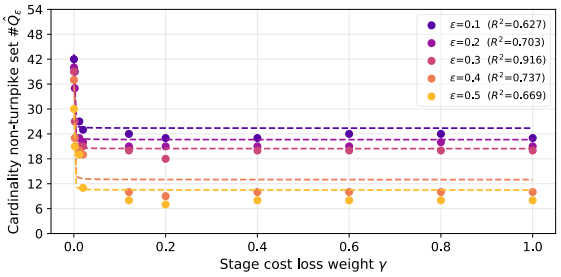


Figure 6: Cardinality of the turnpike set $\#\hat{\mathcal{Q}}_\varepsilon$ and fitted curves $A/\gamma+B$ in dashed over the stage cost loss weight γ for fixed loss thresholds ε. Results are for CIFAR-10, ResNet with $N = 54$ residual blocks and exits at every block ($M = 1$).

To investigate the effect of γ and empirically evaluate the number of non-turnpike layers bounded by the turnpike, Figure 6 shows $\#\hat{\mathcal{Q}}_\varepsilon$ for the varying γ values. Consistent with the bound, the cardinality is largely stable for larger weights ($\gamma \geq 0.2$) and grows sharply as $\gamma \to 0$, indicating that the constant term B dominates for moderate to large weights while the A/γ term becomes relevant only for very small γ. Fitting $f(\gamma) = A/\gamma + B$ at fixed ε yields $R^2 \in [0.627, 0.916]$ confirming the $1/\gamma$ dependence, with the tightest fit at $\varepsilon = 0.3$. We continue this discussion in Appendix A.5.

5 Related Work

Early-Exit Neural Networks The field of early-exit neural networks has seen an evolution from initial deeply supervised nets [Lee et al., 2015] and sparse supervision [Teerapittayanon et al., 2016] to more advanced architecture like MSDNet [Huang et al., 2017] connecting latent features across depth and scale, and SDN [Kaya et al., 2019] leveraging intermediate classifiers to avoid overthinking, i.e., unnecessary and even destructive feature processing, and anticipate misclassifications early on. Han et al. [2023] argues that poor performance of early classifiers affects the final output and hence introduces a knowledge distillation path of the final classifier to the intermediate ones. All of these

works optimize the network and additional classifiers via a joint training regime and consider entropy based confidence thresholds as exit policy. Recently, Regol et al. [2024] introduced jointly trained gating networks that decide when to exit based on a fixed early-exit backbone network. Bajpai and Hanawal [2025] moved beyond static thresholds relying on multi-arm bandit policies to adapt to test time distributions. Kubaty et al. [2025a] showed that even calibrated models can lead to an insufficient compute-accuracy trade-off and suggested to focus instead on failure prediction metrics. These gating-based training regimes are not considered in this work as they can not be cast into our theoretical framework [Scardapane et al., 2020, Kubaty et al., 2025b].

Optimal Control for Deep Learning The intrinsic connections between time-continuous Neural ODEs and time-discrete ResNets lead to numerous advances both theoretically and practically. Benning et al. [2019] studied numerical methods for deep learning models and constrains the time discretization to lie on a simplex to form sparse time steps. Understanding the forward pass as integration time horizon also lead to works aiming for minimal time neural ODEs [Miao and Gatsis, 2024]. Studying asymptotic behavior of neural networks in terms of depth is crucial to guide design choices even for finite time implementations. Typically, the goal is to drive a dynamic system to a desired set of states, which in asymptotics, will be spent most of the time. These turnpikes have been investigated by Esteve-Yagüe and Geshkovski [2023] for l_1 regularized neural ODEs and by Faulwasser et al. [2024] for ResNets. Püttschneider and Faulwasser [2026] use the label smoothing to ensure finite minimizers of the cross-entropy loss and show turnpike behavior with respect to these minimizers. Püttschneider et al. [2025] study the implications of the steady-state property of ResNets in settings without reachability properties. For the cross-entropy loss and ℓ_1 weight decay as integral regularizer, Esteve-Yagüe and Geshkovski [2023] show that the hidden states converge to a stationary point representing a trade-off between the minimization of the loss and the weight decay. However, previous work did not explore the close relations of early-exiting and the optimal control perspective on deep learning. In particular, our turnpike results for entropy and early-exiting criteria appear to be the first of their kind.

6 Conclusion

In this paper, we analyze the joint training of early-exit ResNets taking an optimal control perspective. To this end, we formulate their training as an optimal control problem and we link early exiting to the occurrences of the turnpike phenomenon. Under a mild asymptotic reachability assumption, we derive new turnpike notions for the cross-entropy loss, the accuracy, and the early-exiting criteria entropy and maximum softmax probability. These turnpike guarantees yield explicit bounds on the exit depth sufficient for correct classification of the training data, independent of the total network depth. This provides a theoretical explanation for the success of jointly trained early-exiting networks and justify their ability to avoid overthinking through depth-bounded, confidence-driven inference. We support these findings with experiments on CIFAR-10/100 and TinyImageNet, showing that the turnpike similarity manifests in both the hidden-state trajectories and the resulting early-exit inference behavior.

Future work should broaden the reachability assumptions to architectures beyond fully connected ResNets with ReLU activation. Moreover, the formal verification of these assumptions is of major interest. We remark that while our theoretical analysis extends to other residual-based architectures, such as vision transformers [Dosovitskiy et al., 2021], their reachability and turnpike behavior should be investigated experimentally. Furthermore, given the relevance of early-exiting in transformer-based large language models [Sui et al., 2025], applying the proposed framework to such architectures is a promising direction.

Similarly, the analysis of turnpike properties for architectures without skip connections, such as VGG networks, is an interesting direction for future work, for a first step, we refer to Püttschneider et al. [2026]. In addition, while our theoretical results are centered around training performance, future research will aim to close the gap to statistical learning theory and to provide generalization guarantees. An interesting avenue in this regard is the analysis via out-of-distribution methods like [Meza De la Jara et al., 2025].

Acknowledgments

This work has been partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) via the TRR 391 Spatio-temporal Statistics for the Transition of Energy and Transport (project 520388526) and the research unit Active Learning for Systems and Control (ALeSCo) (project 535860958).

References

D. J. Bajpai and M. K. Hanawal. Beyond greedy exits: Improved early exit decisions for risk control and reliability. In *Advances in Neural Information Processing Systems*, volume 38, pages 170634–170659, 2025.

R. Bellman. The theory of dynamic programming. *Bulletin of the American Mathematical Society*, 60(6):503–515, 1954.

M. Benning, E. Celledoni, M. J. Ehrhardt, B. Owren, and C.-B. Schönlieb. Deep learning as optimal control problems: Models and numerical methods. *Journal of Computational Dynamics*, 6(2): 171–198, 2019.

D. P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 2005. ISBN 978-1-886529-26-7.

V. Boltyanskii, R. Gamkrelidze, and L. Pontryagin. On the theory of optimal processes. *Doklady Akademii Nauk SSSR*, 110:7–10, 1956.

M. Chen, M. Lin, K. Li, Y. Shen, Y. Wu, F. Chao, and R. Ji. CF-ViT: A general coarse-to-fine method for vision transformer. In *AAAI Conference on Artificial Intelligence*, 2023.

R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31, pages 6572–6583, 2018.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

R. Dorfman, P. Samuelson, and R. Solow. *Linear Programming and Economic Analysis*. McGraw-Hill, New York, 1958.

A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

C. Esteve, B. Geshkovski, D. Pighin, and E. Zuazua. Large-time asymptotics in deep learning. *arXiv preprint arXiv:2008.02491*, 2020.

C. Esteve-Yagüe and B. Geshkovski. Sparsity in long-time control of neural ODEs. *Systems & Control Letters*, 172:105452, 2023.

T. Faulwasser and L. Grüne. Turnpike properties in optimal control: An overview of discrete-time and continuous-time results. In *Handbook of Numerical Analysis*, volume 23, pages 367–400. 2022.

T. Faulwasser, A.-J. Hempel, and S. Streif. On the turnpike to design of deep neural networks: Explicit depth bounds. *IFAC Journal of Systems and Control*, 30:100290, 2024.

B. Geshkovski. *Control in moving interfaces and deep learning*. PhD thesis, Universidad Autónoma de Madrid, 2021.

Y. Han, D. Han, Z. Liu, Y. Wang, X. Pan, Y. Pu, C. Deng, J. Feng, S. Song, and G. Huang. Dynamic perceiver for efficient visual recognition. In *IEEE/CVF International Conference on Computer Vision*, pages 5992–6002, 2023.

M. Hardt and T. Ma. Identity matters in deep learning. In *International Conference on Learning Representations*, 2017.

K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, 2016a.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016b.

G. Huang, D. Chen, T. Li, F. Wu, L. van der Maaten, and K. Q. Weinberger. Multi-scale dense networks for resource efficient image classification. In *International Conference on Learning Representations*, 2017.

Y. Kaya, S. Hong, and T. Dumitras. Shallow-deep networks: Understanding and mitigating network overthinking. In *International Conference on Machine Learning*, 2019.

J. Kim, M. Kim, and B. Mozafari. Provable memorization capacity of transformers. In *International Conference on Learning Representations*, 2023.

A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

P. Kubaty, F. Szatkowski, M. Jazbec, and B. Wójcik. Failure prediction is a better performance proxy for early-exit networks than calibration. In *NeurIPS 2025 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2025a.

P. Kubaty, B. Wójcik, B. T. Krzepkowski, M. Michaluk, T. Trzcinski, J. Pomponi, and K. Adamczewski. How to train your multi-exit model? analyzing the impact of training strategies. In *International Conference on Machine Learning*, 2025b.

C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply-supervised nets. In *International Conference on Artificial Intelligence and Statistics*, 2015.

Q. Li, L. Chen, C. Tai, and E. Weinan. Maximum principle based algorithms for deep learning. *The Journal of Machine Learning Research*, 18(1):5998–6026, 2017.

M. Liu, T. Xie, X. Cheng, J. Deng, M. Yang, X. Wang, and M. Liu. FocusedDropout for convolutional neural network. *Applied Sciences*, 12(15), 2022a.

Z. Liu, Z. Xu, H.-J. Wang, T. Darrell, and E. Shelhamer. Anytime dense prediction with confidence adaptivity. In *International Conference on Learning Representations*, 2022b.

A. W. Marshall, I. Olkin, and B. C. Arnold. *Inequalities: Theory of Majorization and Its Applications*. Springer, New York, 2 edition, 2011.

Y. Matsubara, M. Levorato, and F. Restuccia. Split computing and early exiting for deep learning applications: Survey and research challenges. *ACM Computing Surveys*, 55(5), 2022.

L. W. McKenzie. Turnpike theory. *Econometrica*, 44(5):841–865, 1976.

I. Meza De la Jara, C. Rodriguez-Opazo, D. Teney, D. Ranasinghe, and E. Abbasnejad. Mysteries of the deep: Role of intermediate representations in out of distribution detection. In *Advances in Neural Information Processing Systems*, volume 38, pages 110728–110762, 2025.

K. Miao and K. Gatsis. How deep do we need: Accelerating training and inference of neural ODEs via control perspective. In *International Conference on Machine Learning*, 2024.

Q. Nguyen and M. Hein. Optimization landscape and expressivity of deep CNNs. In *International Conference on Machine Learning*, 2018.

N. Passalis, J. Raitoharju, A. Tefas, and M. Gabbouj. Efficient adaptive inference for deep convolutional neural networks using hierarchical early exits. *Pattern Recognition*, 105:107346, 2020.

J. Püttschneider, S. Heilig, A. Fischer, and T. Faulwasser. Towards an optimal control perspective of resnet training. In *ICML 2025 Workshop on High-dimensional Learning Dynamics*, 2025.

J. Püttschneider and T. Faulwasser. On dissipativity of cross-entropy loss in training ResNets — a turnpike towards architecture search. *Automatica*, 186:112767, 2026.

J. Püttschneider, S. Heilig, L. Adilova, A. Fischer, and T. Faulwasser. Turnpikes in deep learning: Beyond resnets and neural odes? In *2026 European Control Conference (ECC)*, pages 1910–1916, 2026.

H. Rahmath P, V. Srivastava, K. Chaurasia, R. G. Pacheco, and R. S. Couto. Early-exit deep neural network – a comprehensive survey. *ACM Computing Surveys*, 57(3), 2024.

F. Regol, J. Chataoui, and M. Coates. Jointly-learned exit and inference for a dynamic neural network. In *International Conference on Learning Representations*, 2024.

D. Ruiz-Balet and E. Zuazua. Neural ODE control for classification, approximation, and transport. *SIAM Review*, 65(3):735–773, 2023.

K. Sakamoto and I. Sato. End-to-end training induces information bottleneck through layer-role differentiation: A comparative analysis with layer-wise training. *Transactions on Machine Learning Research*, 2024.

S. Scardapane, M. Scarpiniti, E. Baccarelli, and A. Uncini. Why should we add early exits to neural networks? *Cognitive Computation*, 12(5):954–966, 2020.

T. Schuster, A. Fisch, J. Gupta, M. Dehghani, D. Bahri, V. Tran, Y. Tay, and D. Metzler. Confident adaptive language modeling. In *Advances in Neural Information Processing Systems*, volume 35, pages 17456–17472, 2022.

Y. Sui, Y.-N. Chuang, G. Wang, J. Zhang, T. Zhang, J. Yuan, H. Liu, A. Wen, S. Zhong, N. Zou, H. Chen, and X. Hu. Stop overthinking: A survey on efficient reasoning for large language models. *Transactions on Machine Learning Research*, 2025.

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

S. Teerapittayanon, B. McDanel, and H. Kung. BranchyNet: Fast inference via early exiting from deep neural networks. In *International Conference on Pattern Recognition*, pages 2464–2469, 2016.

E. Trélat and E. Zuazua. The turnpike property in finite-dimensional nonlinear optimal control. *Journal of Differential Equations*, 258(1):81–114, 2015.

J. von Neumann. Über ein ökonomisches Gleichungssystem und eine Verallgemeinerung des Brouwerschen Fixpunktsatzes. In *Ergebnisse eines Mathematischen Seminars*. 1938.

W. Zhou, C. Xu, T. Ge, J. McAuley, K. Xu, and F. Wei. BERT loses patience: Fast and robust inference with early exit. In *Advances in Neural Information Processing Systems*, volume 33, pages 18330–18341, 2020.

Overview of Appendix

- Appendix A: Experimental setup and additional results on CIFAR-100 and TinyImageNet
- Appendix B, C, D: Background on Early-Exit Networks and specific output design
- Appendix E: Detailed proofs of main results
- Appendix F: Scope and Limitations of our findings

A Experimental Supplements

This appendix supplements the experiments of Section 4 with additional results and ablations. Appendix A.1 details the experimental setup. Appendix A.2 extends the turnpike observations to homogeneous ResNets and additional datasets (CIFAR-100, TinyImageNet). Appendix A.3 complements the inference-time analysis with entropy and SDN confusion score trajectories, showing how intermediate predictions progressively converge toward the final output across network depth, and extends the early-exiting results by examining exit depths for varying backbone depths, demonstrating that these also exhibit the characteristic turnpike similarity across network depths and supervision frequencies. Appendix A.4 examines parameter sharing in early-exits enabled by the proposed exit architecture. Finally, Appendix A.5 ablates the stage-cost weight γ, which controls the relative importance of intermediate output losses.

A.1 Setup

The experimental setup is based on He et al. [2016a]. The networks used in our experiments use a pre-activation ResNet with $N = 54$ residual blocks, with two convolutional layers per residual block, which corresponds to a ResNet-110 in He et al. [2016a]. The standard ResNet architecture consists of an input embedding with of a 3x3 convolution with 16 filters followed by a batch normalization and ReLU. This is followed by $N = 54$ residual blocks organized in three stages, each with 18 residual blocks. Within these stages, The convolutional layers use 16, 32, and 64 filters, respectively. The shallower networks with $N = 36$ and $N = 18$ residual blocks are adapted from this architecture by reducing the number of residual blocks in each stage to 12 and 6 residual blocks respectively. The output block consists of a batch normalization layer, a ReLU activation, global average pooling, and a final linear layer that predicts the logits. Similar to He et al. [2016a], we train for 165 epochs with a batch size of 128 using SGD with an initial learning rate of $\alpha = 0.1$ and a momentum of $\mu = 0.9$. After 82 and 123 epochs, the learning rate is reduced by a factor of 10.

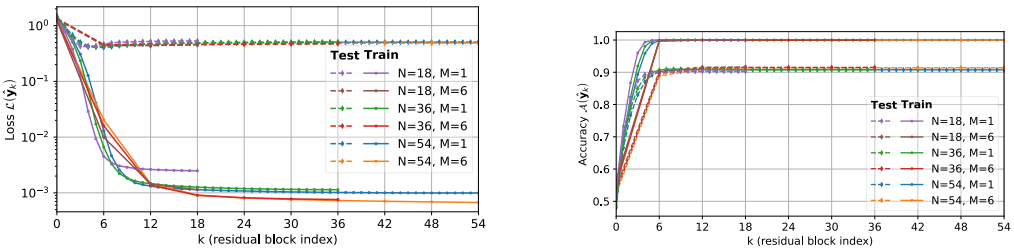
In the joint training formulations, we use a constant weight of $\gamma = 0.02$ for networks in which early-exits are attached to every residual block. This choice assigns approximately equal total weight to the 54 early-exits and to the final output of the ResNet-110, which is weighted by one. When early-exits are attached only to every sixth residual block, we accordingly increase the stage-cost weight to $\gamma = 0.12$. The learning rate for the additional parameters associated with the intermediate outputs is set to α/γ to ensure that their effective learning rate matches that of the final output layer. For each of these states use a separate batch norm statistics, while sharing the batch norm weights and bias for the ResNets sharing weights for early exit.

For homogeneous ResNets with constant state dimension, the number of convolutional filters is kept constant at 64 for CIFAR-10/CIFAR-100 across all residual blocks as well as the input embedding. Apart from this modification, their architecture is identical to the standard ResNet.

All results were obtained using PyTorch (version 2.4.1) on an NVIDIA RTX A6000 (48 GB) and NVIDIA A40 (48 GB).

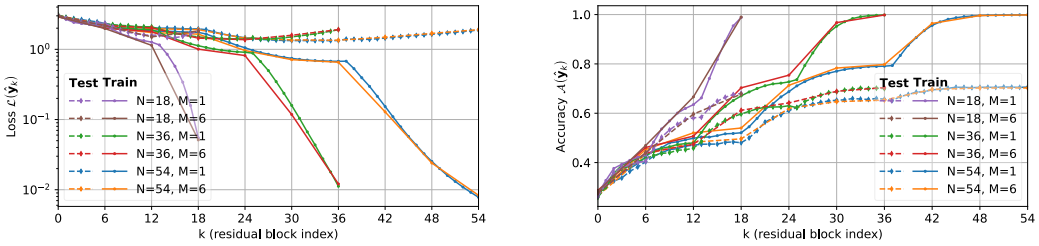
A.2 Can We Observe the Turnpike for Further Architectures and Datasets? (cont. from Section 4.2)

In addition to the results on CIFAR-10 presented in the main paper, we also conduct experiments on CIFAR-100 [Krizhevsky, 2009] and TinyImageNet [Deng et al., 2009] and consider a homogeneous ResNet architecture with a fixed number of filters per residual block.



(a) (b)

Figure 7: (a) Loss and (b) Accuracy trajectories for CIFAR-10 for the homogeneous ResNet architecture over the depth in terms of residual blocks obtained for ResNets with residual blocks $N \in \{18, 36, 54\}$ and with two levels of early-exit supervision: every residual block ($M = 1$) and every sixth ($M = 6$).



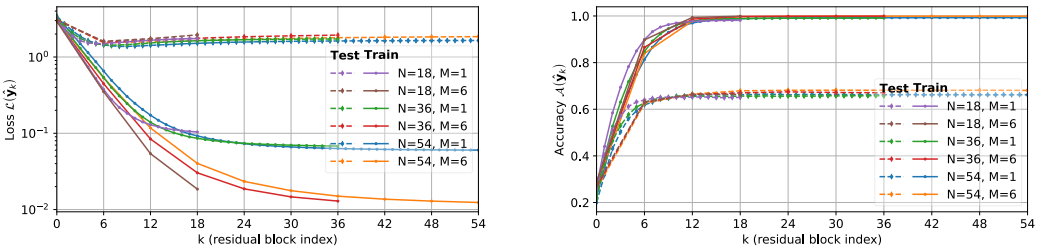
(a) (b)

Figure 8: (a) Loss and (b) Accuracy trajectories for CIFAR-100 for the standard ResNet architecture over the depth in terms of residual blocks obtained for ResNets with residual blocks $N \in \{18, 36, 54\}$ and with two levels of early-exit supervision: every residual block ($M = 1$) and every sixth ($M = 6$).

The kinks observed in the loss and accuracy curves of the standard ResNets in Figure 4 coincide with stage transitions, at which the feature dimension (number of channels) changes and spatial downsampling is applied. At these transitions, the number of trainable parameters per residual block increases by a factor of four, while the state dimension is reduced by a factor of two. Such abrupt changes in representational capacity likely underlie the observed non-smoothness in the training trajectories. To isolate the effect of these stage-wise architectural changes, we also consider homogeneous ResNets with a constant number of convolutional filters, 64, across all residual blocks. For CIFAR-10, the resulting loss and accuracy trajectories are shown in Figure 7 and for CIFAR-100 in Figure 9. In contrast to the standard architecture, the homogeneous networks yield markedly smoother curves without visible kinks, supporting the interpretation that the artifacts arise from stage-wise architectural variation.

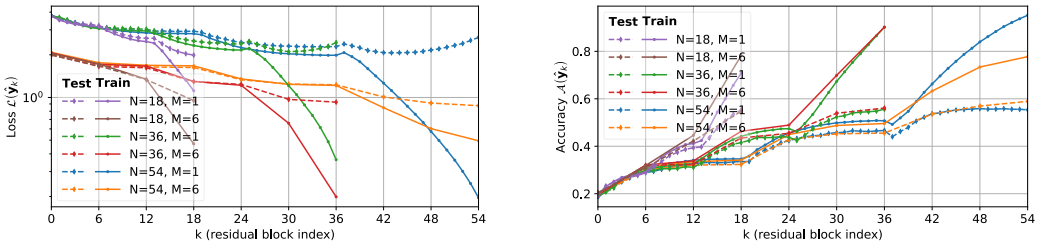
Due to their increased capacity, the homogeneous ResNets reach substantially lower training losses than their standard counterparts. Interestingly, the similarity between the depth-36 and depth-54 trajectories becomes even more pronounced in the homogeneous setting. However, despite having more parameters compared to the standard ResNets, the homogeneous models achieve lower test accuracy, indicating poorer generalization.

Figures 8 and 9 present the loss and accuracy curves on CIFAR-100 for the standard and homogeneous ResNet architectures, respectively. The observed behavior closely resembles that of CIFAR-10 in both loss and accuracy. Owing to the larger number of classes and the increased task difficulty, CIFAR-100 yields higher training and test losses and lower accuracies than CIFAR-10, and the corresponding loss and accuracy curves converge more slowly. In particular, losses decrease and accuracies increase at a reduced rate compared to CIFAR-10. Nevertheless, consistent convergence patterns across network depth and early-exit configurations persist at this reduced performance level. These observations indicate that both CIFAR-10 and CIFAR-100 ResNets exhibit turnpike behavior consistent with our theoretical analysis.



(a) (b)

Figure 9: (a) Loss and (b) Accuracy trajectories for CIFAR-100 for the homogeneous ResNet architecture over the depth in terms of residual blocks obtained for ResNets with residual blocks $N \in \{18, 36, 54\}$ and with two levels of early-exit supervision: every residual block ($M = 1$) and every sixth ($M = 6$).

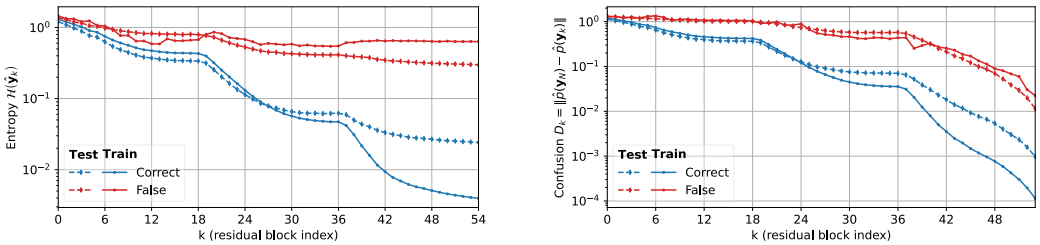


(a) (b)

Figure 10: (a) Loss and (b) Accuracy trajectories for TinyImageNet for ResNet over the depth in terms of residual blocks obtained for ResNets with residual blocks $N \in \{18, 36, 54\}$ and with two levels of early-exit supervision: every residual block ($M = 1$) and every sixth ($M = 6$).

Figure 10 presents the loss and accuracy trajectories for the TinyImageNet dataset [Deng et al., 2009]. Unlike in the CIFAR experiments, the increased task difficulty prevents any network from reaching perfect training accuracy within the available depth, with losses remaining above 10^{-1} and accuracies below $95.2\,\%$. This indicates that all networks are limited by their capacity and depth. Moreover, loss as well as accuracy improve contentiously over the depths of the networks. The loss decay rate satisfies the exponential bound required by Assumption 3.1 throughout all layers, and even exceeds it toward the final layers. From the optimal control perspective, this implies that these depths does not suffice to fully reach the turnpike, suggesting that increasing network depth would lead to continued performance gains. Nevertheless, the characteristic signature of the turnpike phenomenon persists, the trajectories of the training metrics from networks of different depths and supervision densities follow a markedly similar path through the intermediate layer, before diverging near the final blocks. This divergence near the end reflects a more fundamental issue: due to the limited capacity of these networks, the joint training formulation in early-exiting imposes a much stronger trade-off between intermediate outputs and the final prediction than in the CIFAR setting, ultimately constraining the performance of the final classifier.

Despite the larger gaps in training performance, the test accuracies are remarkably similar across networks of varying depth and supervision frequencies and exhibit a much stronger turnpike behavior than the corresponding training trajectories. In addition, due to the limited capacity of the network, the trade-off between intermediate output and final classifier of the joint training introduces a notable gap between the final test accuracies. The ResNets-110 with $N = 54$ residual blocks and early-exits at every block ($M = 1$) reaches $58.3\,\%$ test accuracy and the network with supervision at every sixth block ($M = 6$) reaches $55.9\,\%$, reflecting a stronger deterioration compared to the $62.4\,\%$ accuracy reported for the ResNet-110 without early-exits in Liu et al. [2022a, Table 3].



(a) (b)

Figure 11: (a) Entropy and (b) SDN confusion for training and test data across network depth, conditioned on the final prediction outcome on CIFAR-10 with ResNet $N = 54$, $M = 1$.

A.3 Turnpikes during Test-Time and Early Exiting? (cont. from Section 4.3)

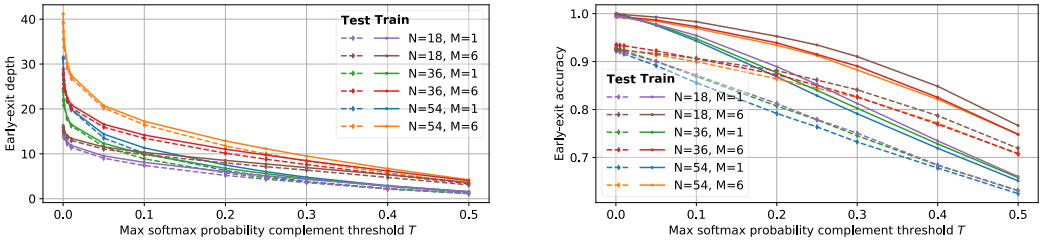
In this section, we further analyze the turnpike behavior on test data and during early-exit inference, considering the exit depth and entropy as the exit criterion.

We begin by examining the convergence of entropy for test samples. In Figure 11a, and similar to Figure 5, we plot the average entropy of samples conditioned on their final prediction outcome for the CIFAR-10 dataset split by training and test samples. The plot shows that the entropy consistently decreases for both the training and test samples, which implies that all samples that are finally classified correctly become increasingly confident in their final correct classification. Thus, these samples are amenable to early-exit by an entropy threshold, thereby reducing the computational demand for inference. The incorrectly classified samples exhibit on average a higher entropy values, implying that they do not reach the confidence threshold required for early-exiting. Thus, a well tuned early-exiting threshold can early-exit the correctly classified samples and avoid overthinking of samples that can be classified correctly before the final layer [Kaya et al., 2019].

To quantify this overthinking effect, Kaya et al. [2019] introduced a confusion score measuring the difference in predicted softmax probabilities between intermediate exits and the final layer. Figure 11b plots the corresponding trajectories, which shows the convergence of the intermediate predictions toward the final prediction, consistent with the turnpike phenomenon. This provides an explanation for why overthinking primarily manifests between the earlier intermediate outputs and the deeper layers, samples are already committed to their final prediction well before the terminal layer is reached.

We now analyze the turnpike behavior of the early-exit depths, complementing our previous analysis of early-exit accuracies. Figure 12 shows the exit depths and accuracies on CIFAR-10 using the max softmax probability complement as the exit criterion. Most notably, however, the exit depth for low thresholds start close to the full backbone depths, with increasing exiting thresholds, they rapidly decrease and converge toward one another. Notably, the trade-off between accuracy and computational cost is clearly visible when comparing the supervision levels. Networks with exits placed every $M = 6$ blocks are higher than their counterparts with exits at every block $M = 1$, the exit depth show that this comes at the price of an increased early-exiting depth and therefore higher computational demand for the inference. Notably, the trade-off between accuracy and computational cost is clearly visible when comparing supervision frequencies. Networks with exits placed every $M = 6$ residual blocks achieve higher early-exit accuracy than their $M = 1$ counterparts, but at the cost of greater average exit depth, implying higher computational demand at inference time. In addition, the early-exit depths for training and test data are virtually indistinguishable, which further supports the generalization of the turnpike property to the inference time.

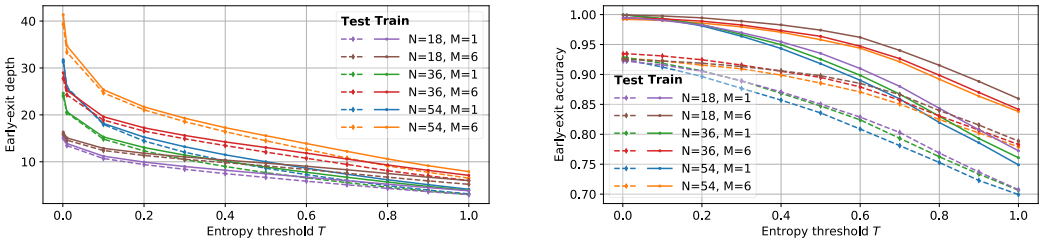
Early-exiting results for the entropy as exiting criterion are show in Figure 13, which show the same turnpike behavior as the max softmax probability criterion. From an optimal control perspective, both criteria robustly detect the turnpike at inference time. As the turnpike corresponds to the overthinking regime, this enables early exits upon reaching it, avoiding unnecessary computation.



(a) Early-exit depths for varying thresholds

(b) Early-exit accuracies for varying thresholds.

Figure 12: Early-exiting results with the max softmax probability complement as exiting criterion on CIFAR-10 for ResNets with residual blocks $N \in \{18, 36, 54\}$ with two levels of early-exit supervision: every residual block ($M = 1$) and every sixth ($M = 6$).



(a) Early-exit depths for varying thresholds

(b) Early-exit accuracies for varying thresholds.

Figure 13: Early-exiting results with the entropy as exiting criterion on CIFAR-10 for ResNets with residual blocks $N \in \{18, 36, 54\}$ with two levels of early-exit supervision: every residual block ($M = 1$) and every sixth ($M = 6$).

A.4 Shared Early-Exit Parameters

The early-exit architecture

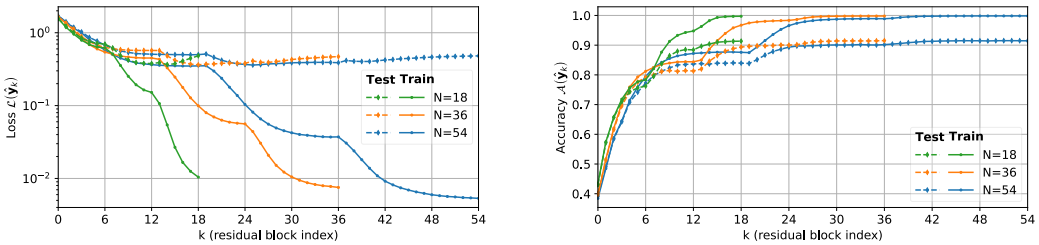
$$\hat{y}_k^{(i)} = \mathcal{O}_k(x_k^{(i)}, \mathbf{w}_{\mathcal{O},k}) = \mathcal{O}(\mathcal{S}_{N-1}(\cdots \mathcal{S}_k(x_k^{(i)}, \mathbf{w}_{\mathcal{S},k}), \cdots, \mathbf{w}_{\mathcal{S},N-1}), \mathbf{w}_{\mathcal{O}})$$

admits two design choices regarding parameterization. Since the layers used by the early-exits are already part of the backbone ResNet, the corresponding parameters can either be shared with the backbone or instantiated separately for each early-exit. We note that reusing parameters of later skip connections in the stage cost of earlier residual blocks formally breaks the causal structure of the OCP, though this does not affect the training objective and its analysis, which is the focus of this work. Sharing the early-exit parameters with the backbone network allows for earlier exiting without introducing additional trainable parameters. However this comes at the cost of reduced model capacity. Figure 14 shows the loss and accuracy trajectories on CIFAR-10 for the standard ResNet when parameters are shared between the backbone and the early-exits. While the training loss is comparable to that obtained with separate early-exit parameters as presented in the main Figure 4a, the test accuracy is significantly lower. This degradation in performance is consistent with the reduced model capacity induced by parameter sharing and indicates inferior generalization compared to the separate-parameter configuration.

A.5 Effect of Stage-Cost Weight on the Turnpike Behavior? (cont. from Section 4.4)

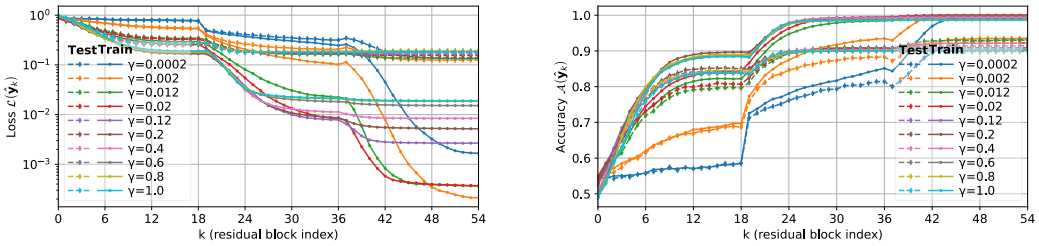
We continue our ablation of the stage-cost weight from Section 4.4 using the same networks as before, ResNets with $N = 54$ residual blocks and early-exits at every block $M = 1$ trained using varying stage cost weights $\gamma \in \{0.0002, 0.002, 0.012, 0.02, 0.12, 0.2, 0.4, 0.6, 0.8, 1.0\}$.

The resulting accuracies and loss curves in Figure 15 show the trade-off between the intermediate outputs and the final classifier. For weights $\gamma \geq 0.2$, the intermediate curves cluster tightly across early layers, but final-block performance deteriorates noticeably. At the other extreme, small weights $\gamma \leq 0.002$ fail to achieve a meaningful convergence in the earlier layers, making them unsuitable for early-exiting. The default values $\gamma = 0.02$ used in our experiments achieves a good balance between intermediate and final-output performance.



(a) (b)

Figure 14: (a) Loss and (b) Accuracy trajectories for CIFAR-10 for the standard ResNet architecture with shared early-exit parameters over the depth in terms of residual blocks obtained for ResNets with residual blocks $N \in \{18, 36, 54\}$ and early-exits at every residual block ($M = 1$).



(a) (b)

Figure 15: (a) Loss and (b) Accuracy trajectories for CIFAR-10 for the ResNet over the depth in terms of residual blocks obtained for ResNets with $N = 54$ residual blocks trained with varying stage-cost weights and early-exits at every residual block ($M = 1$).

B ResNet Basics

We briefly revisit the core components of the ResNet-v2 architecture [He et al., 2016a,b] which is considered here.

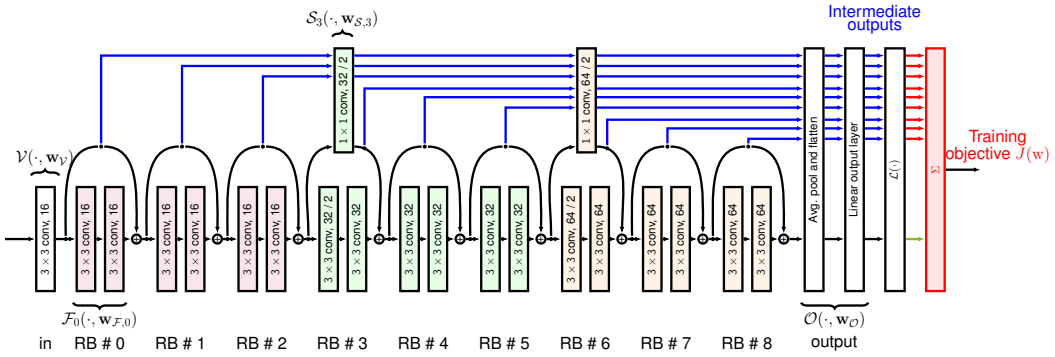


Figure 16: Illustration of a ResNet-v2 architecture while training with our proposed stage cost regularizer penalizing intermediate outputs. We propose to forward the hidden states through the network's non-trivial skip connections to account for dimensionality projections.

As illustrated in Figure 16, in the forward pass a sample $x^{(i)}$ is initially embedded into a latent space via $\mathcal{V}(x^{(i)}, \mathbf{w}_{\mathcal{V}})$, then N residual blocks process the input to eventually forward the final representation $x_N^{(i)}$ to the output head $\mathcal{O}(x_N^{(i)}, \mathbf{w}_{\mathcal{O}})$. We denote the trainable weights as $\mathbf{w} = [\mathbf{w}_{\mathcal{V}}, \mathbf{w}_0, ..., \mathbf{w}_{N-1}, \mathbf{w}_{\mathcal{O}}]$, where $\mathbf{w}_k = [\mathbf{w}_{\mathcal{F},k}, \mathbf{w}_{\mathcal{S},k}]$ corresponds to the kth residual block $f_k(x_k^{(i)}, \mathbf{w}_k)$ that is composed of the convolutional layers $\mathcal{F}_k(x_k^{(i)}, \mathbf{w}_{F,k})$ and the skip connection

$\mathcal{S}_k(x_k^{(i)}, \mathbf{w}_{\mathcal{S},k})$ with optional parameters $\mathbf{w}_{\mathcal{S},k}$. If two consecutive blocks have the same dimension, an identity skip connection is employed, but if there is a change in dimensionality, a 1x1 convolution is used as a projection layer. For the ease of reading, we denote $\mathcal{K}_\mathcal{S}$ as the index set of blocks having non-trivial skip connections. Overall, a ResNet with N residual blocks comprises the following system equations:

$$\begin{aligned} x_0^{(i)} &= \mathcal{V}(x^{(i)}, \mathbf{w}_\mathcal{V}) \ , \\ x_{k+1}^{(i)} = f_k(x_k^{(i)}, \mathbf{w}_k) &= \begin{cases} x_k^{(i)} & +\mathcal{F}_k(x_k^{(i)}, \mathbf{w}_{\mathcal{F},k}), \ k \notin \mathcal{K}_\mathcal{S} \\ \mathcal{S}_k(x_k^{(i)}, \mathbf{w}_{\mathcal{S},k}) & +\mathcal{F}_k(x_k^{(i)}, \mathbf{w}_{\mathcal{F},k}), \ k \in \mathcal{K}_\mathcal{S}, \end{cases} \quad \forall k = 0, ..., N-1 \ , \\ \hat{y}^{(i)} &= \mathcal{O}(x_N^{(i)}, \mathbf{w}_\mathcal{O}) \ . \end{aligned} \tag{5}$$

B.1 Joint Training of Early-Exit ResNets

In this work, we consider standard ResNet-v2 [He et al., 2016a] trained on classification datasets consisting of D samples of features $x^{(i)}$ and labels $y^{(i)}$

$$\mathbb{D} = \left\{ \left(x^{(1)}, y^{(1)} \right), ..., \left(x^{(D)}, y^{(D)} \right) \right\}$$

with C classes.

For training the ResNet on the entire dataset, we denote the propagation of all samples by boldface variables, e.g., $\mathbf{x} = \left[x^{(1)}, x^{(2)}, \ldots, x^{(D)} \right]$. Assuming a constant state dimension in the hidden residual blocks, the propagation of the data in the residual blocks can be understood as a time-varying dynamical system: $\mathbf{x}_{k+1} = f_k(\mathbf{x}_k, \mathbf{w}_k) = \mathbf{x}_k + \mathcal{F}_k(\mathbf{x}_k, \mathbf{w}_{\mathcal{F},k})$, with stacked data $\mathbf{x}_k$ as state variable and the trainable parameters $\mathbf{w}_k$ as control inputs. The objective function of standard neural network training is the loss $\mathcal{L}$ evaluated at the network output, which corresponds to the terminal penalty in optimal control. In addition to common weight decay, optimal control approaches to ResNet training have suggested including a stage cost regularizer $\ell_k(\mathbf{x}_k, \mathbf{w}_k)$ that depends also on the states $\mathbf{x}_k$ [Esteve et al., 2020, Faulwasser et al., 2024, Miao and Gatsis, 2024]. This leads to the training objective

$$J(\mathbf{w}) = \sum_{k=0}^{N-1} \ell_k(\mathbf{x}_k, \mathbf{w}_k) + \mathcal{L}(\hat{\mathbf{y}}).$$

This corresponds to the typical joint training formulations for the training of early-exits [Teerapittayanon et al., 2016, Matsubara et al., 2022]

$$\inf_{\mathbf{w}} \ J_N(\mathbf{w}) = \sum_{k \in \mathcal{K}_\mathcal{O}} \gamma \mathcal{L}(\hat{\mathbf{y}}_k) + \mathcal{L}(\hat{\mathbf{y}}) \quad \text{subject to} \quad (5), (A.4), \ \forall k = 0, ..., N-1,$$

if the stage cost for residual blocks with an early-exits defined as the loss of the intermediate prediction weighted by $\gamma > 0$, $\ell_k(\mathbf{x}_k, \mathbf{w}_k) = \gamma \mathcal{L}(\hat{\mathbf{y}}_k)$, $k \in \mathcal{K}_\mathcal{O}$, and $\ell_k(\mathbf{x}_k, \mathbf{w}_k) = 0$ for residual blocks without early-exits.

For classification tasks, we first compute probabilities for each class using the softmax activation function from the output $\hat{y}^{(i)}$. This gives a probability vector $\hat{p}(\hat{y}) \in (0,1)^C$, which is element-wise defined as

$$[\hat{p}(\hat{y})]_c = \frac{e^{[\hat{y}]_c}}{\sum_{i=1}^C e^{[\hat{y}]_i}},$$

where by $[x]_j$ denotes the j-th element of the vector x. With the softmax probability vector, we compute the cross-entropy loss for the label y by taking the negative logarithm of the softmax probability for the class y

$$\mathcal{L}(\hat{y}) = -\log\left([\hat{p}(\hat{y})]_y \right).$$

Here, we omit the ground-truth label as an additional argument of the loss function. For the full dataset, we compute the empirical loss by averaging the loss over the dataset

$$\mathcal{L}(\hat{\mathbf{y}}) = \frac{1}{D} \sum_{i=1}^{D} \mathcal{L}(\hat{y}^{(i)}).$$

The entropy of the softmax probability distribution reads

$$\mathcal{H}(\hat{y}) = -\sum_{c=1}^{C} [\hat{p}(\hat{y})]_c \log\left([\hat{p}(\hat{y})]_c\right).$$

Similar to the cross-entropy loss, the entropy $\mathcal{H}$ of the output for all data samples $\hat{\mathbf{y}}$ is computed as

$$\mathcal{H}(\hat{\mathbf{y}}) = \frac{1}{D}\sum_{i=1}^{D} \mathcal{H}(\hat{y}^{(i)}).$$

The class with the maximum softmax probability gives the class prediction

$$\hat{c}(\hat{y}) = \underset{c}{\operatorname{argmax}} [\hat{p}(\hat{y})]_c.$$

For binary classification, $C = 2$, we adopt the convention that ties are broken in favor of the true class. Since the set of logits inducing a tie, i.e. $\{\hat{y} : [\hat{p}(\hat{y})]_1 = [\hat{p}(\hat{y})]_2\} \subset \mathbb{R}^2$, is a null set with respect to the Lebesgue measure on $\mathbb{R}^2$, this convention has no effect for continuously distributed logits in practice and is merely introduced to make the threshold results in the following section well-defined at the boundary. With this class prediction, we can compute the classification accuracy of the network over the entire dataset

$$\mathcal{A}(\hat{\mathbf{y}}) = \frac{\#\left\{i = 1, ..., D \middle| \hat{c}(\hat{y}^{(i)}) = y^{(i)}\right\}}{D}.$$

For the early-exit placement and the final output $\mathcal{K}_\mathcal{O} \cup \{N\} = \{k_1, \ldots, k_m\}$ with $k_1 < k_2 < \cdots < k_m = N$, we denote by $\operatorname{seq}\left(\mathcal{K}_\mathcal{O} \cup \{N\}\right)_j = k_{\min\{j,\, m\}}$ the index of the residual block with the j-th output, truncated at the final block N if $j > m$.

C Intermediate Output Architecture

In early-exiting, intermediate outputs must remain shallow for efficiency. ResNets provide a natural mechanism for this: if the parameters of the residual branches for $M \leq k \leq N-1$ are set to zero $\mathbf{w}_{\mathcal{F},k} = 0$, then the residual blocks are zero $\mathcal{F}_k(x_k^{(i)}) = 0$ for all samples $x^{(i)}$. Such that the hidden state $x_M^{(i)}$ is forwarded only by the skip connections of the following residual blocks, resulting in the output

$$\hat{y}^{(i)} = \mathcal{O}(\mathcal{S}_{N-1}(\cdots \mathcal{S}_M(x_M^{(i)}, \mathbf{w}_{\mathcal{S},M}), \cdots, \mathbf{w}_{\mathcal{S},N-1}), \mathbf{w}_\mathcal{O}).$$

Moreover, since most residual blocks have identity skip connections, the remaining number of layers the state $x_M^{(i)}$ gets propagated is comparatively low. Hence, we can use those non-trivial skip connections as building blocks for obtaining the intermediate output of state $x_M^{(i)}$, since, similar to the standard design of early-exits, they consist of few layers for computational efficiency [Rahmath P et al., 2024, Matsubara et al., 2022].

In addition, since the layers contained in this intermediate output architecture are already part of the backbone network, their parameters can either be shared with the backbone network or can be additional parameters for increased representational capacity. This way, we can re-interpret or fine-tune ResNets trained without our intermediate outputs, since all necessary parameters are at hand.

This leads to the definition of intermediate output heads $\mathcal{O}_k$, which predict $\hat{y}_k^{(i)}$ based on intermediate states $x_k^{(i)}$ via:

$$\hat{y}_k^{(i)} = \mathcal{O}_k(x_k^{(i)}, \mathbf{w}_{\mathcal{O},k}) = \mathcal{O}(\mathcal{S}_{N-1}(\cdots \mathcal{S}_k(x_k^{(i)}, \mathbf{w}_{\mathcal{S},k}), \cdots, \mathbf{w}_{\mathcal{S},N-1}), \mathbf{w}_\mathcal{O}), \quad \forall k = 0, \ldots, N-1.$$

Figure 16 illustrates the proposed flow of data through the skip connections of the original residual blocks.

This choice of intermediate output architecture enables the following theoretical guarantee, which establishes a connection to steady states (or controlled equilibria) of dynamic systems.

Theorem C.1 (Identity mapping)**.** *Consider the ResNet* (5)*, with the intermediate outputs chosen according to* (A.4)*. Let the intermediate outputs of residual block M produce the prediction $\hat{\mathbf{y}}_M$, and let the parameters of all following residual terms be zero, $\mathbf{w}_{\mathcal{F},k} = 0$, $k = M, ..., N-1$. Then all subsequent intermediate outputs and the final output are equal*

$$\hat{\mathbf{y}} = \hat{\mathbf{y}}_k = \hat{\mathbf{y}}_M, \tag{6}$$

for all $k = M, ..., N$.

Proof. If $\mathbf{w}_{\mathcal{F},k} = 0$, for the residual blocks $k = M, ..., N-1$, then $\mathcal{F}_k(\mathbf{x}_k, 0) = 0$ for each of these blocks. Hence, the state is being propagated by $\mathbf{x}_{k+1} = \mathcal{S}_k(\mathbf{x}_k, \mathbf{w}_{\mathcal{S},k})$ for all $k = M, ..., N-1$. This gives the output $\hat{\mathbf{y}} = \mathcal{O}(\mathbf{x}_N, \mathbf{w}_{\mathcal{O}})$ and the intermediate outputs (A.4), which corresponds to (6). □

D Asymptotic Early-Exiting

In early-exit architectures, determining the placement of intermediate classifiers is a critical hyperparameter choice. Common practice is to distribute these exits uniformly across the network depth.

Similar, when considering the asymptotic-depth regime in our later turnpike analysis, the question of how early-exit placement should scale with network depth arises naturally. One may fix the number of early-exits while letting the depth tend to infinity, which leads to asymptotically sparse exit positions. Alternatively, one may maintain a fixed spacing between exits, thereby preserving a dense placement; in this case, the number of early-exits increases asymptotically with the network depth.

Consider a sparse placement of M early-exits, where N is divisible by $M+1$, which leads to the joint training objective

$$\sum_{k=1}^{M} \gamma \mathcal{L}\left(\hat{\mathbf{y}}_{k\frac{N}{M+1}}\right) + \mathcal{L}\left(\hat{\mathbf{y}}_N\right),$$

where the spacing $\frac{N}{M+1}$ becomes sparse as the depth increases. Alternatively, maintaining a fixed distance by placing early-exits every M residual blocks

$$\sum_{k=1}^{\frac{N}{M}-1} \gamma \mathcal{L}\left(\hat{\mathbf{y}}_{kM}\right) + \mathcal{L}\left(\hat{\mathbf{y}}_N\right).$$

In the first variant, the depth of the first early-exit is $\frac{N}{M+1}$ and decreases with the depth of the backbone networks. In the second variant, the depth of the first early-exit is fixed at M. Thus, the first variant is not suitable for early-exiting, since the depth of the exit would be unbounded. Hence, we consider the second variant for the remainder of the paper.

E Turnpike Analysis of Early-Exiting

In this section, we present our main theoretical results on the turnpike behavior of early-exiting ResNets and provide their proofs.

In deep ResNets the later layers primarily act as refinements of the features. This behavior is further emphasized when using a stage cost loss, i.e., a regularization for the hidden states dependent on the considered loss function. Below, we prove specifically that a ResNet trained with a stage cost loss exhibits the turnpike phenomenon with respect to the loss from optimal control, i.e., for any given loss threshold, only a bounded number of intermediate outputs exceed this threshold. We establish this *loss turnpike* behavior under the assumption of exponential loss decay, which we subsequently verify for a specific ResNet architecture in Appendix E.1. This implies that, for most of the depth of the network, the hidden states remain near an optimal representation resulting in near zero loss for their intermediate outputs. Moreover, the loss turnpike directly implies that a trained ResNet also exhibits an *accuracy turnpike*, i.e., the number of intermediate outputs with less than perfect accuracy is bounded from above. These implications are illustrated in Figure 17.

Finally, in Appendix E.5, we show that combining accuracy turnpikes with the early-exit criterion guarantees the existence of an early-exit threshold that ensures both bounded early-exit depth and correct classification on the training data.

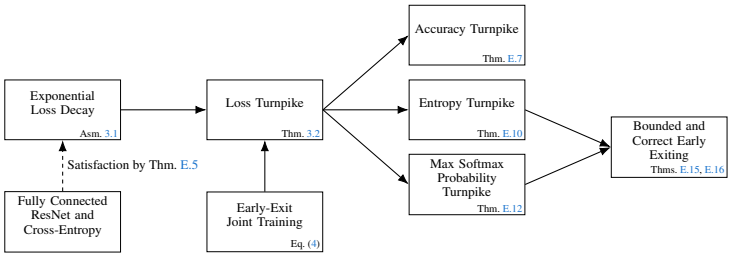


Figure 17: Overview of the main theoretical contributions.

Assumption E.1 (Exponential loss decay)**.** For the given training dataset $\mathbb{D}$, there exist constants $\rho \in [0, 1)$ and $\beta > 0$, and an infinite-depth ResNet with early-exits attached to each residual block, with parameters $\tilde{\mathbf{w}}_{\mathcal{V}}$ for the input block, $\tilde{\mathbf{w}}_k$ for the infinite sequence of residual blocks $k \in \mathbb{N}_{[0,\infty)}$, and $\tilde{\mathbf{w}}_{\mathcal{O},k}$ for the corresponding intermediate outputs, such that the training loss of the intermediate outputs $\tilde{\mathbf{y}}_k$ satisfies $\mathcal{L}(\tilde{\mathbf{y}}_k) \leq \beta\rho^k$, for $k \in \mathbb{N}_{[0,\infty)}$.

In essence, this assumption requires that any finite positive loss can be achieved with some network of finite depth. Next, we will show that under this assumption, the ResNet trained with stage cost loss exhibits the turnpike phenomenon. In Appendix E.1, we show that this Assumption holds for a specific ResNet architecture with fully connected layers. Moreover, a similar result for neural ODEs can be found in [Geshkovski, 2021, Proposition 7.4].

Theorem 3.2 (Loss turnpike in ResNet training)**.** *Suppose that Assumption 3.1 holds. Let a ResNet (1) with depth N be trained with stage cost loss weighted by $0 < \gamma \leq 1$ using the optimal control training formulation (4) with the early-exit placement $\mathcal{K}_{\mathcal{O}}$, and let its parameters $\mathbf{w}$ be δ-suboptimal for some fixed $0 \leq \delta < \infty$, i.e., $J_N(\mathbf{w}) \leq \inf_{\mathbf{w}'} J_N(\mathbf{w}') + \delta$, then the training objective is bounded independent of the depth N and the early-exit placement $\mathcal{K}_{\mathcal{O}}$ by*

$$J_N(\mathbf{w}) \leq \bar{J} = \frac{\gamma\beta}{1-\rho} + \beta + \delta,$$

and the ResNet exhibits a loss turnpike on the training data. More precisely, for a loss level $\varepsilon > 0$, consider the set of intermediate outputs with training losses exceeding ε, $\hat{\mathcal{Q}}_\varepsilon \doteq \{k \in \mathcal{K}_{\mathcal{O}} \cup \{N\} \,|\, \mathcal{L}(\hat{\mathbf{y}}_k) > \varepsilon\}$. Then, for every loss level $\varepsilon > 0$ there exists a constant $\kappa(\varepsilon) < \infty$, independent of the network depth N and the early-exit placement $\mathcal{K}_{\mathcal{O}}$, such that the cardinality of $\hat{\mathcal{Q}}_\varepsilon$ is bounded $\#\hat{\mathcal{Q}}_\varepsilon \leq \kappa(\varepsilon) = \frac{\bar{J}}{\gamma\varepsilon}$.

Proof. First, Assumption 3.1 gives that there exist ResNet parameters $\tilde{\mathbf{w}}$ and constants β and $\rho \in [0, 1)$, such that the training losses of the intermediate outputs $\tilde{\mathbf{y}}_k$ decays exponentially

$$\mathcal{L}(\tilde{\mathbf{y}}_k) \leq \beta\rho^k,$$

for early-exits attached to every residual block $k \in \tilde{\mathcal{K}}_{\mathcal{O}} \cup \{N\} = \{0, ..., N\}$. Hence, the infimum of the joint training objective J_N is upper bounded by the corresponding geometric series

$$\inf_{\mathbf{w}'} J_N(\mathbf{w}') \leq \sum_{k\in\mathcal{K}_{\mathcal{O}}} \gamma\mathcal{L}(\tilde{\mathbf{y}}_k) + \mathcal{L}(\tilde{\mathbf{y}}) \leq \sum_{k\in\mathcal{K}_{\mathcal{O}}} \gamma\beta\rho^k + \beta\rho^N \leq \sum_{k=0}^{N-1} \gamma\beta\rho^k + \beta\rho^N, \tag{7}$$

where $\tilde{\mathbf{y}}_k$ and $\tilde{\mathbf{y}}$ are the intermediate and final outputs of the ResNet with parameters $\tilde{\mathbf{w}}$ for the training data. Since $\rho \in [0, 1)$ and $N \geq 0$, we have that $\rho^N \leq 1$ and therefore

$$\sum_{k\in\mathcal{K}_{\mathcal{O}}} \gamma\beta\rho^k + \beta\rho^N \leq \sum_{k=0}^{N-1} \gamma\beta\rho^k + \beta\rho^N \leq \gamma\beta\frac{1-\rho^N}{1-\rho} + \beta\rho^N \leq \frac{\gamma\beta}{1-\rho} + \beta \doteq \bar{J}_{\text{inf}}.$$

Hence, the infimum of the training objective satisfies $\inf_{\mathbf{w}'} J_N(\mathbf{w}') \leq \bar{J}_{\text{inf}}$ for all depths N and all early-exit placements $\mathcal{K}_{\mathcal{O}}$. Therefore, for the δ-suboptimal parameters $\mathbf{w}$, we have $J_N(\mathbf{w}) \leq \inf_{\mathbf{w}'} J_N(\mathbf{w}') + \delta \leq \bar{J}_{\text{inf}} + \delta \doteq \bar{J}$. Moreover, from $0 < \gamma \leq 1$, we have $\mathcal{L}(\hat{\mathbf{y}}) \geq \gamma\mathcal{L}(\hat{\mathbf{y}}) = \gamma\mathcal{L}(\hat{\mathbf{y}}_N)$ and therefore

$$\sum_{k\in\mathcal{K}_{\mathcal{O}}\cup\{N\}} \gamma\mathcal{L}(\hat{\mathbf{y}}_k) \leq \sum_{k\in\mathcal{K}_{\mathcal{O}}} \gamma\mathcal{L}(\hat{\mathbf{y}}_k) + \mathcal{L}(\hat{\mathbf{y}}) = J_N(\mathbf{w}) \leq \bar{J}.$$

Moreover, with $\hat{\mathcal{Q}}_\varepsilon \doteq \{k \in \mathcal{K}_\mathcal{O} \cup \{N\} | \mathcal{L}(\hat{\mathbf{y}}_k) > \varepsilon\}$, each $k \in \hat{\mathcal{Q}}_\varepsilon$ contributes at least $\mathcal{L}(\hat{\mathbf{y}}_k) > \varepsilon$ to the joint training objective and hence $\gamma\varepsilon \# \hat{\mathcal{Q}}_\varepsilon < \sum_{k \in \mathcal{K}_\mathcal{O} \cup \{N\}} \gamma \mathcal{L}(\hat{\mathbf{y}}_k)$. With this we obtain a bound on the cardinality of $\hat{\mathcal{Q}}_\varepsilon$

$$\gamma\varepsilon \# \hat{\mathcal{Q}}_\varepsilon < \sum_{k \in \mathcal{K}_\mathcal{O} \cup \{N\}} \gamma \mathcal{L}(\hat{\mathbf{y}}_k) \leq \bar{J}, \tag{8}$$

and hence $\# \hat{\mathcal{Q}}_\varepsilon \leq \kappa(\varepsilon) \doteq \frac{\bar{J}}{\gamma\varepsilon} = \frac{1}{\gamma\varepsilon} \left[\frac{\gamma\beta}{1-\rho} + \beta + \delta \right]$. This finishes the proof. □

Next, we consider the influence of our proposed architecture for the loss trajectories of the intermediate outputs (A.4).

Theorem E.2 (Decreasing loss behavior)**.** *Consider a ResNet of depth N be trained with stage cost loss weighted by $\gamma > 0$ and the joint training formulation (4) and the intermediate outputs chosen as (A.4), and let its parameters $\mathbf{w}$ be δ-suboptimal for some fixed $0 \leq \delta < \infty$, i.e., $J_N(\mathbf{w}) \leq \inf_{\mathbf{w}'} J_N(\mathbf{w}') + \delta$, then the loss of the final output is bounded by*

$$\mathcal{L}(\hat{\mathbf{y}}) \leq \mathcal{L}(\hat{\mathbf{y}}_k) + \delta,$$

compared to all intermediate outputs $k \in \mathcal{K}_\mathcal{O}$.

Proof. Consider a ResNet trained by joint training with the δ-suboptimal parameters $\mathbf{w} = [\mathbf{w}_\mathcal{V}, \mathbf{w}_0, ..., \mathbf{w}_{N-1}, \mathbf{w}_\mathcal{O}]$. Let $j = \max \operatorname{argmin}_{k \in \mathcal{K}_\mathcal{O}} \mathcal{L}(\hat{\mathbf{y}}_k)$ be the index of last intermediate output with the smallest loss value.

Next, we construct parameters $\tilde{\mathbf{w}}$ that contradict the δ-suboptimality of the parameters $\mathbf{w}$. Consider the parameters $\tilde{\mathbf{w}}$ which are equal to $\mathbf{w}$, except for the parameters of the residual terms following j, which are chosen as zero $\tilde{\mathbf{w}}_{\mathcal{F},k}$ for $k = j, \dots, N-1$. By Theorem C.1 the zero parameters for the residual blocks leads to an identity mapping for the following intermediate outputs $\tilde{\mathbf{y}}_k = \hat{\mathbf{y}}_j$ for $k \in \{k \in \mathcal{K}_\mathcal{O} | k \geq j\}$ and the output $\tilde{\mathbf{y}} = \hat{\mathbf{y}}_j$. While the intermediate outputs before j are the equal to the original intermediate outputs, $\tilde{\mathbf{y}}_k = \hat{\mathbf{y}}_k$ for $k \in \{k \in \mathcal{K}_\mathcal{O} | k < j\}$. This leads to the joint training objective

$$J_N(\tilde{\mathbf{w}}) = \sum_{k \in \mathcal{K}_\mathcal{O}} \gamma \mathcal{L}(\tilde{\mathbf{y}}_k) + \mathcal{L}(\tilde{\mathbf{y}}) = \sum_{\substack{k \in \mathcal{K}_\mathcal{O} \\ k \leq j}} \gamma \mathcal{L}(\hat{\mathbf{y}}_k) + \sum_{\substack{k \in \mathcal{K}_\mathcal{O} \\ k > j}} \gamma \mathcal{L}(\hat{\mathbf{y}}_j) + \mathcal{L}(\hat{\mathbf{y}}_j).$$

The difference between the joint training objectives of the parameters $\mathbf{w}$ and $\tilde{\mathbf{w}}$ is

$$J_N(\mathbf{w}) - J_N(\tilde{\mathbf{w}}) = \mathcal{L}(\hat{\mathbf{y}}) - \mathcal{L}(\hat{\mathbf{y}}_j) + \sum_{\substack{k \in \mathcal{K}_\mathcal{O} \\ k > j}} \gamma \left(\mathcal{L}(\hat{\mathbf{y}}_k) - \mathcal{L}(\hat{\mathbf{y}}_j) \right).$$

Since $\inf_{\mathbf{w}'} J_N(\mathbf{w}') \leq J_N(\tilde{\mathbf{w}})$, the δ-suboptimality of the parameters $\mathbf{w}$ gives $J_N(\mathbf{w}) - J_N(\tilde{\mathbf{w}}) \leq \inf_{\mathbf{w}'} J_N(\mathbf{w}') + \delta - J_N(\tilde{\mathbf{w}}) \leq \delta$. Since j is the early-exit with the smallest loss, we have $\mathcal{L}(\hat{\mathbf{y}}_k) - \mathcal{L}(\hat{\mathbf{y}}_j) \geq 0$ for all $k \in \mathcal{K}_\mathcal{O}$ and all summands are non-negative such that $\mathcal{L}(\hat{\mathbf{y}}) - \mathcal{L}(\hat{\mathbf{y}}_j) \leq \delta$. Finally, $\mathcal{L}(\hat{\mathbf{y}}_j) \leq \mathcal{L}(\hat{\mathbf{y}}_k)$ for all $k \in \mathcal{K}_\mathcal{O}$ yields $\mathcal{L}(\hat{\mathbf{y}}) \leq \mathcal{L}(\hat{\mathbf{y}}_k) + \delta$. □

From the optimal control point of view, together with the turnpike results from Theorem 3.2, the immediate implication of the previous result is that the leaving arc of the ResNet is bounded by its suboptimality δ. This is in contrast to the optimal trajectories of the example OCP show in Figure 2, where the trajectories towards the end of the horizon depart from the optimal steady state. Fore more details on turnpike and leaving arcs, we refer to Faulwasser and Grüne [2022].

E.1 Exponential Loss Decay

We now consider a specific ResNet architecture for which the exponential decay of the training loss, as stated in Assumption 3.1, can be guaranteed. This ResNet architecture consists of fully connected layers with ReLU activations, $\sigma(x) = \max\{0, x\}$, and is designed to transfer established results by Hardt and Ma [2017] to the intermediate outputs

$$x_0^{(i)} = \mathcal{V}(x^{(i)}, \mathbf{w}_\mathcal{V}) \qquad = Ex^{(i)}, \tag{9a}$$

$$x_{k+1}^{(i)} = f_k(x_k^{(i)}, \mathbf{w}_k) \qquad = x_k^{(i)} + W_k \sigma(A_k x_k^{(i)} + b_k), \qquad k = 0, \dots, N-1, \tag{9b}$$

$$\hat{y}_k^{(i)} = \mathcal{O}_k(x_k^{(i)}, \mathbf{w}_{\mathcal{O},k}) \qquad = T_k x_k^{(i)}, \qquad k = 0, \dots, N. \tag{9c}$$

The ResNet starts with the input layer $\mathcal{V}$, with parameters $\mathbf{w}_\mathcal{V} = [E]$, which embeds the input sample $x^{(i)} \in \mathbb{R}^{d_{\text{in}}}$ into the hidden state $x_0^{(i)} \in \mathbb{R}^d$ by the linear layer $E \in \mathbb{R}^{d \times d_{\text{in}}}$. The hidden state dimension d is constant across all residual blocks, $x_k^{(i)} \in \mathbb{R}^d$ for all $k = 0, \dots, N$. Each residual block f_k consists of two linear layers with parameters A_k, b_k and W_k collected into $\mathbf{w}_k = [A_k, b_k, W_k]$. The intermediate outputs $\mathcal{O}_k$ with parameters $\mathbf{w}_{\mathcal{O},k} = [T_k]$ predict $\hat{y}_k^{(i)} \in \mathbb{R}^C$, $k = 0, \dots, N$, via a linear layers $T_k \in \mathbb{R}^{C \times d}$, and the final prediction is given by the last intermediate output $\hat{y} = \hat{y}_N$.

The realizability results for this ResNet rely on the following assumption on the regularity of the training dataset.

Assumption E.3 (Regular dataset)**.** Consider the training dataset $\mathbb{D} = \left\{\left(x^{(1)}, y^{(1)}\right), ..., \left(x^{(D)}, y^{(D)}\right)\right\}$, then for any two distinct samples $i, j = 1, .., D$ with $i \neq j$, the distance between their features is lower bounded, i.e., $\|x^{(i)} - x^{(j)}\| \geq \rho$ for some $\rho > 0$.

Given this assumption, we now prove a slightly adapted version of the realizability results by Hardt and Ma [2017].

Theorem E.4 (Finite depth reachability for ResNets)**.** *Let the training data $\mathbb{D}$ satisfy Assumption E.3 and consider the ResNet architecture (9). Moreover, for each class $c = 1, \dots, C$, we consider target vectors $\bar{y}_c \in \mathbb{R}^C$ in the output space. Then, there exist a finite hidden state dimension $d < \infty$ and depth $N < \infty$ and parameters $\mathbf{w}$ such that, for every sample $i = 1, ..., D$, the network output $\hat{y}^{(i)}$ matches the target corresponding to its label*

$$\hat{y}_N^{(i)} = \bar{y}_{y^{(i)}}.$$

Furthermore, the first C components of the final hidden states $x_N^{(i)}$ represent a one-hot encoding of the labels

$$[x_N^{(i)}]_j = \begin{cases} 1 & \text{if } y^{(i)} = j \\ 0 & \text{if } y^{(i)} \neq j, \end{cases} \qquad j = 1, \dots, C.$$

Proof. Consider the alternative ResNet with width $\tilde{d}$ and depth $\tilde{N}$, given by

$$\tilde{x}_0^{(i)} = \tilde{\mathcal{V}}(x^{(i)}, \tilde{\mathbf{w}}_\mathcal{V}) \qquad = \tilde{E}x^{(i)}, \tag{10a}$$

$$\tilde{x}_{k+1}^{(i)} = \tilde{f}_k(\tilde{x}_k^{(i)}, \tilde{\mathbf{w}}_k) \qquad = \tilde{x}_k^{(i)} + \tilde{W}_k \sigma(\tilde{A}_k \tilde{x}_k^{(i)} + \tilde{b}_k), \qquad k = 0, \dots, \tilde{N} - 1, \tag{10b}$$

$$\tilde{y}^{(i)} = \tilde{\mathcal{O}}(\tilde{x}_{\tilde{N}}^{(i)}, \tilde{\mathbf{w}}_\mathcal{O}) \qquad = \tilde{W}_{\tilde{N}} \sigma(\tilde{A}_{\tilde{N}} \tilde{x}_{\tilde{N}}^{(i)} + \tilde{b}_{\tilde{N}}), \tag{10c}$$

where the output block projects the hidden state dimension to the number of classes C by $\tilde{A}_{\tilde{N}} \in \mathbb{R}^{C \times \tilde{d}}$, $\tilde{b}_{\tilde{N}} \in \mathbb{R}^C$, and $\tilde{W}_{\tilde{N}} \in \mathbb{R}^{C \times C}$, which we aggregate into $\tilde{\mathbf{w}}_\mathcal{O} = \left[\tilde{A}_{\tilde{N}}, \tilde{b}_{\tilde{N}}, \tilde{W}_{\tilde{N}}\right]$.

By the reachability results of Hardt and Ma [2017, Theorem 3.2], [4] there exists a finite width $\tilde{d}$ and depth $\tilde{N}$ and parameters $\tilde{\mathbf{w}} = \left[\tilde{\mathbf{w}}_\mathcal{V}, \tilde{\mathbf{w}}_0, ..., \tilde{\mathbf{w}}_{\tilde{N}-1}, \tilde{\mathbf{w}}_\mathcal{O}\right]$ such that, for all training samples $i = 1, .., D$, the alternative network outputs a one-hot encoding of the label

$$[\tilde{y}^{(i)}]_j = \begin{cases} 1 & \text{if } y^{(i)} = j \\ 0 & \text{if } y^{(i)} \neq j. \end{cases}. \tag{11}$$

To transfer these reachability results to the architecture (9), we embed the dynamics of the alternative ResNet (10) into (9). In particular, we introduce an additional residual block to account for the output residual block, leading to the total depth $N = \tilde{N} + 1$, and choose the hidden state dimension based on the number of classes and the hidden state dimension of the alternative ResNet $d = 2\tilde{d} + C$. Given the parameters $\tilde{\mathbf{w}}$ from Hardt and Ma [2017, Theorem 3.2], we construct parameters $\mathbf{w}$ for the ResNet (9) such that the first C components of the final state of the ResNet match the output of the alternative ResNet, i.e., $[x_N^{(i)}]_{1:C} = \tilde{y}^{(i)}$ for all samples $i = 1, \dots, D$. The parameters $\mathbf{w}$ for ResNet (9) are

[4] Notice that the published paper erroneously shows a skip connection despite a dimensionality mismatch. In their rebuttal the authors clarified to Reviewer 3 that this skip connection should be omitted; see https://openreview.net/forum?id=ryxB0Rtxx.

1. Input embedding
$$E = \begin{bmatrix} \tilde{E} \\ \mathbf{0}_{d-\tilde{d}\times d_{\text{in}}} \end{bmatrix}. \tag{12}$$

2. First residual blocks $k = 0, ..., N-2$
$$A_k = \begin{bmatrix} \tilde{A}_k & \mathbf{0}_{\tilde{d}\times d-\tilde{d}} \\ \mathbf{0}_{d-\tilde{d}\times\tilde{d}} & \mathbf{0}_{d-\tilde{d}\times d-\tilde{d}} \end{bmatrix} \quad b_k = \begin{bmatrix} \tilde{b}_k \\ \mathbf{0}_{d-\tilde{d}} \end{bmatrix} \quad W_k = \begin{bmatrix} \tilde{W}_k & \mathbf{0}_{\tilde{d}\times d-\tilde{d}} \\ \mathbf{0}_{d-\tilde{d}\times\tilde{d}} & \mathbf{0}_{d-\tilde{d}\times d-\tilde{d}} \end{bmatrix}.$$

3. Last residual block $k = N-1 = \tilde{N}$ from output residual block
$$A_{N-1} = \begin{bmatrix} \tilde{A}_{\tilde{N}} & \mathbf{0}_{C\times d-\tilde{d}} \\ \mathbf{I}_{\tilde{d}\times\tilde{d}} & \mathbf{0}_{\tilde{d}\times d-\tilde{d}} \\ -\mathbf{I}_{\tilde{d}\times\tilde{d}} & \mathbf{0}_{\tilde{d}\times d-\tilde{d}} \end{bmatrix} \quad b_{N-1} = \begin{bmatrix} \tilde{b}_{\tilde{N}} \\ \mathbf{0}_{\tilde{d}} \\ \mathbf{0}_{\tilde{d}} \end{bmatrix} \quad W_{N-1} = \begin{bmatrix} \tilde{W}_{\tilde{N}} & -\mathbf{I}_{C\times\tilde{d}} & \mathbf{I}_{C\times\tilde{d}} \\ \mathbf{0}_{\tilde{d}\times C} & \mathbf{0}_{\tilde{d}\times\tilde{d}} & \mathbf{0}_{\tilde{d}\times\tilde{d}} \\ \mathbf{0}_{\tilde{d}\times C} & \mathbf{0}_{\tilde{d}\times\tilde{d}} & \mathbf{0}_{\tilde{d}\times\tilde{d}} \end{bmatrix},$$
where we denote by $\mathbf{I}_{C\times\tilde{d}} \in \mathbb{R}^{C\times\tilde{d}}$, the rectangular identity matrix defined by $[\mathbf{I}_{C\times\tilde{d}}]_{i,j} = 1$ if $i = j$ and $[\mathbf{I}_{C\times\tilde{d}}]_{i,j} = 0$ otherwise, for $i = 1, \dots, C$ and $j = 1, \dots, \tilde{d}$. In this construction, we use that for the ReLU activation function $x = \sigma(x) - \sigma(-x)$ for all $x \in \mathbb{R}$.

4. Output projector matrix
$$T_N = [\bar{y}_1 \quad ... \quad \bar{y}_C \quad \mathbf{0}_{C\times d-C}].$$

These parameters embed the dynamics of the alternative ResNet (10) into ResNet (9). The resulting hidden states are
$$x_k^{(i)} = \begin{bmatrix} \tilde{x}_k^{(i)} \\ \mathbf{0}_{d-\tilde{d}} \end{bmatrix}$$
for every data sample $i = 1, \dots, D$ and each residual block $k = 0, \dots, N-1$. Moreover, for this terminal state this implies
$$[x_N^{(i)}]_{1,\dots,C} = \tilde{y}^{(i)}.$$
Hence, by (11), the first C components of the state after the last residual block coincide with the output of the alternative ResNet, which is the one-hot encoding of the label
$$[x_N^{(i)}]_j = \begin{cases} 1 & \text{if } j = y^{(i)} \\ 0 & \text{if } j \in \{1, .., C\} \setminus \{y^{(i)}\} \end{cases}, \qquad j = 1, \dots, C.$$
Multiplication with the output projector matrix T_N yields the selected target for the label $\bar{y}_{y^{(i)}}$
$$\hat{y}_N^{(i)} = T_N x_N^{(i)} = \bar{y}_{y^{(i)}}.$$
□

This result shows that any label-dependent targets $\bar{y}_y$ can be reached. Moreover, selecting $\bar{y}_y$ as the output minimizing the loss function guarantees zero loss at finite depth and also ensures exponential loss decay—for instance, for the L_2 loss [Faulwasser et al., 2024]. However, the most commonly used loss, the cross-entropy with one-hot labels, has no finite minimizer; rather, it approaches its infimum at zero only as the network outputs (logits) grow exponentially in absolute value, thereby asymptotically ensuring maximal separation between the classes. We now extend these finite depth reachability results as a starting point to design parameters that ensure exponentially diverging states such that the cross-entropy loss decays exponentially.

Theorem E.5 (Exponential decrease of cross-entropy)**.** *Consider the ResNet architecture (9), and let the training data $\mathbb{D}$ satisfy Assumption E.3. Let $\mathcal{L}$ be the cross-entropy loss. Then, there exists an infinite-depth ResNet with finite width $d < \infty$ and parameters such that the resulting intermediate outputs $\tilde{\mathbf{y}}_k$ satisfy the exponential loss decrease from Assumption 3.1.*

Proof. The cross-entropy loss has no finite minimizer but converges to zero as the network states diverge to infinity [Püttschneider and Faulwasser, 2026, Lemma 2]. To establish exponential loss decay, we construct an infinite-depth ResNets for which the states diverge exponentially. This construction leverages the finite-depth reachability results from Theorem E.4. The finite-depth

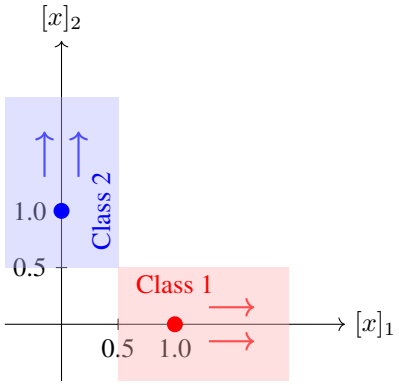


Figure 18: Illustration construction of the infinite-depth ResNet that ensures exponential decay of the cross-entropy loss. First, Theorem E.4 guarantees that all hidden states can be moved to the illustrated configuration in finite depth, where the state of each data point corresponds to the unity vector of its label. Then, the infinite-depth ResNet causes the hidden states to diverge exponentially along the axes of their label thereby ensuring an exponential decay of the cross-entropy loss.

ResNet are extended to an infinite-depth ResNet using a simple choice of parameters that induces the required exponential divergence of the states for cross-entropy loss decay. The construction of the infinite-depth ResNet is illustrated in Figure 18.

Throughout the remainder of this proof, we use the constant output projector matrix $\tilde{T}_k$ for all layers $k = P, ..., \infty$.

$$\tilde{T}_k = \begin{bmatrix} I_{C\times C} & 0_{C\times(d-C)} \end{bmatrix}.$$

With this choice, the network output corresponds to the first C components of the hidden state.

By Theorem E.4, there exists a ResNet with finite width $d < \infty$, depth $P < \infty$, and parameters $\tilde{\mathbf{w}}$, such that all samples are separated according to their label. After these P residual blocks, for every sample $i = 1, \ldots, D$, the hidden state is for $j = 1, \ldots, C$

$$[\tilde{x}_P^{(i)}]_j = \begin{cases} 1 & \text{if } j = y^{(i)} \\ 0 & \text{if } j \neq y^{(i)}. \end{cases}$$

That is, after P residual blocks, the state of datapoint lies on the coordinate axis corresponding to its label with a value of one.

On the finite horizon $k = 0, ..., P$ all intermediate output $\tilde{\mathbf{y}}_k$ are finite, leading to finite loss values $\mathcal{L}(\tilde{\mathbf{y}}_k)$, and hence, there exists $\beta_1 \geq 0$ and $\rho_1 \in [0, 1)$ such that

$$\mathcal{L}(\tilde{\mathbf{y}}_k) \leq \beta_1 \rho_1^k.$$

With the states $\tilde{x}_P^{(i)}$ separated by their labels after P residual blocks, we design simple infinite-depth ResNet parameters that result in an exponential growth of the state and thus prove the exponential decay of the cross-entropy. For $k = P, ..., \infty$ we choose the parameters $\tilde{W}_k = I$, $\tilde{A}_k = 2I$, $\tilde{b}_k = -0.5\mathbf{1}$. This results in the ResNet dynamics

$$\tilde{x}_{k+1}^{(i)} = \tilde{x}_k^{(i)} + \sigma(2\tilde{x}_k^{(i)} - 0.5),$$

for $k \geq P$ and all samples $i = 1, .., D$.

To show the exponential growth of the states, we consider the dynamics component-wise, which for the j-th state component leads to

$$[\tilde{x}_{k+1}^{(i)}]_j = [\tilde{x}_k^{(i)}]_j + \sigma(2[\tilde{x}_k^{(i)}]_j - 0.5),$$

for all residual blocks $k \geq P$ and all samples $i = 1, \ldots, D$. We then show that the state component corresponding to the label growth exponentially, $[\tilde{x}_{k+1}^{(i)}]_j \geq 2[\tilde{x}_k^{(i)}]_j$ for $j = y^{(i)}$, while all other components remain at zero $[\tilde{x}_{k+1}^{(i)}]_j = 0$ for $j \neq y^{(i)}$.

First, for the state dimension corresponding to the label, $j = y^{(i)}$, we have $[\tilde{x}_k^{(i)}]_j \geq 1$, which implies $2[\tilde{x}_k^{(i)}]_j - 0.5 \geq 0$, so that for the ReLU activation $\sigma(2[\tilde{x}_k^{(i)}]_j - 0.5) = 2[\tilde{x}_k^{(i)}]_j - 0.5$. Moreover, since $2[\tilde{x}_k^{(i)}]_j - 0.5 \geq [\tilde{x}_k^{(i)}]_j$, it follows that

$$[\tilde{x}_{k+1}^{(i)}]_j = [\tilde{x}_k^{(i)}]_j + \sigma(2[\tilde{x}_k^{(i)}]_j - 0.5) = [\tilde{x}_k^{(i)}]_j + 2[\tilde{x}_k^{(i)}]_j - 0.5 \geq [\tilde{x}_k^{(i)}]_j + [\tilde{x}_k^{(i)}]_j = 2[\tilde{x}_k^{(i)}]_j.$$

For the other state dimensions, $j \neq y^{(i)}$, we have $[\tilde{x}_k^{(i)}]_j = 0$, so that the ReLU argument is negative, and therefore $\sigma(2[\tilde{x}_k^{(i)}]_j - 0.5) = 0$. So, the off-label state components $j \neq y^{(i)}$ remain at zero

$$[\tilde{x}_{k+1}^{(i)}]_j = [\tilde{x}_k^{(i)}]_j + \sigma(2[\tilde{x}_k^{(i)}]_j - 0.5) = [\tilde{x}_k^{(i)}]_j + 0 = 0.$$

With the constant output projector $\tilde{T}_k = \begin{bmatrix} I_{C\times C} & 0_{C\times(d-C)} \end{bmatrix}$, the output is equal to the C components of the state corresponding to the number of classes

$$\tilde{y}_{k+1}^{(i)} = [\tilde{x}_{k+1}^{(i)}]_{1:C}.$$

This implies that for $k \geq P$, the intermediate outputs satisfy $[\tilde{y}_{k+1}^{(i)}]_j \geq 2[\tilde{y}_k^{(i)}]_j$ and $[\tilde{y}_k^{(i)}]_j \geq 1$ for the component corresponding to the label $j = y^{(i)}$, while keeping the other components at zero, $[\tilde{y}_{k+1}^{(i)}]_j = 0$ for $j \neq y^{(i)}$.

Now we consider the effect on the cross-entropy loss and show that this leads to exponentially decreasing loss, i.e., there exists $\rho_2 \in [0, 1)$ such that $\tilde{\mathcal{L}}([\tilde{y}_{k+1}^{(i)}]_{y^{(i)}}) \leq \rho_2 \tilde{\mathcal{L}}([\tilde{y}_k^{(i)}]_{y^{(i)}})$ for all residual blocks $k = P, ..., \infty$ and all samples $i = 1, ..., D$. First, using that $[\tilde{y}^{(i)}]_j = 0$ for $j \neq y^{(i)}$, we rewrite the cross-entropy loss $\tilde{\mathcal{L}}$ only as a function of the component corresponding to the label $[\tilde{y}^{(i)}]_{y^{(i)}}$

$$\begin{aligned}
\mathcal{L}(\tilde{y}^{(i)}) &= -\log\left(\frac{e^{[\tilde{y}^{(i)}]_{y^{(i)}}}}{\sum_{j=1}^{C} e^{[\tilde{y}^{(i)}]_j}}\right) = \log\left(\sum_{j=1}^{C} e^{[\tilde{y}^{(i)}]_j - [\tilde{y}^{(i)}]_{y^{(i)}}}\right) \\
&= \log\left(1 + \sum_{\substack{j=1 \\ j\neq y^{(i)}}}^{C} e^{[\tilde{y}^{(i)}]_j - [\tilde{y}^{(i)}]_{y^{(i)}}}\right) = \log\left(1 + \sum_{\substack{j=1 \\ j\neq y^{(i)}}}^{C} e^{-[\tilde{y}^{(i)}]_{y^{(i)}}}\right) \\
&= \log\left(1 + (C-1)e^{-[\tilde{y}^{(i)}]_{y^{(i)}}}\right) := \tilde{\mathcal{L}}([\tilde{y}^{(i)}]_{y^{(i)}}).
\end{aligned}$$

The loss $\tilde{\mathcal{L}}([\tilde{y}^{(i)}]_{y^{(i)}})$ is strictly decreasing in the correct class score $[\tilde{y}^{(i)}]_{y^{(i)}}$, since the logarithm is increasing and the negative exponential function decreases with $[\tilde{y}^{(i)}]_{y^{(i)}}$. Since $\tilde{\mathcal{L}}$ is decreasing, the inequality $[\tilde{y}_{k+1}^{(i)}]_{y^{(i)}} \geq 2[\tilde{y}_k^{(i)}]_{y^{(i)}}$, implies that

$$\tilde{\mathcal{L}}([\tilde{y}_{k+1}^{(i)}]_{y^{(i)}}) \leq \tilde{\mathcal{L}}(2[\tilde{y}_k^{(i)}]_{y^{(i)}}).$$

Next we show that there exists $\rho_2 \in [0, 1)$ such that $\tilde{\mathcal{L}}([\tilde{y}_{k+1}^{(i)}]_{y^{(i)}}) \leq \rho_2 \tilde{\mathcal{L}}([\tilde{y}_k^{(i)}]_{y^{(i)}})$ for $k = P, ..., \infty$. To this end, consider the ratio

$$\frac{\tilde{\mathcal{L}}([\tilde{y}_{k+1}^{(i)}]_{y^{(i)}})}{\tilde{\mathcal{L}}([\tilde{y}_k^{(i)}]_{y^{(i)}})} \leq \frac{\tilde{\mathcal{L}}(2[\tilde{y}_k^{(i)}]_{y^{(i)}})}{\tilde{\mathcal{L}}([\tilde{y}_k^{(i)}]_{y^{(i)}})} = \frac{\log\left(1 + (C-1)e^{-2[\tilde{y}_k^{(i)}]_{y^{(i)}}}\right)}{\log\left(1 + (C-1)e^{-[\tilde{y}_k^{(i)}]_{y^{(i)}}}\right)}.$$

The numerator and denominator only differ by the additional factor 2 in the exponent of the numerator, which causes the numerator to decrease faster than the denominator, hence the ratio attains its maximum for the smallest possible $[\tilde{y}_k^{(i)}]_{y^{(i)}}$, which is $[\tilde{y}_k^{(i)}]_{y^{(i)}} = 1$ and hence for all $k = P, ..., \infty$

$$\frac{\tilde{\mathcal{L}}([\tilde{y}_{k+1}^{(i)}]_{y^{(i)}})}{\tilde{\mathcal{L}}([\tilde{y}_k^{(i)}]_{y^{(i)}})} \leq \frac{\log\left(1 + (C-1)e^{-2[\tilde{y}_k^{(i)}]_{y^{(i)}}}\right)}{\log\left(1 + (C-1)e^{-[\tilde{y}_k^{(i)}]_{y^{(i)}}}\right)} \leq \frac{\log\left(1 + (C-1)e^{-2}\right)}{\log\left(1 + (C-1)e^{-1}\right)} = \rho_2 < 1.$$

We obtain an exponential bound on the deeper residual blocks as $\beta_2 = \mathcal{L}(\tilde{y}_P^{(i)})\rho_2^{-P}$, $\mathcal{L}(\tilde{y}_k^{(i)}) \leq \beta_2\rho_2^k$ for $k = P, ..., \infty$ for all data samples $i = 1, ..., D$. Moreover, since the decay holds for all data samples it also hold for the empirical risk

$$\mathcal{L}(\tilde{\mathbf{y}}_k) = \frac{1}{D}\sum_{i=1}^{D}\mathcal{L}(\tilde{y}_k^{(i)}) \leq \frac{1}{D}\sum_{i=1}^{D}\beta_2\rho_2^k = \beta_2\rho_2^k.$$

For the entire depth, $k = 0, ..., \infty$, we obtain an exponentially decaying bound by taking the maximum of the two bounds $\mathcal{L}(\tilde{\mathbf{y}}_k) \leq \beta_1\rho_1^k$ for $k = 0, ..., P$ and $\mathcal{L}(\tilde{\mathbf{y}}_k) \leq \beta_2\rho_2^k$ for $k = P, ..., \infty$. Then with $\beta = \max\{\beta_1, \beta_2\}$ and $\rho = \max\{\rho_1, \rho_2\} < 1$, we get the exponentially decaying bound

$$\mathcal{L}(\tilde{\mathbf{y}}_k) \leq \beta\rho^k,$$

for the entire depth $k = 0, ..., \infty$. Such that the exponential loss decay from Assumption 3.1 is satisfied. □

E.2 Accuracy Turnpike

After showing that turnpikes exist for the cross-entropy loss, we now show turnpikes also occur for other metrics, in this section, we start with the accuracy. We start by showing that an upper bound on the loss provides a lower bound on the accuracy.

Lemma E.6 (Loss bound implies accuracy bound)**.** *Let $\mathbb{D}$ be a dataset of size D, and let $\hat{\mathbf{y}}$ denote the corresponding output. If the average cross-entropy loss satisfies for some number of samples $B \in \mathbb{N}_{[0,D]}$*

$$\mathcal{L}(\hat{\mathbf{y}}) \leq \frac{B+1}{2D}\log(2),$$

then at most B samples are misclassified, and hence the accuracy is lower bounded by $\mathcal{A}(\hat{\mathbf{y}}) \geq 1 - \frac{B}{D}$.

Proof. First, we show that, for a single sample with prediction $\hat{y}$ and label y the upper bound on the loss

$$\mathcal{L}(\hat{y}) < \log(2).$$

ensures correct classification of this sample, i.e., $\operatorname{argmax}_c [\hat{p}(\hat{y})]_c = y$. If $\mathcal{L}(\hat{y}) = -\log([\hat{p}(\hat{y})]_y) < \log(2)$, then $[\hat{p}(\hat{y})]_y > 0.5$. Moreover, since the softmax probabilities are non-negative and sum up to one, it follows that $[\hat{p}(\hat{y})]_c < 0.5$ for any $c \in \{1, ..., C\} \setminus \{y\}$ Therefore $[\hat{p}(\hat{y})]_y > [\hat{p}(\hat{y})]_c$ for any $c \in \{1, ..., C\} \setminus \{y\}$ and therefore the sample is classified correctly.

For the empirical risk on the full dataset, this implies that each misclassified sample must incur a loss larger or equal to $\log(2)$. Hence, if the average cross-entropy satisfies

$$\mathcal{L}(\hat{\mathbf{y}}) = \frac{1}{D}\sum_{i=1}^{D}\mathcal{L}(\hat{y}^{(i)}) < \frac{B+1}{D}\log(2),$$

then at most B samples are misclassified. Moreover, since $\frac{B+1}{2D} < \frac{B+1}{D}$ for all $B \geq 0$ and $D \geq 1$, this holds in particular if $\mathcal{L}(\hat{\mathbf{y}}) \leq \frac{B+1}{2D}\log(2)$. □

With this relation between the loss and the accuracy, we now show that the loss turnpike implies an accuracy turnpike.

Theorem E.7 (Accuracy turnpike)**.** *Suppose that Assumption 3.1 holds, and consider a ResNet of depth N with early-exits attached to set the set of residual blocks $\mathcal{K}_\mathcal{O}$ trained with stage cost loss and weight $0 < \gamma \leq 1$ by the optimal control training (4), and let its parameters $\mathbf{w}$ be δ-suboptimal for some fixed $0 \leq \delta < \infty$, i.e., $J_N(\mathbf{w}) \leq \inf_{\mathbf{w}'} J_N(\mathbf{w}') + \delta$. Then the following hold:*

(i) Then the ResNet exhibits an accuracy turnpike on the training data: for any number of misclassified samples $B \in \mathbb{N}_{[0,D]}$, the number of intermediate outputs which misclassify more than B samples is bounded by some $\kappa < \infty$ independent of the depth of the ResNet N and the early-exit placement $\mathcal{K}_\mathcal{O} \subseteq \mathbb{N}_{[0,N-1]}$.

(ii) For the set of early-exits with accuracy $\mathcal{A}(\hat{\mathbf{y}}_k) < 1 - \frac{B}{D}$ this implies

$$\#\left\{k \in \mathcal{K}_{\mathcal{O}} \cup \{N\} \middle| \mathcal{A}(\hat{\mathbf{y}}_k) < 1 - \frac{B}{D}\right\} \leq \kappa,$$

where $\hat{\mathbf{y}}_k$, $k = 0, ..., N$, denotes the intermediate outputs on the training data.

Proof. By Lemma E.6, $\mathcal{L}(\hat{\mathbf{y}}_k) \leq \frac{B+1}{2D}\log(2)$ implies $\mathcal{A}(\hat{\mathbf{y}}_k) \geq 1 - \frac{B}{D}$ and therefore $\mathcal{A}(\hat{\mathbf{y}}_k) < 1 - \frac{B}{D}$ requires $\mathcal{L}(\hat{\mathbf{y}}_k) > \frac{B+1}{2D}\log(2)$. For the set of early-exits with accuracy less than $1 - \frac{B}{D}$ this implies

$$\#\left\{k \in \mathcal{K}_{\mathcal{O}} \middle| \mathcal{A}(\hat{\mathbf{y}}_k) < 1 - \frac{B}{D}\right\} \leq \#\left\{k \in \mathcal{K}_{\mathcal{O}} \middle| \mathcal{L}(\hat{\mathbf{y}}_k) > \frac{B+1}{2D}\log(2)\right\}.$$

Under Assumption 3.1, by Theorem 3.2 the jointly trained ResNet exhibits a loss turnpike and therefore, there exists $\kappa < \infty$ such that

$$\#\left\{k \in \mathcal{K}_{\mathcal{O}} \middle| \mathcal{L}(\hat{\mathbf{y}}_k) > \frac{B+1}{2D}\log(2)\right\} \leq \kappa,$$

and therefore $\left\{k \in \mathcal{K}_{\mathcal{O}} \middle| \mathcal{A}(\hat{\mathbf{y}}_k) < 1 - \frac{B}{D}\right\} \leq \kappa$. □

E.3 Entropy Turnpike

In this section, we show that the cross-entropy loss turnpike also induces a turnpike in the entropy, a commonly used criterion for early-exiting [Matsubara et al., 2022]. We begin by establishing a relationship between the the cross-entropy loss and the entropy.

Theorem E.8 (Bound on cross-entropy implies bound on entropy)**.** *Let $\mathbb{D}$ be any dataset of size D, and let $\hat{\mathbf{y}}$ denote the predicted output. Suppose that the cross-entropy loss satisfies the bound*

$$\mathcal{L}(\hat{\mathbf{y}}) \leq \bar{\mathcal{L}}$$

for $\bar{\mathcal{L}} > 0$, then the entropy $\mathcal{H}$ is bounded by

$$\mathcal{H}(\hat{\mathbf{y}}) \leq \bar{\mathcal{H}}(D\bar{\mathcal{L}}),$$

where

$$\bar{\mathcal{H}}(\bar{\mathcal{L}}) = \begin{cases} \log(C), & \bar{\mathcal{L}} \geq \log(C), \\ \bar{\mathcal{L}}e^{-\bar{\mathcal{L}}} - \left(1 - e^{-\bar{\mathcal{L}}}\right)\log\left(\frac{1-e^{-\bar{\mathcal{L}}}}{C-1}\right), & \bar{\mathcal{L}} < \log(C). \end{cases} \tag{13}$$

Moreover, the entropy for every sample $i = 1, ..., D$ satisfies

$$\mathcal{H}(\hat{y}^{(i)}) \leq \bar{\mathcal{H}}(D\bar{\mathcal{L}}).$$

Proof. First, we consider a single sample with label y and the cross-entropy and entropy as a function of the probabilities $p \in [0,1]^C$. We show that if the cross-entropy loss satisfies

$$\mathcal{L}_p(p) = -\log([p]_y) \leq \bar{\mathcal{L}}$$

with $\bar{\mathcal{L}} \geq 0$, then there exist a corresponding bound on the entropy $\bar{\mathcal{H}}(\bar{\mathcal{L}})$ such that

$$\mathcal{H}_p(p) = -\sum_{c=1}^{C}[p]_c \log([p]_c) \leq \bar{\mathcal{H}}(\bar{\mathcal{L}}).$$

First, we rewrite the upper bound on the cross-entropy $\mathcal{L}_p(p) = -\log([p]_y) \leq \bar{\mathcal{L}}$ as a lower bound on the softmax probability of the label class y, $[p]_y \geq e^{-\bar{\mathcal{L}}}$. Next, consider the maximization of the entropy on the probability simplex subject to the label probability constraint $[p]_y \geq e^{-\bar{\mathcal{L}}}$

$$\bar{\mathcal{H}}(\bar{\mathcal{L}}) = \max_{p \in \mathbb{R}^C} \quad \mathcal{H}_p(p) = -\sum_{c=1}^{C}[p]_c \log([p]_c) \tag{14a}$$

$$\text{subject to} \quad \sum_{c=1}^{C}[p]_c = 1, \tag{14b}$$

$$[p]_c \geq 0, \qquad c = 1, \ldots, C, \tag{14c}$$

$$[p]_y \geq e^{-\bar{\mathcal{L}}}. \tag{14d}$$

The feasible set of this optimization problem is

$$\mathcal{P} = \left\{ p \in [0,1]^C \,\middle|\, [p]_y \geq e^{-\bar{\mathcal{L}}}, \sum_{c=1}^{C} [p]_c = 1 \right\}.$$

We define a candidate probability vector $p \in \mathcal{P}$

$$[p]_i = \begin{cases} \frac{1}{C} & \text{if } \bar{\mathcal{L}} \geq \log(C), \\ e^{-\bar{\mathcal{L}}} & \text{if } \bar{\mathcal{L}} < \log(C) \text{ and } i = y, \\ \frac{1-e^{-\bar{\mathcal{L}}}}{C-1} & \text{if } \bar{\mathcal{L}} < \log(C) \text{ and } i \neq y. \end{cases}$$

The probability distribution vector p is being majorized by any other feasible probability distribution $q \in \mathcal{P}$. That is, if $[q]_c^{\downarrow}$ and $[p]_c^{\downarrow}$ denote the components of q and p sorted in decreasing order, respectively, then

$$\sum_{c=1}^{j} [q]_c^{\downarrow} \geq \sum_{c=1}^{j} [p]_c^{\downarrow} \quad \text{for all } j = 1, \ldots, C-1,$$

with equality for $j = C$ due to normalization. And since the entropy $\mathcal{H}_p$ is Schur-concave [Marshall et al., 2011] it follows that $\mathcal{H}_p(p) \geq \mathcal{H}_p(q)$ for all $q \in \mathcal{P}$, so p is a maximizer to (14). The resulting maximum value is $\bar{\mathcal{H}}(\bar{\mathcal{L}})$ from (13).

With the softmax probability distribution $\hat{p}(\hat{y})$ of the network output $\hat{y}$ for a single sample, the relationship between the loss bound $\mathcal{L}(\hat{y}) = \mathcal{L}_p\left(\hat{p}(\hat{y})\right) \leq \bar{\mathcal{L}}$ and $\mathcal{H}(\hat{y}) = \mathcal{H}_p\left(\hat{p}(\hat{y})\right) \leq \bar{\mathcal{H}}(\bar{\mathcal{L}})$ directly translates to the loss and entropy as a function of the network output $\hat{y}$.

Next, we prove that an upper bound on the empirical loss

$$\mathcal{L}(\hat{\mathbf{y}}) = \frac{1}{D} \sum_{i=1}^{D} \mathcal{L}(\hat{y}^{(i)}) \leq \bar{\mathcal{L}}$$

implies a corresponding bound on the entropy averaged over the dataset

$$\mathcal{H}(\hat{\mathbf{y}}) = \frac{1}{D} \sum_{i=1}^{D} \mathcal{H}(\hat{y}^{(i)}) \leq \bar{\mathcal{H}}(D\bar{\mathcal{L}}).$$

Since the loss function is non-negative, it follows that for each sample, $i = 1, ..., D$, $\mathcal{L}(\hat{y}^{(i)}) \leq D\bar{\mathcal{L}}$. And thus, the entropy for each sample, $i = 1, ..., D$, is bounded by $\mathcal{H}(\hat{y}^{(i)}) \leq \bar{\mathcal{H}}(D\bar{\mathcal{L}})$. Hence, the averaged entropy suffices

$$\mathcal{H}(\hat{\mathbf{y}}) = \frac{1}{D} \sum_{i=1}^{D} \mathcal{H}(\hat{y}^{(i)}) \leq \frac{1}{D} \sum_{i=1}^{D} \bar{\mathcal{H}}(D\bar{\mathcal{L}}) = \bar{\mathcal{H}}(D\bar{\mathcal{L}}).$$

□

Theorem E.9 (Entropy bound from loss bound)**.** *For every $T > 0$, there exists $\varepsilon > 0$, such that $\mathcal{L}(\hat{\mathbf{y}}) \leq \varepsilon$ implies $\mathcal{H}(\hat{\mathbf{y}}) \leq T$.*

Proof. By Theorem E.8, the entropy satisfies the bound

$$\mathcal{H}(\hat{\mathbf{y}}) \leq \bar{\mathcal{H}}(D\mathcal{L}(\hat{\mathbf{y}}))\,,$$

where $\bar{\mathcal{H}}$ is a continuous, nonnegative function with

$$\lim_{\bar{\mathcal{L}} \to 0} \bar{\mathcal{H}}(D\bar{\mathcal{L}}) = 0.$$

Therefore, for any $T > 0$, continuity of $\bar{\mathcal{H}}$ guarantees the existence of $\varepsilon > 0$ such that $\mathcal{L}(\hat{\mathbf{y}}) \leq \varepsilon$ implies $\bar{\mathcal{H}}(D\mathcal{L}(\hat{\mathbf{y}})) \leq T$. □

Theorem E.10 (Entropy turnpike)**.** *Suppose that Assumption 3.1 holds, and consider a ResNet of depth N with early-exit attached to the set of residual blocks $\mathcal{K}_{\mathcal{O}}$ trained with stage cost loss $0 < \gamma \leq 1$ by the optimal control training (4) and $\mathcal{L}$ as the cross-entropy loss, and let its parameters $\mathbf{w}$ be δ-suboptimal for some fixed $0 \leq \delta < \infty$, i.e., $J_N(\mathbf{w}) \leq \inf_{\mathbf{w}'} J_N(\mathbf{w}') + \delta$. Then, the ResNet exhibits an entropy turnpike on the training data, i.e., for any entropy threshold $T > 0$, there exists a depth bound $\kappa < \infty$ such that*

$$\#\left\{k \in \mathcal{K}_{\mathcal{O}} \cup \{N\} | \mathcal{H}(\hat{\mathbf{y}}_k) > T\right\} \leq \kappa, \tag{15}$$

where $\hat{\mathbf{y}}_k$ denotes the intermediate outputs on the training data.

Proof. By Theorem E.9, for any $T > 0$, there exists $\varepsilon > 0$, such that $\mathcal{L}(\hat{\mathbf{y}}) \leq \varepsilon$ implies $\mathcal{H}(\hat{\mathbf{y}}) \leq T$, hence $\mathcal{H}(\hat{\mathbf{y}}_k) > T$ requires $\mathcal{L}(\hat{\mathbf{y}}_k) > \varepsilon$ and hence

$$\#\left\{k \in \mathcal{K}_{\mathcal{O}} \cup \{N\} | \mathcal{H}(\hat{\mathbf{y}}_k) > T\right\} \leq \#\left\{k \in \mathcal{K}_{\mathcal{O}} \cup \{N\} | \mathcal{L}(\hat{\mathbf{y}}_k) > \varepsilon\right\}.$$

Moreover, under Assumption 3.1 by Theorem 3.2, the ResNet trained with stage cost loss weight $0 < \gamma \leq 1$ exhibits a loss turnpike, i.e., for every $\varepsilon > 0$, there exists $\kappa < \infty$ such that

$$\#\left\{k \in \mathcal{K}_{\mathcal{O}} \cup \{N\} | \mathcal{L}(\hat{\mathbf{y}}_k) > \varepsilon\right\} \leq \kappa,$$

and hence $\#\left\{k \in \mathcal{K}_{\mathcal{O}} \cup \{N\} | \mathcal{H}(\hat{\mathbf{y}}_k) > T\right\} \leq \kappa$. □

E.4 Maximum Softmax Probability Turnpike

In this section, we show that the cross-entropy loss turnpike also induces a turnpike in the maximum softmax probability, another commonly used criterion for early-exiting Matsubara et al. [2022]. We begin by establishing a relationship between the cross-entropy loss and the maximum softmax probability.

Theorem E.11 (Maximum softmax probability bound by loss bound)**.** *For every $\underline{p} \in (0, 1)$, if the cross-entropy loss $\mathcal{L}(\hat{\mathbf{y}})$ for some network output $\hat{\mathbf{y}}$ satisfies*

$$\mathcal{L}(\hat{\mathbf{y}}) \leq \bar{\mathcal{L}} = \frac{1}{D} \min\left\{\log(2), -\log\left(1 - \underline{p}\right)\right\},$$

then for every sample $i = 1, \ldots, D$, the maximum predicted class probability is bounded from below by

$$\max_c \left[\hat{p}(\hat{y}^{(i)})\right]_c \geq 1 - \underline{p},$$

and moreover, the predicted class is the label class, i.e.,

$$\operatorname*{argmax}_c \left[\hat{p}(\hat{y}^{(i)})\right]_c = y^{(i)}.$$

Proof. Suppose that the averaged cross-entropy loss satisfies

$$\mathcal{L}(\hat{\mathbf{y}}) = \frac{1}{D} \sum_{i=1}^{D} \mathcal{L}(\hat{y}^{(i)}) \leq \bar{\mathcal{L}}.$$

Since the cross-entropy is nonnegative for each sample, $\mathcal{L}(\hat{y}^{(i)}) \geq 0$, it follows that

$$\mathcal{L}(\hat{y}^{(i)}) = -\log\left(\left[\hat{p}(\hat{y}^{(i)})\right]_{y^{(i)}}\right) \leq D\bar{\mathcal{L}},$$

and hence

$$\left[\hat{p}(\hat{y}^{(i)})\right]_{y^{(i)}} \geq e^{-D\bar{\mathcal{L}}}.$$

Since $D\bar{\mathcal{L}} \leq \log(2)$, we have that the probability of the label class is lower bounded by $\left[\hat{p}(\hat{y}^{(i)})\right]_{y^{(i)}} \geq \frac{1}{2}$. The softmax probabilities for all other classes are strictly larger than zero, i.e. $\left[\hat{p}(\hat{y}^{(i)})\right]_c > 0$ for all classes $c = 1, ..., C$. Hence, we have that $\operatorname{argmax}_c \left[\hat{p}(\hat{y}^{(i)})\right]_c = y^{(i)}$ and $\left[\hat{p}(\hat{y}^{(i)})\right]_{y^{(i)}} = \max_c \left[\hat{p}(\hat{y}^{(i)})\right]_c$. With $\bar{\mathcal{L}} = \frac{1}{D} \min\left\{\log(2), -\log\left(1 - \underline{p}\right)\right\}$, we have that for each data sample $i = 1, ..., D$

$$\max_c \left[\hat{p}(\hat{y}^{(i)})\right]_c = \left[\hat{p}(\hat{y}^{(i)})\right]_{y^{(i)}} \geq e^{-D\bar{\mathcal{L}}} = \max\left\{\frac{1}{2}, 1 - \underline{p}\right\} \geq 1 - \varepsilon.$$

□

Theorem E.12 (Maximum softmax probability turnpike)**.** *Let Assumption 3.1 hold. Consider the ResNet of depth N with the early-exit placements $\mathcal{K}_{\mathcal{O}}$ trained by (4) with stage cost weight $0 < \gamma \leq 1$ and $\mathcal{L}$ as the cross-entropy loss, and let its parameters $\mathbf{w}$ be δ-suboptimal for some fixed $0 \leq \delta < \infty$, i.e., $J_N(\mathbf{w}) \leq \inf_{\mathbf{w}'} J_N(\mathbf{w}') + \delta$. Then, the ResNet exhibits an max softmax probability on the training data, i.e., for any threshold $\underline{p} \in (0,1)$, there exists a bound $\kappa < \infty$ on the number of early-exits for which a data sample has a softmax probability less than $1 - \underline{p}$, i.e.,*

$$\#\left\{k \in \mathcal{K}_{\mathcal{O}} \cup \{N\} \middle| \exists i = 1, ..., D, \ \max_c \left[\hat{p}(\hat{y}_k^{(i)})\right]_c < 1 - \underline{p}\right\} \leq \kappa. \tag{16}$$

Proof. By Theorem E.11, for any $\underline{p} > 0$, there exists $\varepsilon > 0$, such that $\mathcal{L}(\hat{\mathbf{y}}_k) \leq \varepsilon$ implies $\max_c \left[\hat{p}(\hat{y}_k^{(i)})\right]_c \geq 1 - \underline{p}$ for all samples $i = 1, ..., D$, therefore, if there exists $i = 1, ..., D$, $\max_c \left[\hat{p}(\hat{y}_k^{(i)})\right]_c < 1 - \underline{p}$, then $\mathcal{L}(\hat{\mathbf{y}}_k) > \varepsilon$ and thus and hence

$$\#\left\{k \in \mathcal{K}_{\mathcal{O}} \cup \{N\} \middle| \exists i = 1, ..., D, \ \max_c \left[\hat{p}(\hat{y}_k^{(i)})\right]_c < 1 - \underline{p}\right\} \leq \#\left\{k \in \mathcal{K}_{\mathcal{O}} \cup \{N\} | \mathcal{L}(\hat{\mathbf{y}}_k) > \varepsilon\right\}.$$

Moreover, under Assumption 3.1 by Theorem 3.2, the ResNet trained with stage cost weight $0 < \gamma \leq 1$ exhibits a loss turnpike, i.e., for every $\varepsilon > 0$, there exists $\kappa < \infty$ such that

$$\#\left\{k \in \mathcal{K}_{\mathcal{O}} \cup \{N\} \middle| \exists i = 1, \ldots, D, \ \max_c \left[\hat{p}(\hat{y}_k^{(i)})\right]_c < 1 - \underline{p}\right\} \leq \#\left\{k \in \mathcal{K}_{\mathcal{O}} \cup \{N\} | \mathcal{L}(\hat{\mathbf{y}}_k) > \varepsilon\right\} \leq \kappa,$$

which gives (16). □

E.5 Implication of Turnpikes on Early-Exiting

Next, we analyze the implications of the turnpikes in the entropy and the maximum softmax probabilities on the early-exiting schemes, which use entropy and maximum softmax probability threshold as the early-exit criterion. First we show that under an upper bound on the loss, there exists an entropy threshold that guarantees correct classification.

Theorem E.13 (Entropy threshold for correct classification under loss bound)**.** *Suppose that the loss function satisfies*

$$\mathcal{L}(\hat{\mathbf{y}}) \leq \bar{\mathcal{L}},$$

for $\bar{\mathcal{L}} > 0$, then the early-exiting threshold

$$T(\bar{\mathcal{L}}) = \begin{cases} \infty & \bar{\mathcal{L}} < \frac{1}{D}\log(2) \\ D\bar{\mathcal{L}}e^{-D\bar{\mathcal{L}}} - \left(1 - e^{-D\bar{\mathcal{L}}}\right)\log\left(1 - e^{-D\bar{\mathcal{L}}}\right) & \bar{\mathcal{L}} \geq \frac{1}{D}\log(2), \end{cases} \tag{17}$$

guarantees that if a sample $i = 1, ..., D$ satisfies

$$\mathcal{H}(\hat{y}^{(i)}) < T(\bar{\mathcal{L}}),$$

then this sample is classified correctly

$$\hat{c}(\hat{y}^{(i)}) = y^{(i)}.$$

Proof. The bound

$$\mathcal{L}(\hat{\mathbf{y}}) = \frac{1}{D}\sum_{i=1}^{D} \mathcal{L}(\hat{y}^{(i)}) \leq \bar{\mathcal{L}}$$

implies, that for each sample $i = 1, ..., D$, $\mathcal{L}(\hat{y}^{(i)}) \leq D\bar{\mathcal{L}}$. The cross-entropy upper bound $\mathcal{L}_p(p) = -\log([p]_y) \leq D\bar{\mathcal{L}}$ can be rewritten as a lower bound on the probability of the label class $[p]_y \geq e^{-D\bar{\mathcal{L}}}$. If $\bar{\mathcal{L}} < \frac{1}{D}\log(2)$, then $[p]_y > 0.5$, and thus correct classification is ensured independent of the entropy.

Consequently, we only need to consider the nontrivial case where $\bar{\mathcal{L}} \geq \frac{1}{D}\log(2)$, so that the loss bound alone cannot guarantee correct classification. For this case, we want to find the minimum value of the entropy for which a false classification occurs. For this, we search for the probability distribution vector $p \in [0,1]^C$ with the minimal entropy value and $[p]_y \geq e^{-D\bar{\mathcal{L}}}$, for which a false

classifications occurs, i.e, there exists a class $j \in \mathbb{N}_{[1,C]} \setminus \{y\}$ with $[p]_j \geq [p]_y$. This leads to the optimization problem for the entropy as a function of the probability distribution $p \in [0,1]^C$

$$T(\bar{\mathcal{L}}) = \min_p \mathcal{H}_p(p) \tag{18a}$$

$$\text{s.t.} \quad [p]_y \geq e^{-D\bar{\mathcal{L}}} \tag{18b}$$

$$[p]_j \geq [p]_y \tag{18c}$$

$$[p]_c \geq 0 \qquad c \in \{1, ..., C\} \setminus \{j, y\} \tag{18d}$$

$$\sum_{i=1}^{C} [p]_i = 1. \tag{18e}$$

The feasible set of this optimization problem is

$$\mathcal{P} = \left\{ p \in [0,1]^C \middle| [p]_y \geq e^{-D\bar{\mathcal{L}}},\ [p]_j \geq [p]_y,\ \sum_{c=1}^{C} [p]_c = 1, \right\}.$$

Define the feasible distribution $p \in \mathcal{P}$ on the boundary of $\mathcal{P}$ element-wise as

$$[p]_i = \begin{cases} e^{-D\bar{\mathcal{L}}}, & i = y, \\ 1 - e^{-D\bar{\mathcal{L}}}, & i = j \neq y, \\ 0, & i \notin \{y, j\}. \end{cases}$$

Then p majorizes any other probability distribution $q \in \mathcal{P}$; that is, if $[q]_c^{\downarrow}$ and $[p]_c^{\downarrow}$ denote the components of q and p sorted in decreasing order, respectively, then

$$\sum_{c=1}^{j} [p]_c^{\downarrow} \geq \sum_{c=1}^{j} [q]_c^{\downarrow} \quad \text{for all } j = 1, \ldots, C-1,$$

with equality for $j = C$ due to normalization. And since the entropy $\mathcal{H}_p$ is strictly Schur-concave Marshall et al. [2011] it follows that $\mathcal{H}_p(p) \leq \mathcal{H}_p(q)$ for all $q \in \mathcal{P}$, so p is a minimizer of (18). Moreover, due to the strict Schur-concavity it holds that p is the unique minimizer and $entropy_p(p) < entropy_p(q)$ for $q \in \mathcal{P} \setminus \{p\}$. The corresponding minimal entropy value is (17). Hence, if a probability distribution $p \in [0,1]^C$ satisfies $[p]_y \geq e^{-D\bar{\mathcal{L}}}$ and $\mathcal{H}_p(p) < T(\bar{\mathcal{L}})$, then $[p]_c < [p]_y$ for all $c \in \{1, ..., C\} \setminus \{y\}$.

This gives that if $\mathcal{L}(\hat{\mathbf{y}}) \leq \bar{\mathcal{L}}$ and $\mathcal{H}(\hat{y}^{(i)}) < T(\bar{\mathcal{L}})$, then $\hat{y}^{(i)}$ is classified correctly, i.e. $\hat{c}(\hat{y}^{(i)}) = y^{(i)}$. □

Theorem E.14 (Max softmax probability threshold for correct classification under loss bound)**.** *Suppose that the loss function satisfies*

$$\mathcal{L}(\hat{\mathbf{y}}) \leq \bar{\mathcal{L}},$$

for $\bar{\mathcal{L}} > 0$, then the early-exiting threshold

$$\underline{p}(\bar{\mathcal{L}}) = \begin{cases} 1 & \bar{\mathcal{L}} < \frac{1}{D}\log(2), \\ e^{-D\bar{\mathcal{L}}} & \bar{\mathcal{L}} \geq \frac{1}{D}\log(2), \end{cases} \tag{19}$$

guarantees that if a sample $i = 1, ..., D$ satisfies

$$\max_c \left[\hat{p}(\hat{y}^{(i)})\right]_c > 1 - \underline{p}(\bar{\mathcal{L}}),$$

then that sample is classified correctly

$$\hat{c}(\hat{y}^{(i)}) = y^{(i)}.$$

Proof. The loss bound

$$\mathcal{L}(\hat{\mathbf{y}}) = \frac{1}{D}\sum_{i=1}^{D} \mathcal{L}(\hat{y}^{(i)}) \leq \bar{\mathcal{L}}$$

implies, that for each sample, $i = 1, ..., D$, $\mathcal{L}(\hat{y}^{(i)}) \leq D\bar{\mathcal{L}}$. The cross-entropy upper bound $\mathcal{L}_p(p) = -\log([p]_y) \leq D\bar{\mathcal{L}}$ can be rewritten as a lower bound on the probability of the label class $[p]_y \geq e^{-D\bar{\mathcal{L}}}$. If $\bar{\mathcal{L}} < \frac{1}{D}\log(2)$, then $[p]_y > 0.5$, and thus correct classification is ensured independent of the maximum softmax probability.

Consequently, we only need to consider the nontrivial case where $\bar{\mathcal{L}} \geq \frac{1}{D}\log(2)$, so that the loss bound alone cannot guarantee correct classification. With $[p]_y \geq e^{-D\bar{\mathcal{L}}}$ and since the probabilities sum up to one it follows that for all off label classes $c \neq y^{(i)}$ $[p]_c \leq 1 - e^{-D\bar{\mathcal{L}}}$. Therefore, whenever $\max_c \left[\hat{p}(\hat{y}^{(i)})\right]_c > 1 - \underline{\mathrm{p}}(\bar{\mathcal{L}})$, then $\hat{c}(\hat{y}^{(i)}) = \operatorname{argmax}_c \left[\hat{p}(\hat{y}^{(i)})\right]_c = y^{(i)}$. □

Theorem E.15 (Bounded and correct early-exiting with entropy by turnpikes)**.** *Let Assumption 3.1 hold. And consider the ResNet with depth N and early-exits attached to the set of residual blocks $\mathcal{K}_\mathcal{O}$ trained by (4) with stage cost weight $0 < \gamma \leq 1$ and $\mathcal{L}$ as the cross-entropy loss, and let its parameters $\mathbf{w}$ be δ-suboptimal for some fixed $0 \leq \delta < \infty$, i.e., $J_N(\mathbf{w}) \leq \inf_{\mathbf{w}'} J_N(\mathbf{w}') + \delta$. Then consider the early-exiting scheme with entropy thresholds T, where the propagation of a data sample $x^{(i)}$ is prematurely terminated at RB k with the output $\hat{y}_k^{(i)}$ if $\mathcal{H}(\hat{y}_k^{(i)}) \leq T$. Then, there exists an early-exiting threshold $T > 0$, such that if a sample $i = 1, ..., D$ exits before the final output, i.e., $\mathcal{H}(\hat{y}_k^{(i)}) \leq T$ for $k \in \mathcal{K}_\mathcal{O}$, it is classified correctly*

$$\hat{c}(\hat{y}_k^{(i)}) = y^{(i)}.$$

Moreover, there exists κ such that each sample $i = 1, ..., D$ is exited at latest at the $(\kappa + 1)$-th early-exit

$$\#\left\{k \in \mathcal{K}_\mathcal{O} \cup \{N\} \,|\mathcal{H}(\hat{y}_k^{(i)}) > T\right\} \leq \kappa,$$

independent of the depth N and the early-exit placement. For the early-exit depth in terms of the number of residual blocks this implies that the exit depth is bounded by

$$k^{(i)} = \min\left(\left\{k \in \mathcal{K}_\mathcal{O} \cup \{N\} \,|\mathcal{H}(\hat{y}_k^{(i)}) \leq T\right\} \cup \{N\}\right) \leq \operatorname{seq}\left(\mathcal{K}_\mathcal{O} \cup \{N\}\right)_{\kappa+1}.$$

Proof. We begin by showing that there exists an entropy threshold that guarantees correct classification, which requires deriving an upper bound on the intermediate losses. By Assumption 3.1 and Theorem 3.2, there exists a constant $\bar{J}$ such that, for every depth N and every placement $\mathcal{K}_\mathcal{O}$, the training objective evaluated for the δ-suboptimal parameters $\mathbf{w}$ satisfies

$$J_N(\mathbf{w}) \leq \bar{J}.$$

Hence, we have

$$\sum_{k \in \mathcal{K}_\mathcal{O}} \gamma\mathcal{L}(\hat{\mathbf{y}}_k) + \mathcal{L}(\hat{\mathbf{y}}) \leq \bar{J},$$

where $\hat{\mathbf{y}}_k$ denotes the intermediate outputs and $\hat{\mathbf{y}}$ the final output for the training data. Since $\gamma \leq 1$, we have $\mathcal{L}(\hat{\mathbf{y}}) \geq \gamma\mathcal{L}(\hat{\mathbf{y}})$, which gives the bound

$$\sum_{k \in \mathcal{K}_\mathcal{O} \cup \{N\}} \gamma\mathcal{L}(\hat{\mathbf{y}}_k) \leq \sum_{k \in \mathcal{K}_\mathcal{O}} \gamma\mathcal{L}(\hat{\mathbf{y}}_k) + \mathcal{L}(\hat{\mathbf{y}}) \leq \bar{J}.$$

Moreover, due to the non-negativity of the loss function, each intermediate loss is individually bounded:

$$\mathcal{L}(\hat{\mathbf{y}}_k) \leq \frac{\bar{J}}{\gamma}, \quad k \in \mathcal{K}_\mathcal{O} \cup \{N\},$$

for any ResNet depth N and any early-exit placement $\mathcal{K}_\mathcal{O}$.

By Theorem E.13, if the loss of an intermediate output satisfies $\mathcal{L}(\hat{\mathbf{y}}_k) \leq \frac{\bar{J}}{\gamma}$, then the entropy threshold $T\left(\frac{\bar{J}}{\gamma}\right)$ guarantees correct classification. That is, if $\mathcal{H}(\hat{y}_k^{(i)}) < T\left(\frac{\bar{J}}{\gamma}\right)$ for a sample $i = 1, \ldots, D$, then the predicted class equals the label, $\hat{c}(\hat{y}_k^{(i)}) = y^{(i)}$. Hence, we choose $T = \frac{1}{2}T\left(\frac{\bar{J}}{\gamma}\right)$ and get that if $\mathcal{H}(\hat{y}_k^{(i)}) \leq T$, then the sample is classified correctly.

To relate this sample-wise early-exiting criterion to the average entropy, observe that it is sufficient to ensure

$$\mathcal{H}(\hat{\mathbf{y}}) = \frac{1}{D}\sum_{i=1}^{D}\mathcal{H}(\hat{y}^{(i)}) \leq \frac{1}{D}T.$$

Then, since entropy of each individual sample is non-negative $\mathcal{H}(\hat{y}^{(i)}) \geq 0$ for all $i = 1, \ldots, D$, it follows that for every data sample $i = 1, \ldots, D$ we have $\mathcal{H}(\hat{y}^{(i)}) \leq T$. Moreover, under Assumption 3.1, a ResNet trained by (4) exhibits an entropy turnpike by Theorem E.10. That is, there exists constant $\kappa < \infty$ such that the number of intermediate outputs violating this entropy threshold is bounded

$$\#\left\{k \in \mathcal{K}_{\mathcal{O}} \cup \{N\} \middle| \mathcal{H}(\hat{\mathbf{y}}_k) > \frac{1}{D}T\right\} \leq \kappa,$$

and hence, the early-exit depth for all samples $i = 1, ..., D$ is bounded by the $(\kappa+1)$-th early-exit

$$k^{(i)} = \min\left(\left\{k \in \mathcal{K}_{\mathcal{O}} \cup \{N\} \,|\mathcal{H}(\hat{y}_k^{(i)}) \leq T\right\} \cup \{N\}\right) \leq \operatorname{seq}\left(\mathcal{K}_{\mathcal{O}} \cup \{N\}\right)_{\kappa+1}.$$

□

Theorem E.16 (Bounded and correct early-exiting with max softmax probabilities by turnpikes)**.** *Let Assumption 3.1 hold. And consider the ResNet with depth N trained by (4) with stage cost weight $0 < \gamma \leq 1$ and $\mathcal{L}$ as the cross-entropy loss, and let its parameters $\mathbf{w}$ be δ-suboptimal for some fixed $0 \leq \delta < \infty$, i.e., $J_N(\mathbf{w}) \leq \inf_{\mathbf{w}'} J_N(\mathbf{w}') + \delta$. Then consider the early-exiting scheme with the probability threshold $\underline{p}$, where the propagation of a data sample $x^{(i)}$ is prematurely terminated at RB $k \in \mathcal{K}_{\mathcal{O}}$ with the output $\hat{y}_k^{(i)}$ if $\max_c \left[\hat{p}(\hat{y}_k^{(i)})\right]_c \geq 1 - \underline{p}$. Then, there exists an early-exiting threshold $\underline{p} > 0$ and early-exiting depth κ, such that for every sample $i = 1, ..., D$ the early-exiting depth is bounded independent of the network depth N*

$$\#\left\{k \in \mathcal{K}_{\mathcal{O}} \cup \{N\} \middle| \max_c \left[\hat{p}(\hat{y}_k^{(i)})\right]_c < 1 - \underline{p}\right\} \leq \kappa. \tag{20}$$

Moreover, the early-exit depth in terms of the residual block is bounded by

$$k^{(i)} = \min\left(\left\{k \in \mathcal{K}_{\mathcal{O}} \cup \{N\} \middle| \max_c \left[\hat{p}(\hat{y}_k^{(i)})\right]_c \geq 1 - \underline{p}\right\} \cup \{N\}\right) \leq \operatorname{seq}\left(\mathcal{K}_{\mathcal{O}} \cup \{N\}\right)_{\kappa+1}. \tag{21}$$

and if a sample exits before the final output, i.e., $\max_c \left[\hat{p}(\hat{y}_k^{(i)})\right]_c \geq 1 - \underline{p}$ for $k \in \mathcal{K}_{\mathcal{O}}$, it is classified correctly

$$\hat{c}(\hat{y}_k^{(i)}) = y^{(i)}.$$

Proof. We begin by showing that there exists an max softmax probability threshold that guarantees correct classification, which requires deriving an upper bound on the intermediate losses. By Assumption 3.1 and Theorem 3.2, there exists a constant $\bar{J}$ such that, for every depth N, the training objective evaluated for the δ-suboptimal parameters $\mathbf{w}$ satisfies

$$J_N(\mathbf{w}) \leq \bar{J}.$$

Hence, we have

$$\sum_{k \in \mathcal{K}_{\mathcal{O}}} \gamma\mathcal{L}(\hat{\mathbf{y}}_k) + \mathcal{L}(\hat{\mathbf{y}}) \leq \bar{J},$$

where $\hat{\mathbf{y}}_k$ denotes the intermediate outputs and $\hat{\mathbf{y}} = \hat{\mathbf{y}}_N$ the final output for the training data. Since $\gamma \leq 1$, we have $\mathcal{L}(\hat{\mathbf{y}}) \geq \gamma\mathcal{L}(\hat{\mathbf{y}})$, which gives the bound

$$\sum_{k \in \mathcal{K}_{\mathcal{O}} \cup \{N\}} \gamma\mathcal{L}(\hat{\mathbf{y}}_k) \leq \sum_{k \in \mathcal{K}_{\mathcal{O}}} \gamma\mathcal{L}(\hat{\mathbf{y}}_k) + \mathcal{L}(\hat{\mathbf{y}}) \leq \bar{J}.$$

Moreover, due to the non-negativity of the loss function, each intermediate loss is individually bounded:

$$\mathcal{L}(\hat{\mathbf{y}}_k) \leq \frac{\bar{J}}{\gamma}, \quad k \in \mathcal{K}_{\mathcal{O}} \cup \{N\},$$

for any ResNet depth N and any early-exit placement $\mathcal{K}_{\mathcal{O}}$.

By Theorem E.14, if the loss of an intermediate output satisfies $\mathcal{L}(\hat{\mathbf{y}}_k) \leq \frac{\bar{J}}{\gamma}$, then the probability threshold $\underline{p}\left(\frac{\bar{J}}{\gamma}\right)$ guarantees correct classification. That is, if $\max_c \left[\hat{p}(\hat{y}^{(i)})\right]_c > 1 - \underline{p}\left(\frac{\bar{J}}{\gamma}\right)$ for a sample $i = 1, \ldots, D$, then the predicted class equals the label, $\hat{c}(\hat{y}_k^{(i)}) = y^{(i)}$. Hence, we choose $\underline{p} = \frac{1}{2}\underline{p}\left(\frac{\bar{J}}{\gamma}\right)$ and get that if $\max_c \left[\hat{p}(\hat{y}^{(i)})\right]_c \geq 1 - \underline{p}$, then the sample is classified correctly.

Secondly, under Assumption 3.1 a ResNet trained by (4) exhibits a max softmax probability turnpike by Theorem E.12. That is, there exists a κ, such that for every sample $i = 1, ..., D$ the number of early-exits with softmax probability less than $1 - \underline{p}$ is bounded

$$\# \left\{ k \in \mathcal{K}_{\mathcal{O}} \cup \{N\} \middle| \exists i = 1, ..., D, \ \max_c \left[\hat{p}(\hat{y}_k^{(i)})\right]_c < 1 - \underline{p} \right\} \leq \kappa, \tag{22}$$

independent of the depth N and the early-exit placement $\mathcal{K}_{\mathcal{O}}$. Moreover, under Assumption 3.1, a ResNet trained by (4) exhibits an max softmax probability turnpike by Theorem E.10. That is, since $1 - \underline{p} > 0$, there exists constant $\kappa < \infty$ such that

$$\begin{aligned} k^{(i)} &= \min\left(\left\{ k \in \mathcal{K}_{\mathcal{O}} \cup \{N\} \middle| \max_c \left[\hat{p}(\hat{y}_k^{(i)})\right]_c \geq 1 - \underline{p} \right\} \cup \{N\} \right) \\ &\leq \operatorname{seq}\left(\mathcal{K}_{\mathcal{O}} \cup \{N\}\right)_{\kappa+1} \end{aligned}$$

for all samples $i = 1, ..., D$, and hence the early-exiting depth is bounded by $\operatorname{seq}\left(\mathcal{K}_{\mathcal{O}} \cup \{N\}\right)_{\kappa+1}$. □

F Scope and Limitations

Our findings and theoretical insights contribute to our understanding of early-exit neural networks. Crucially, by connecting practical training concepts to optimal control theory, we are able to introduce the turnpike phenomenon to the deep learning community and utilize the tools to uncover a systematic similarity behavior of the latent dynamics involved in joint training early-exits of ResNets. We want to highlight, that the analysis involved common assumptions about the asymptotic optimality of the data problem leading us to the starting point of our reasoning in Assumption 3.1. As noted in the main text, we were able to prove it for a realistic scenario of cross-entropy loss and fully connected ResNets with ReLU activations. It remains to future work, to extend this loss decay result to other classes of networks, commonly employed for early-exiting, i.e., Transformer backbones and even networks without the residual connection. Even though our experimental suite strongly supports our theoretical findings, we want to highlight the inevitable gap between asymptotic statements for networks trained to global optimality and the finite precision of the stochastic gradient training. We believe that our work contributes greatly to the rigor involved in bridging the gap between optimal control theory and practical deep learning experiments.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The main claims of the abstract are supported: the theoretical contributions are described in Section 3.2 and Section 3.3, with proofs provided in Appendix E. The experimental findings supporting these claims are presented in Section 4 and further supplemented in Appendix A.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: In Appendix F, we discuss the limitations of this work.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: The paper provides the full proofs and the assumptions in Appendix E.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: The paper discloses all information necessary to reproduce the main experimental results. We use standard ResNet architectures as backbone networks and evaluate on standard benchmark datasets, with full details — including hyperparameters, training configuration, and dataset splits — provided in Appendix A.1. The early-exit architecture is described in Section 4.1.

Guidelines:

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

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

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

   Answer: [Yes]

   Justification: The code and instructions to reproduce our results is available at https://github.com/OptCon/early-exit-turnpike. All datasets used in this work are publicly available and are established benchmark datasets in the community.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: All training and test details are specified. We follow standard data splits for all benchmarks. Hyperparameters and optimizer settings follow established conventions for ResNet architectures, with full details in Appendix A.1. The early-exit architecture and its additional hyperparameters are described in Section 4.1 and Appendix A.1, and ablated in Section 4.4 and Appendix A.5.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: Error bars and formal statistical significance tests are not reported, as the primary contribution of this paper is theoretical and our extensive ResNet trainings impose across all configurations and datasets an infeasible computational overhead. Nevertheless, the robustness of our findings is supported by consistent results across multiple datasets (Section A.2) and the ablation of the hyperparameter stage cost weight in Appendix A.5.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: GPU resources used in our experiments are reported in Appendix A.1. Execution times are not reported, as our primary contribution is a theoretical analysis of hidden representations, which is independent of the underlying hardware.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: This research conforms with the NeurIPS Code of Ethics. Our primary contribution is theoretical, and all experiments are conducted on publicly available benchmark datasets, raising no ethical concerns regarding data privacy, fairness, or misuse.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: As our primary contribution is a theoretical analysis of hidden representations in early-exit networks, we do not foresee any direct negative societal impacts. A broader discussion of the societal implications of early-exit networks in general is beyond the scope of this work.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This paper presents a theoretical analysis and experiments on standard benchmark datasets. No new data, pre-trained models, or other high-risk assets are released, so safeguards are not applicable.

    Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: All assets used in this work are properly credited. The benchmark datasets are referenced in Section 4 and Appendix A.1, and the PyTorch version used for implementation is specified in Appendix A.1. All assets are used in accordance with their respective licenses.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

Answer: [Yes]

Justification: The only new asset introduced in this paper is our code, which will be publicly released with the camera-ready version and will include documentation sufficient to reproduce our experimental results.

Guidelines:

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

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

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

Answer: [N/A]

Justification: This paper does not involve crowdsourcing or research with human subjects.

Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing or research with human subjects, and therefore requires no IRB approval.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: No LLMs were used as a component of the core methods in this research. Any use of LLMs was limited to writing and editing assistance, which does not impact the methodology or scientific contributions of the paper.

    Guidelines:

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