# Pruning via Sparsity-indexed ODE: a Continuous Sparsity Viewpoint

Zhanfeng Mo [1] Haosen Shi [1 2] Sinno Jialin Pan [1 3]

## Abstract

Neural pruning, which involves identifying the optimal sparse subnetwork, is a key technique for reducing the complexity and improving the efficiency of deep neural networks. To address the challenge of solving neural pruning at a specific sparsity level directly, we investigate the evolution of optimal subnetworks with continuously increasing sparsity, which can provide insight into how to transform an unpruned dense model into an optimal subnetwork with any desired level of sparsity. In this paper, we proposed a novel pruning framework, coined Sparsity-indexed ODE (SpODE) that provides explicit guidance on how to best preserve model performance while ensuring an infinitesimal increase in model sparsity. On top of this, we develop a pruning algorithm, termed Pruning via Sparsity-indexed ODE (PSO), that enables effective pruning via traveling along the SpODE path. Empirical experiments show that PSO achieves either better or comparable performance compared to state-of-the-art baselines across various pruning settings. Our implementations are now available on GitHub[1].

## 1. Introduction

In recent years, there has been an unprecedented surge in the widespread use of overparameterized neural networks in various complex real-world applications, due to the astounding success of big models in areas such as vision (Radosavovic et al., 2020; Dosovitskiy et al., 2021), language generation (Brown et al., 2020; Devlin et al., 2018), speech recognition (Babu et al., 2021), recommendation (Chen et al., 2019), among others (Janner et al., 2021; Jumper et al., 2021). In the era of big models, model compression methods are essential for the pursuit of acceleration and the deployment of large models on edge devices.

As a highly promising compression paradigm, neural pruning has garnered increasing attention from both academia and industry (Hoefler et al., 2021; Liang et al., 2021). A neural pruner aims to eliminate the majority of the parameters in a dense reference model while maximizing retention of model performance until a specific parameter budget is reached. This pruning procedure is typically followed by a retraining procedure to regain model performance (Han et al., 2016).

Pruned models often suffer from a significant drop in performance. This is due to the fact that identifying the optimal sparse subnetwork is a high-dimensional zero-one programming problem, which is NP-hard and challenging to solve (Papadimitriou & Steiglitz, 1998). Numerous pruning methods (a.k.a pruners) have been developed to address this intractable problem, including score-based pruners, regularization-based pruners, sparse-training pruners, etc. Score-based pruners (Han et al., 2015; Lee et al., 2019; Wang et al., 2020; Rachwan et al., 2022) first evaluate the importance scores of each weight and then wipe out the unimportant parameters with the lowest scores. The importance score function is designed to reflect the performance drop when a typical parameter is pruned. However, these importance scores fail to provide strong guarantees in highly sparse regimes, since they are only valid within the current model's vicinity, and merely represent an upper bound on the potential pruning error. When the target sparsity is extremely high, the importance scores are inclined to provide misleading pruning guidance. An alternative approach is to perform regularization-based pruning (Louizos et al., 2018; Chen et al., 2021; Zhang et al., 2018), which tries to address the hard pruning problem by relaxing the hard sparsity constraints to softer ones. Specifically, the pruner optimizes over the parameter mask variables to minimize a penalized objective, which is essentially a linear combination of the training loss and soft sparsity regularization. However, the performance of regularization-based pruners is limited in practice because the soft sparsity penalty is numerically unstable. To alleviate this problem, sparse-training approaches are proposed in (Zhu & Gupta, 2018; Frankle & Carbin, 2019) to identify a sparse subnetwork

[1] School of Computer Science and Engineering, NTU, Singapore [2] Continental-NTU Corporate Lab, NTU Singapore [3] Department of Computer Science and Engineering, Chinese University of Hong Kong. Correspondence to: Zhanfeng Mo <ZHANFENG001@ntu.edu.sg>.

*Proceedings of the $40^{th}$ International Conference on Machine Learning*, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

[1] https://github.com/mzf666/sparsity-indexed-ode

with comparable performance to the dense model via iterative score-based pruning and retraining. However, such prune-and-retrain iterations introduce significant computational overhead and can be difficult to converge, making it infeasible for large-scale models and datasets.

The limitations of conventional pruners can be attributed to the inherent sparsity and irregularity of the hard neural pruning problem. Given the difficulty of directly solving the hard pruning problem at high sparsity levels, it's natural to wonder whether neural pruning would be easier to solve if an optimal subnetwork with slightly lower sparsity was provided beforehand. Intuitively, one can make small changes to a few parameters while not harming an overparameterized model's optimality (Srinivas & Babu, 2015; Hu et al., 2016). Thus, it is possible to obtain the desired solution by slightly altering the optimal subnetwork of lower sparsity. This motivates us to investigate the evolution of optimal subnetworks as their sparsity increases. If we understand the evolution from an optimal subnetwork of lower sparsity to one with higher sparsity, we are then able to move from the unpruned full model to the desired solution via a path of optimal subnetworks with increasing sparsity. By following the path of optimal subnetworks, we circumvent the intractability of directly solving high-sparsity neural pruning.

**Contributions.** To address the challenge of neural pruning, we propose a novel Sparsity-indexed Ordinary Differential Equation (SpODE) pruning framework that utilizes the evolution of optimal subnetworks with continuously increasing sparsity. At each sparsity level, the SpODE provides explicit guidance on how to best preserve model performance while ensuring an infinitesimal increase in model sparsity. The contributions of this paper are three-fold.

- A Polarized Soft Neural Pruning model is proposed (Section 4.1). It is a more tractable proxy of hard neural pruning that allows for continuous sparsity levels while still maintaining the innate sparsity of neural pruning.
- A novel Sparsity-indexed ODE (SpODE) pruning framework is proposed (Section 4.3). The framework is based on a closed-form sparsity-indexed ODE system, which provides a mathematically tractable way to approximate the evolution of the subnetworks over the continuously increasing sparsity.
- A pruning algorithm, termed Pruning via Sparsity-indexed ODE (PSO), is implemented (Section 4.4), which enables us to travel from an unpruned reference model to an optimal subnetwork of any desired sparsity level by leveraging the SpODE. Empirical experiments demonstrate that PSO achieves either better or parallel performance compared to both structured and unstructured baseline pruning methods across various models and scales of datasets.

## 2. Related Works

**Score-based pruners.** A score-based pruner determines which weights to prune by ranking all the parameters in terms of their importance scores. These scores are designed to reflect the risk of performance degradation when individual weights are removed from the neural network. To this end, many score functions are proposed based on various metrics such as, weight magnitude (Han et al., 2015), connection sensitivity (Lee et al., 2019), synaptic saliency (Tanaka et al., 2020), second order information (Wang et al., 2020; Lubana & Dick, 2021), Neural Tangent Kernel signals (Jacot et al., 2018; Rachwan et al., 2022), etc. Essentially, our PSO can be considered a score-based pruner, where the importance scores are determined by the destination of the SpODE process. On the contrary, conventional scores are primarily based on either standard Taylor expansion arguments or artificial heuristics, which may offer misleading pruning guidance in high sparsity regimes.

**Regularization-based pruners.** Regularization-based pruners tackle the hard neural pruning problem by adding regularization terms to the loss function to encourage sparsity in the model. For example, (Louizos et al., 2018) utilizes a soften $\ell_0$ norm regularizer, (Chen et al., 2021) adopts a mixed $\ell_2/\ell_1$ norm penalty, (Zhang et al., 2018) uses an indicator-like penlaty, (Zhuang et al., 2020) uses a neuron polarization regularizer, etc. However, the optimization process of the regularized problem suffers from numerical instability, and it may not always result in models with the precise level of sparsity that is desired. In contrast, PSO is able to transform the unpruned full mask to an estimated optimal mask for any given level of sparsity via SpODE. It is important to note that our PSO cannot be regarded as a regularization-based pruner, since the dynamic of SpODE cannot be depicted by the gradient flow of any regularized objective function (Proposition 2).

**Sparse-training pruners.** Sparse-training pruners refer to pruning algorithms that perform pruning and training simultaneously (Zhu & Gupta, 2018; Frankle & Carbin, 2019; Mostafa & Wang, 2019; Lin et al., 2020; Xiao et al., 2019). One of the most well-known sparse-training pruners is iterative magnitude pruning (Zhu & Gupta, 2018), which has been widely used as a strong baseline (Zimmer et al., 2022) for various pruning tasks. The Lottery Ticket Hypothesis (LTH) proposed in (Frankle & Carbin, 2019) also shows that a sparse subnetwork with comparable performance to the dense model can be found via a combination of IMP and weight resetting (Frankle & Carbin, 2019; Morcos et al., 2019). Unfortunately, both IMP and LTH pruners are computationally expensive due to their iterative nature and require many training iterations to converge. In practice, our PSO is able to converge at a much lower computational overhead. PSO proves to be a more efficient alternative to

IMP and other iterative pruning methods, as it achieves comparable or better performance in a one-shot manner while reducing computational overhead.

## 3. Preliminary

Suppose a deep neural network of interest is parameterized by $\boldsymbol{\theta} \in \boldsymbol{\Theta} \subset \mathbb{R}^d$, where $\boldsymbol{\Theta}$ is the parameter space and $d$ is the number of prunable model parameters. $\mathbf{v}[i]$ denotes the $i$-th entry of vector $\mathbf{v} \in \mathbb{R}^d$. The zero norm $\|\cdot\|_0$ of $\mathbf{v}$ is defined as the number of nonzero entries, and the $\ell_p$ norm is $\|\mathbf{v}\|_p \triangleq (\sum_i |\mathbf{v}[i]|^p)^{1/p}$. The standard Hadamard (element-wise) product of $\mathbf{v}$ and $\mathbf{w}$ is denoted as $\mathbf{v}\odot\mathbf{w}$. $\mathbb{I}\{\cdot\}$ denotes the entry-wise indicator function. $\operatorname{top}_k(\mathbf{v})[i] \triangleq \mathbb{I}\{|\mathbf{v}[i]| \text{ is among the top-}k \text{ in magnitude}\}$. Without additional specification, we define $\mathbf{1} \in \mathbb{R}^d$ as the vector with all entries equaling 1.

### 3.1. Neurwal Pruning

The goal of neural pruning is to determine the optimal sparse neural network among $\boldsymbol{\Theta}$ with at most $d'$ non-zero parameters, where $d'$ (usually, $d' \ll d$) is called the target budget, and $1 - d'/d$ is the so-called target sparsity. To achieve this, pruning algorithms focus on compressing a dense reference model $\boldsymbol{\theta}^* \in \boldsymbol{\Theta}$ by wiping out less important parameters, while trying to maximally retain model performance. The resultant pruned model is then retrained for a further performance boost (Han et al., 2016). In general, neural pruning can be interpreted as the following constrained energy preservation problem.

**Definition 1** (Neural Pruning)**.** Given a reference model $\boldsymbol{\theta}^* \in \mathbb{R}^d$, target parameter budget $d' \leqslant d$ and an enery function $\mathcal{E}(\cdot) : \mathbb{R}^d \mapsto \mathbb{R}_+$, the neural pruning problem is defined as

$$\min_{\mathbf{m}\in\{0,1\}^d} \mathcal{E}(\mathbf{m} \odot \boldsymbol{\theta}^*), \text{ s.t. } \|\mathbf{m}\|_0 = d', \tag{1}$$

where $\mathbf{m}$ is the zero-one hard parameter mask $\mathbf{m}[i] \triangleq \mathbb{I}\{\boldsymbol{\theta}^*[i] \text{ is preserved}\}$.

Intuitively, neural pruning can severely impair the regularity of the reference model and leads to an explosion of some sort of model energy, such as evaluation loss, classification error, and model capacity penalties. Thus, neural pruning aims to cancel a given amount of parameters with minimal energy explosion. As shown in Table 7, by specifying different energy functions $\mathcal{E}(\cdot)$, neural pruning methods with various objectives can be regarded as instantiations of (1).

### 3.2. A Tour via Optimal Masks of Increasing Sparsity

Due to the intrinsic sparsity and irregularity of optimal masks, solving (1) of a given target sparsity from scratch is challenging. A question naturally arises is that: would it be much easier to find the optimal mask of sparsity $1 - d'/d$, if the optimal mask of sparsity $1 - (d' + 1)/d$ is known in advance? For simplicity, we denote by $\mathbf{m}_t$ a solution to neural pruning of sparsity $t$. In practice, one can alter a few parameters of an overparameterized neural network without significantly changing its functionality. Thus, if the minimal energy is attained at $\mathbf{m}_{1-(d'+1)/d}$ among all $1 - (d' + 1)/d$-sparsity masks, we should be able to increase its sparsity to $1 - d'/d$ by carefully altering a few parameters, while retaining its optimality in energy preservation. In this case, the denser solution $\mathbf{m}_{1-(d'+1)/d}$ provides a strong inductive bias that, we should be able to find $\mathbf{m}_{1-d'/d}$ in the vicinity of $\mathbf{m}_{1-(d'+1)/d}$.

This Gedanken experiment motivates us to study how the optimal mask $\mathbf{m}_t$ evolves as the sparsity $t$ continuously increases from $0$ to $1 - d'/d$. Once the evolution of $(\mathbf{m}_t)_{t\in[0,1-d'/d]}$ is known, we are able to travel from the unpruned mask $\mathbf{1}$ to $\mathbf{m}_{1-d'/d}$ by way of optimal masks of each sparsity level, without solving the original neural pruning problem brutally. Suppose the displacement from $\mathbf{m}_t$ to $\mathbf{m}_{t+\Delta t}$ is a function of $\mathbf{m}_t$, e.g.

$$\mathbf{m}_{t+\Delta t} - \mathbf{m}_t = F(\mathbf{m}_t)\Delta t, \tag{2}$$

by taking an unrigorous limitation $\Delta t \to 0$, one can show that the evolution of $\mathbf{m}_t$ follows a sparsity-indexed ODE

$$\mathrm{d}\mathbf{m}_t = F(\mathbf{m}_t)\mathrm{d}t, \; t \in [0, 1 - d'/d].$$

Ideally, the sparsity-indexed ODE starting from the unpruned mask $\mathbf{m}_0 \triangleq \mathbf{1}$ eventually arrives a desirable solution $\mathbf{m}_{1-d'/d}$ as the sparsity $t$ varies from 0 to $1 - d'/d$.

Before we can formally establish such a sparsity-indexed ODE framework and develop the associated pruning algorithm, two technical issues remain to be fixed. Firstly, due to the discreteness of the zero norm, the change of sparsity $\Delta t$ is at least $1/d$ and it prevents us from simply taking the limitation $\Delta t \to 0$. Secondly, for any sparsity level $t$, the minimal energy can be attained at multiple optimal masks $\mathbf{m}_t$. Thus, we should avoid ambiguity when determining $\mathbf{m}_{t+\Delta t}$, i.e. the successive destination of $\mathbf{m}_t$.

## 4. Pruning via Sparsity-indexed ODE

In this section, we propose a novel pruning framework, coined Sparsity-indexed ODE (SpODE). We first introduce Polarized Soft Neural Pruning (Definition 2), a soft relaxation of the original neural pruning that is more amenable. For the polarized soft neural pruning, we show that SpODE (Proposition 3) depicts how the optimal masks of sparsity $t$ evolve as $t$ increases continuously from 0 to target sparsity $1 - d'/d$. On top of that, we propose the Pruning via Sparsity-indexed ODE (PSO) algorithm (Algorithm 1), which performs neural pruning by traveling from the unpruned mask toward an optimal mask via the SpODE.

### 4.1. Polarized Soft Neural Pruning

As illustrated in Section 3.2, in order to study the dynamic of optimal masks with respect to continuously increasing sparsity, we need to relax the hard sparsity constraint to a softer one, thereby we can extend the definition of sparsity to the whole $[0,1]$ interval. Meanwhile, we must avoid using a dense mask when evaluating energy function, which may distort the inherent sparsity of the hard neural pruning problem. This requires us to polarize the dense mask variables to be nearly binary during the energy evaluation. To this end, we propose the following Polarized Soft Neural Pruning model.

**Definition 2** (Polarized Soft Neural Pruning)**.** Given a reference model $\boldsymbol{\theta}^* \in \boldsymbol{\Theta}$, a target sparsity $t \in [0,1]$, a smooth energy function $\mathcal{E}(\cdot) : \mathbb{R}^d \mapsto \mathbb{R}_+$ and a smooth soft sparsity function $G(\cdot) : \mathbb{R}^d \mapsto [0,1]$ satisfying $G(\cdot)|_{\{0,1\}^d} = 1 - \|\cdot\|_0/d$ (e.g., $1 - \|\cdot\|_p^p/d$), for any $\varepsilon > 0$, we define $I_\varepsilon \triangleq ([0,\varepsilon] \cup [1-\varepsilon, 1])^d$, $\Gamma_t \triangleq \{\mathbf{m} : G(\mathbf{m}) = t\}$, then the Polarized Soft Neural Pruning model is defined as

$$\min_{\mathbf{m}\in\mathbb{R}^d} \mathcal{E}_\varepsilon(\mathbf{m}) \triangleq \mathcal{E}(\mathcal{P}_\varepsilon(\mathbf{m}) \odot \boldsymbol{\theta}^*), \ \ \text{s.t.} G(\mathbf{m}) = t, \tag{3}$$

where $\mathcal{P}_\varepsilon(\cdot) : \mathbb{R}^d \mapsto I_\varepsilon \cap \Gamma_t$ is the Mask Polarizer, which is defined as the projection[2] to $I_\varepsilon \cap \Gamma_t$ (w.r.t norm $\|\cdot\|$), i.e.

$$\mathcal{P}_\varepsilon(\mathbf{m}) \triangleq \arg \min_{\mathbf{m}' \in I_\varepsilon \cap \Gamma_t} \|\mathbf{m}' - \mathbf{m}\|. \tag{4}$$

The soften sparsity metric $G(\cdot)$ is a generalization of the conventional zero-norm sparsity, and has the same values as $\|\cdot\|_0$ on the $\{0,1\}^d$ lattice. The term 'sparsity of $\mathbf{m}$' in the rest of the paper refers to $G(\mathbf{m})$ without further specification. Before the energy function is evaluated, the soft mask $\mathbf{m}$ is projected to be a nearly binary mask in $I_\varepsilon$. For a sufficiently small $\varepsilon$, each entry of polarized mask $\mathcal{P}_\varepsilon(\mathbf{m})$ resides tightly around either 0 or 1. When $\varepsilon \to 0$, $I_\varepsilon$ degenerates to $\{0,1\}^d$, and the hard neural pruning is recovered by the polarized soft pruning model. In this way, (3) provides a desirable soft approximation of (1) without losing the innate sparsity of zero-one masks. From now on, we only focus on the evolution of optimal masks of (3) and use $\mathbf{m}_t$ to represent an optimal mask with sparsity $t$.

### 4.2. Greedy Path of Optimal Masks

Recall that our ultimate goal is to study how the optimal masks change under an infinitesimal increase in sparsity. To achieve this, we should first understand the one-step evolution from a known $\mathbf{m}_t$ to $\mathbf{m}_{t+\Delta t}$ for a small $\Delta t$. As mentioned in Section 3.2, since the minimal energy is usually attained at more than one $t$-sparsity mask, there are multiple choices of the next station of $\mathbf{m}_t$. However, these candidate optimal masks are not equally reachable for $\mathbf{m}_t$: a candidate that is distant from $\mathbf{m}_t$ may have a drastically different sparse pattern, and thus cannot be derived from $\mathbf{m}_t$ via a slight alteration; on the contrary, a candidate nearby tends to share a similar sparse pattern with $\mathbf{m}_t$. Thus, it is more preferable to greedily select the nearest [3] $\mathbf{m}_{t+\Delta t}$ (w.r.t to $\|\cdot\|$) to be the successive state of $\mathbf{m}_t$. The following proposition demonstrates that such a greedy selection scheme yields a greedy path of optimal masks.

**Proposition 1** (Greedy Path of Optimal Masks (informal version of Proposition 5))**.** *Suppose the aforementioned greedy selection scheme induces a series of optimal masks with gradually increase sparsity* $\{\mathbf{m}_{k\Delta t}\}_{0\leqslant k\leqslant[1/\Delta t]}$. *Under some regularity conditions, when the resolution* $\Delta t \to 0$, *the series tends to a densely indexed process* $(\mathbf{m}_t^*)_{t\in[0,1]}$, *termed the Greedy Path of Optimal Masks.*

The greedy path $t \mapsto \mathbf{m}_t^*$ is indeed a function that maps any $t$ to an optimal mask of that sparsity level $t$. Besides, the construction of $(\mathbf{m}_t^*)_{t\in[0,1]}$ indicates that the greedy path guides us to travel from $\mathbf{m}_t^*$ to $\mathbf{m}_{t+\mathrm{d}t}^*$ with minimal mask alteration.

### 4.3. Sparsity-indexed ODE

In order to design neural pruning algorithms by leveraging $(\mathbf{m}_t^*)_{t\in[0,1]}$, we still need to figure out the displacement from sparsity $t$ to $t + \Delta t$, i.e. the term $F(\mathbf{m}_t)\Delta t$ in (2). This motivates us to establish an estimation of $\mathbf{m}_{t+\Delta t}$ based on $\mathbf{m}_t$ via a localization trick: instead of solving (3) directly, we turn to a localized neural pruning problem with linearized objective and constraint

$$\widehat{\mathcal{E}}_\varepsilon(\mathbf{m}) \triangleq \mathcal{E}_\varepsilon(\mathbf{m}_t) + \nabla\mathcal{E}_\varepsilon(\mathbf{m}_t)^\top(\mathbf{m} - \mathbf{m}_t), \tag{5}$$

$$\widehat{G}(\mathbf{m}) \triangleq G(\mathbf{m}_t) + \nabla G(\mathbf{m}_t)^\top(\mathbf{m} - \mathbf{m}_t). \tag{6}$$

Hence, we are able to solve the optimal one-step displacement $\mathbf{m}_{t+\Delta t} - \mathbf{m}_t$ from this localized optimization problem.

**Definition 3** (Localized One-step Evolution)**.** Following the notations in Definition 2, suppose $\mathcal{E}_\varepsilon$ and $G$ are differentiable at $\mathbf{m}_t$, the Localized One-step Evolution problem is defined as

$$\begin{aligned} &\min_{\boldsymbol{\delta}\in\mathbb{R}^d} \nabla\mathcal{E}_\varepsilon(\mathbf{m}_t)^\top\boldsymbol{\delta}, \\ &\text{s.t. } \nabla G(\mathbf{m}_t)^\top\boldsymbol{\delta} = \Delta t, \ \|\boldsymbol{\delta}\| \leqslant r_t\Delta t, \end{aligned} \tag{7}$$

where $r_t > 0$ is a localized radius hyparameter and needs to satisfy $r_t > 1/\|\nabla G(\mathbf{m}_t)\|$ to ensure the feasibility of (7).

[2] Since $I_\varepsilon$ is non-convex, to avoid ambiguity, we should slightly revise the definition of the polarizer to ensure $\mathcal{P}_\varepsilon(\cdot)$ to return exactly one polarized mask. To achieve this, we can define a total order '$\preceq$' over where the minimum of $\|\cdot - \mathbf{m}\|$ is attained, such that $\mathcal{P}_\varepsilon(\cdot)$ can return the $\preceq$-minimum element as the desirable projection. For notation simplicity, we postpone the detailed definition of $\mathcal{P}_\varepsilon$ to Appendix B, and we can feel free to treat $\mathcal{P}_\varepsilon$ as a common projection.

[3] If there exist multiple 'nearest' candidates, we can solve such a candidate collision issue by selecting the minimum element w.r.t a total order.

We define $\boldsymbol{\delta}_t$, the solution to (7), as the Optimal Localized One-step Displacement.

*Remark* 1. As illustrated in Figure 1, $\boldsymbol{\delta}_t$ is the best displacement that ensures a $\Delta t$ increase in sparsity with minimal energy explosion. The localized one-step evolution provides us a local estimation of $\mathbf{m}_{t+\Delta t}$ based on $\mathbf{m}_t$, which allows us to travel from a known optimal mask to a $\Delta t$-sparser one.

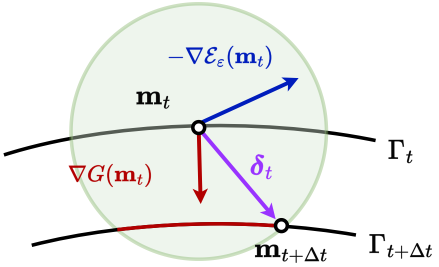
$-\nabla\mathcal{E}_\varepsilon(\mathbf{m}_t)$
$\mathbf{m}_t$
$\boldsymbol{\Gamma}_t$
$\nabla G(\mathbf{m}_t)$
$\boldsymbol{\delta}_t$
$\mathbf{m}_{t+\Delta t}$
$\boldsymbol{\Gamma}_{t+\Delta t}$

*Figure 1.* A geometric illustration of the optimal localized one-step displacement. The green ball represents the localization region.

For a sufficiently small $\Delta t$, the linearized objective and constraint serve as good proxies of the original ones. Hence, we are able to estimate the oracle one-step evolution by solving the more amenable (7).

**Proposition 2** (Optimal Localized One-step Displacement)**.** *Following the conditions in Definition 3, the solution to* (7) *admits a closed-form solution* $\boldsymbol{\delta}_t = F(\mathbf{m}_t)\Delta t$*, with*

$$F(\mathbf{m}) \triangleq \begin{cases} \mathbf{g}/\|\mathbf{g}\|^2, \text{ if } \|\mathbf{g}\|\|\mathbf{e}\| = |\mathbf{g}^\top\mathbf{e}|^2, \\ x\mathbf{e} + y\mathbf{g}, \text{ else,} \end{cases} \tag{8}$$

$$x \triangleq \sqrt{((r^2-1)/((\|\mathbf{g}\|\|\mathbf{e}\|)^2 - (\mathbf{g}^\top\mathbf{e})^2))},$$
$$y \triangleq (1 - \mathbf{g}^\top\mathbf{e}x)/\|\mathbf{g}\|^2,$$

*where* $\mathbf{e} \triangleq -\nabla\mathcal{E}_\varepsilon(\mathbf{m})$*,* $\mathbf{g} \triangleq \nabla G(\mathbf{m})$ *and* $r \triangleq r_t\|\mathbf{g}\| > 1$.

*Remark* 2. When the descending direction of energy and the ascending direction of sparsity reach a consensus, the $\boldsymbol{\delta}_t$ follows the greedy sparsity descent direction. Otherwise, $\boldsymbol{\delta}_t$ becomes a weighted sum of $-\nabla\mathcal{E}_\varepsilon(\mathbf{m}_t)$ and $\nabla G(\mathbf{m}_t)$, which attains the minimal energy within the local regime while ensuring a $\Delta t$-sparsity ascent. We also need to emphasize that, $\boldsymbol{\delta}_t$ is not a linear combination in terms of $-\nabla\mathcal{E}_\varepsilon(\mathbf{m}_t)$ and $\nabla G(\mathbf{m}_t)$. Thus, such $\boldsymbol{\delta}_t$ can not be obtained by minimizing a regularized objective $\mathcal{E}_\varepsilon + \lambda R(G, d')$ with a standard gradient descent step.

So far, we have established an explicit expression of the optimal local one-step displacement, which fills the blank in the aforementioned argument (2). By carefully taking $\Delta t \to 0$, we are now able to establish the Sparsity-indexed ODE, which sheds light on the dynamic of $(\mathbf{m}_t^*)_{t\in[0,1]}$.

**Proposition 3** (Sparsity-indexed ODE (informal version of Proposition 6))**.** *Following the notations of Proposition 2, under some regularity conditions on* $\mathcal{E}_\varepsilon$*, by carefully taking* $\Delta t \to 0$*, the series* $\{\widetilde{\mathbf{m}}_{k\Delta t}\}$ *constructed by* $\widetilde{\mathbf{m}}_0 \triangleq \mathbf{1}$ *and* $\widetilde{\mathbf{m}}_{t+\Delta t} \triangleq \widetilde{\mathbf{m}}_t + \boldsymbol{\delta}_t$ *converges to a Sparsity-indexed ODE (SpODE), which is given by*

$$\mathrm{d}\widetilde{\mathbf{m}}_t = F(\widetilde{\mathbf{m}}_t)\mathrm{d}t,\ t \in [0,1],\ \text{and}\ \widetilde{\mathbf{m}}_0 = \mathbf{1}, \tag{9}$$

*where* $F(\cdot)$ *is defined in* (8).

The SpODE permits a piecewise smooth transition from $\widetilde{\mathbf{m}}_t$ to $\widetilde{\mathbf{m}}_{t+\mathrm{d}t}$ with minimal energy explosion. By running the SpODE from $t = 0$ to $t = 1 - d'/d$, we are able to travel from the unpruned dense mask $\widetilde{\mathbf{m}}_0$ to $\widetilde{\mathbf{m}}_{1-d'/d}$, which is a desirable approximation of the oracle $\mathbf{m}^*_{1-d'/d}$.

**Proposition 4** (SpODE Travels via Optimal Masks (informal version of Proposition 7))**.** *Let* $(\mathbf{m}_t^*)_{t\in[0,1]}$ *be the greedy path defined in Proposition 1, and* $(\widetilde{\mathbf{m}}_t)_{t\in[0,1-d'/d]}$ *follows the SpODE with a well-designed localization scheme. Then it holds that* $\widetilde{\mathbf{m}}_t = \mathbf{m}_t^*$*,* $\forall\, t \in [0, 1-d'/d]$.

### 4.4. Pruning via Sparsity-indexed ODE

Inspired by the SpODE framework, we propose a novel pruning algorithm, called Pruning via Sparsity-indexed ODE (PSO). For a given target parameter budget $d' < d$, we run the discretized SpODE from $t = 1$ to $t = 1 - d'/d$ to obtain $\widetilde{\mathbf{m}}_{1-d'/d}$. Then, we polarize $\widetilde{\mathbf{m}}_{1-d'/d}$ to a nearly sparse mask $\mathcal{P}_\varepsilon(\widetilde{\mathbf{m}}_{1-d'/d})$, and cancel out $d - d'$ parameters with the smallest mask values.

**Algorithm 1** Pruning via Sparsity-indexed ODE (PSO)

```
Input: reference model $\boldsymbol{\theta}^*$, target parameter budget $d'$, empirical mask polarizer $\widehat{\mathcal{P}}_{\varepsilon,\alpha}(\cdot)$, localization shceme $r_t$, SpODE discretization steps number $N$.
Output: a hard mask pruned by SpODE $\widehat{\mathbf{m}} \in \{0,1\}^d$.
Initialization $t \leftarrow 0$, $\widetilde{\mathbf{m}}_t \leftarrow \mathbf{1}$, $\Delta t \leftarrow (1 - d'/d)/N$
for $i = 1$ to $N$ do
  $\widetilde{\mathbf{m}}_{t+\Delta t} \leftarrow \widetilde{\mathbf{m}}_t + F(\widetilde{\mathbf{m}}_t)\Delta t$ {SpODE discretization}
  $t \leftarrow t + \Delta t$
end for
$\widehat{\mathbf{m}} \leftarrow \mathrm{top}_{d'}(\widehat{\mathcal{P}}_{\varepsilon,\alpha}(\widetilde{\mathbf{m}}_{d'/d}))$ {Mask Polarization}
return $\widehat{\mathbf{m}}$
```

As one shall see, PSO merely depends on the parameterization of energy and soft sparsity functions. Thus, it can be applied to either structured or unstructured pruning for any model structures, by setting the mask variable $\mathbf{m}$ to be either the node masks, channel masks, or weight masks. A more detailed version of PSO is provided in Algorithm 2.

Besides the parameterization of $\mathbf{m}$, the implementation of the polarizer $\mathcal{P}_\varepsilon$ is also crucial to pruning performance and numerical stability of PSO. For any soft mask $\mathbf{m}$, the polarizer $\mathcal{P}_\varepsilon(\cdot)$ aims to find a nearly binary proxy in $I_\varepsilon$ with the same sparsity as $\mathbf{m}$. To circumvent the difficulty of computing the projection to $I_\varepsilon \cap \Gamma_t$, we employ an empirical

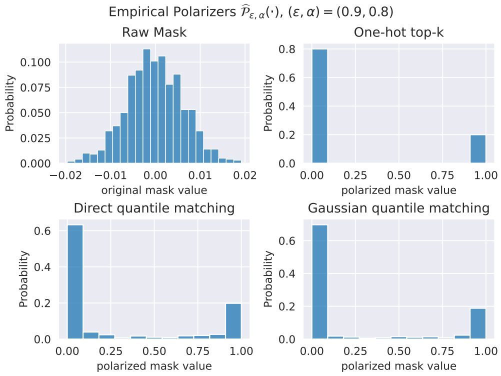

*Figure 2.* Visualization of the implementations of 3 empirical polarizers: one-hot polarizer, quantile polarizer, and Gaussian polarizer.

polarizer as a more amenable proxy of $\mathcal{P}_\varepsilon$. Specifically, an empirical polarizer $\widehat{\mathcal{P}}_{\varepsilon,\alpha}(\cdot)$ is expected to satisfy: 1) at least $1-\alpha$ of the entries of $\widehat{\mathcal{P}}_{\varepsilon,\alpha}(\mathbf{m})$ falls in $[0,\varepsilon] \cup [1-\varepsilon, 1]$. 2) $\widehat{\mathcal{P}}_{\varepsilon,\alpha}(\cdot)$ preserves the soft sparsity within a small error, i.e. $|G(\widehat{\mathcal{P}}_{\varepsilon,\alpha}(\mathbf{m})) - G(\mathbf{m})| \leqslant 1/d$. As shown in Figure 2, we implement three types of $\widehat{\mathcal{P}}_{\varepsilon,\alpha}$ that empirically works well with neural pruning. Roughly speaking, such $\widehat{\mathcal{P}}_{\varepsilon,\alpha}$ transforms each entry of $\mathbf{m}$ to either $[0,\varepsilon]$ or $[1-\varepsilon, 1]$ by matching the distribution of $|\mathbf{m}|$ to a sigmoid distribution or using hard threshold, without changing the order of $|\mathbf{m}|$. The definitions of $\widehat{\mathcal{P}}_{\varepsilon,\alpha}$ are detailed in the Appendix A.3.

## 5. Experiments

In this section, we conduct extensive experiments to demonstrate the effectiveness of the proposed PSO (Algorithm 1) algorithm for both unstructured and structured pruning, as well as in both one-shot and iterative pruning scenarios. In general, the evaluation process for each setting is as follows: 1) Compress a pretrained teacher model $\boldsymbol{\theta}^*$ to the target sparsity level using either PSO or other baseline methods; 2) Retrain the pruned sparse model for some epochs to achieve convergence; 3) Evaluate the average top-1 classification accuracy of the associated pruned model in the last several fine-tuning epochs. For fair comparisons, different pruning settings vary only in step (1). More implementation details are elaborated on in the Appendix A.

### 5.1. CIFAR-10 / 100 Experiments

We compare the performance of our PSO wtih several benchmark pruners, including magnitude pruning (Han et al., 2015), SNIP (Lee et al., 2019), Synflow (Tanaka et al., 2020), GraSP (Wang et al., 2020) and $\ell_0$ norm regularizer (Louizos et al., 2018) on the CIFAR-10/100 benchmarks (Krizhevsky, 2009) using different model architectures (ResNet-20 (He et al., 2016), VGG16-bn (Simonyan & Zisserman, 2015) and WRN-20 (Zagoruyko & Komodakis, 2016)) at different target sparsity levels.

As illustrated in Figure 3, PSO achieves the best one-shot pruning performance across various models and sparsity levels. Though the SpODE mask is evaluated globally, PSO excels even at high sparsity levels, indicating it can handle extreme compression and avoid layer-collapse issues (Tanaka et al., 2020) faced by conventional global pruners.

Notably, as shown in Table 1, the one-shot PSO outperforms or matches the performance of the Iterative Magnitude Pruner (IMP), a strong benchmark (Zimmer et al., 2022), with a much smaller overhead. Additionally, incorporating an iterative pruning scheme can further improve PSO's performance to achieve SOTA results (rightest column of Table 1). Detailed results are shown in Figure 4.

| | Sparsity | Mag | IMP | **PSO (ours)** | **Itr-PSO (ours)** |
|---|---|---|---|---|---|
| ResNet-20<br>(70.38) | 90% | 62.52 | 63.41 | 63.38 | **64.95** |
| | 93% | 58.41 | 60.23 | 60.09 | **62.08** |
| | 95% | 53.64 | 55.57 | 56.11 | **59.24** |
| | 96.5% | 47.10 | 50.44 | 51.67 | **54.83** |
| | 98% | 29.52 | 35.95 | 40.92 | **45.29** |
| VGG16-bn<br>(75.68) | 90% | 74.53 | 74.61 | 74.53 | **74.70** |
| | 93% | 73.76 | 74.02 | 74.24 | **74.30** |
| | 95% | 71.86 | **73.99** | 73.98 | 73.85 |
| | 96.5% | 70.02 | 73.14 | 72.92 | **73.29** |
| | 98% | 64.62 | 71.53 | 71.07 | **71.64** |
| WRN-20<br>(75.22) | 90% | 70.62 | 70.86 | 71.00 | **71.26** |
| | 93% | 69.11 | 70.05 | 69.74 | **70.09** |
| | 95% | 66.47 | 68.31 | 67.88 | **68.93** |
| | 96.5% | 63.19 | 65.19 | 65.40 | **67.79** |
| | 98% | 54.96 | 59.21 | 59.92 | **63.17** |

*Table 1.* Comparison results of unstructured Pruning on CIFAR-100. One-shot PSO is able to outperform IMP. The underline represents results better or comparable (in 0.1%) than IMP. Itr-PSO stands for PSO with iterative pruning scheme.

### 5.2. Tiny-ImageNet and ImageNet Experiments

In one-shot pruning scenarios, our PSO achieves SOTA results for both unstructured and structured pruning on Tiny-ImageNet (Le & Yang, 2015) with ResNet-50, VGG19-bn, and WRN-34, and ImageNet (Deng et al., 2009) with VGG16-bn and ResNet-50 as shown in Table 2 and Table 3. In addition, PSO narrows the performance difference between unstructured and structured pruning, where the latter is traditionally considered more challenging and can lead to significant model acceleration. Table 3 shows that both the one-shot PSO can be scaled up to large-scale datasets, e.g., ImageNet, and is able to maintain high performance even in high sparsity regimes. Furthermore, we performed a comparison between the performance of PSO on ImageNet with batch sizes of 64 and 128 in Table 4. The results indicate that our PSO exhibits further improvement when the batch size of SpODE is increased. We hypothesize that this enhancement in performance can be attributed to a more precise estimation of the energy gradient, resulting in a more accurate discretization of SpODE.

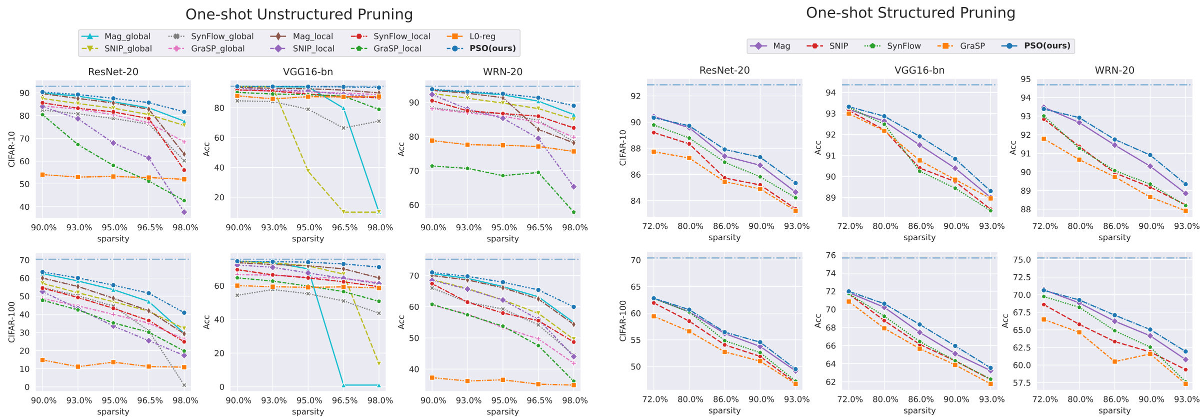

*Figure 3.* Results of one-shot pruning on CIFAR-10/100 dataset. The x-axis is the sparsity and the y-axis is the top-1 accuracy of the tuned sparse model. The horizontal dash line represents the performance of the unpruned model $\boldsymbol{\theta}^*$.

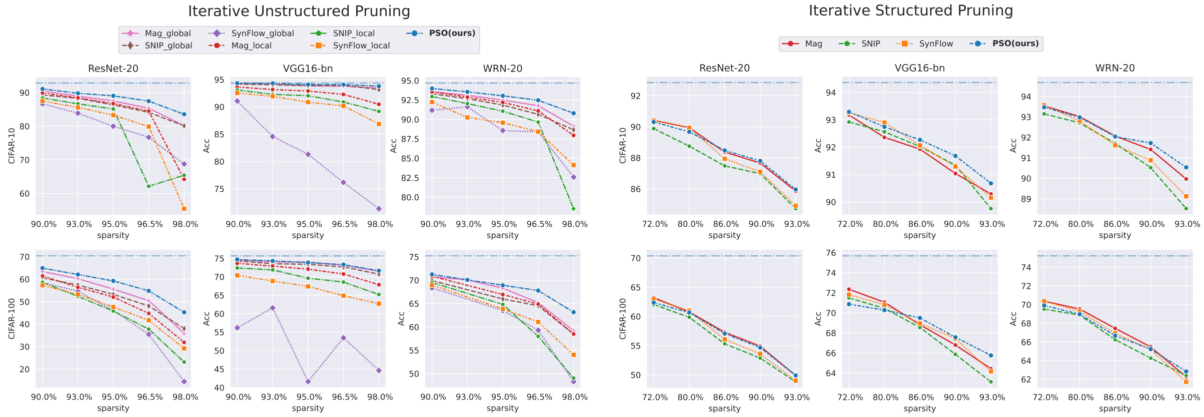

*Figure 4.* Results of iterative pruning on CIFAR-10/100 dataset. The x-axis is the sparsity and the y-axis is the top-1 accuracy of the tuned sparse model. The horizontal dash line is the accuracy of the unpruned model.

To validate the effectiveness of our PSO method and compare it fairly with several iterative pruning algorithms, including Powerpropagation (Schwarz et al., 2021), STR (Kusupati et al., 2020), Woodfisher (Singh & Alistarh, 2020), and ProbMask (Zhou et al., 2021), we conducted iterative pruning experiments on ImageNet using the same ResNet-50 checkpoint and experimental setup as (Kusupati et al., 2020). Specifically, we ran the PSO method for prune-finetune iterations, with each iteration involving a SpODE with a batch size of 256 and $N = 1000$, followed by 5 epochs of model retraining. Finally, we finetuned the pruned ResNet-50 for 50 epochs to further enhance its performance. Therefore, our PSO method was run for a total of $100 = 5 \times 10 + 50$ epochs, just like the other iterative pruning baselines.

As shown in the Table 5, when evaluated under the same iterative pruning setting, our PSO method shows either superior or comparable pruning performance compared to these baseline algorithms. It's worth noting that running the SpODE with a batch size of 256 for $1000 \times 10$ steps incurs only a small computational overhead, which is equivalent to that of finetuning the model for just ONE additional epoch on ImageNet.

| | Sparsity | Mag | SNIP | SynFlow | GraSP | **PSO (ours)** |
|---|---|---|---|---|---|---|
| ResNet-50<br>(67.06) | 90% / 72% | 60.51 / 63.40 | 57.62 / 64.26 | 56.09 / 63.78 | 52.78 / 59.14 | **63.47 / 63.60** |
| | 93% / 80% | 57.18 / 62.84 | 53.91 / 63.03 | 54.79 / 63.07 | 49.16 / 57.21 | **59.37 / 63.37** |
| | 95% / 86% | 56.64 / 61.36 | 54.33 / 61.77 | 53.85 / 61.37 | 43.95 / 57.00 | **61.20 / 62.73** |
| | 96.5% / 90% | 53.42 / 58.98 | 53.87 / 56.16 | 50.33 / 58.45 | 32.42 / 54.76 | **57.61 / 61.63** |
| | 98% / 93% | 51.90 / 56.81 | 52.94 / 58.07 | 41.00 / 57.23 | 10.56 / 53.99 | **53.82 / 59.62** |
| VGG19-bn<br>(63.47) | 90% / 72% | 62.32 / 59.58 | 61.66 / 59.27 | 0.50 / 0.50 | 43.73 / 56.84 | **62.67 / 60.37** |
| | 93% / 80% | 61.65 / 58.33 | 60.22 / 58.01 | 0.50 / 0.50 | 43.52 / 55.51 | **62.05 / 59.08** |
| | 95% / 86% | **61.74 / 57.34** | 56.08 / 55.75 | 0.50 / 0.50 | 42.86 / 50.90 | 61.54 / 57.08 |
| | 96.5% / 90% | 60.46 / 54.99 | 46.54 / 52.38 | 0.50 / 0.50 | 42.37 / 46.16 | **60.60 / 55.22** |
| | 98% / 93% | 53.26 / 49.82 | 23.33 / 46.75 | 0.50 / 0.50 | 39.42 / 41.06 | **57.63 / 50.65** |
| WRN-34<br>(64.74) | 90% / 72% | 61.59 / 60.23 | 61.43 / 60.32 | 57.87 / 60.44 | 53.37 / 59.85 | **62.36 / 61.11** |
| | 93% / 80% | 61.11 / 59.35 | 60.16 / 59.28 | 57.57 / 59.61 | 52.75 / 57.13 | **61.17 / 59.91** |
| | 95% / 86% | 59.97 / 58.30 | 51.31 / 58.14 | 50.11 / 58.20 | 49.67 / 55.30 | **60.14 / 58.38** |
| | 96.5% / 90% | **58.98** / 57.25 | 56.45 / 55.65 | 54.32 / 56.23 | 49.67 / 53.11 | 58.75 / **57.28** |
| | 98% / 93% | 56.39 / 55.40 | 54.60 / 54.12 | 50.42 / 52.03 | 47.17 / 51.62 | **57.54 / 57.03** |

*Table 2.* Comparison results of one-shot Unstructured / Structured Pruning on Tiny-ImageNet. The numbers in the parentheses are the performance of the unpruned model.

| | VGG16-bn (73.36) | | ResNet-50 (76.128) | |
|---|---|---|---|---|
| Sparsity | 90% | 95% | 90% | 95% |
| Mag | 73.15 | 70.89 | 73.07 | 68.88 |
| **PSO (ours)** | **73.25** | **71.11** | **73.19** | **68.94** |

*Table 3.* One-shot unstructured pruning on ImageNet. The numbers in the parentheses are the performance of the unpruned model.

| ResNet-50 | Unpruned Acc. | Sparisty | SpODE Batchsize | Finetune Epochs | Final Acc. |
|---|---|---|---|---|---|
| Mag | 76.13% | 90.00% | 64 | 45 | 73.07% |
| PSO (ours) | 76.13% | 90.00% | 64 | 45 | 73.19% |
| PSO (ours) | 77.01% | 90.00% | 64 | 100 | 73.54% |
| **PSO (ours)** | 77.01% | 90.00% | **128** | 100 | **74.20%** |

*Table 4.* The performance of one-shot PSO can be further improved with a larger batchsize in the discretization of SpODE.

| ResNet-50 | Unpruned Acc. | Sparisty | Final Acc. |
|---|---|---|---|
| STR (Kusupati et al., 2020) | 77.01% | 87.70% | 74.73% |
| WoodFisher (Singh & Alistarh, 2020) | 77.01% | 90.00% | 75.21% |
| Powerprop (Schwarz et al., 2021) | 76.80% | 90.00% | 74.40% |
| ProbMask (Zhou et al., 2021) | 77.01% | 90.00% | 74.68% |
| **PSO (ours)** | 77.01% | 90.00% | **75.10%** |

*Table 5.* Iterative PSO achieves either better or comparable performance than the state-of-the-art baselines on ImageNet. The experiment follows the same settings of (Kusupati et al., 2020).

### 5.3. Ablation Studies

To evaluate the effectiveness of PSO, we perform detailed ablation experiments and present the results and analysis in this section.

**Ablation on empirical polarizers.** Recall that in Section 4.4, the choice of the empirical polarizer $\widehat{\mathcal{P}}_{\varepsilon,\alpha}(\cdot)$ is crucial to pruning performance. A smaller $\varepsilon$ value implies a less irregular relaxed pruning problem but a larger approximation error of the hard pruning problem, while a smaller $\alpha$ value results in a closer approximation of $\mathcal{P}_\varepsilon$ and stronger polarization effect. To empirically determine the effects of $\varepsilon, \alpha$, we evaluate the one-shot unstructured pruning performance of the three empirical polarizers proposed in Figure 2 with varying $(\varepsilon, \alpha)$. The ablation results in Figure 5 show that for all settings, except $(\varepsilon = 0.4, \alpha = 0.3)$, PSO achieves better or comparable performance to the baseline pruner. This implies that PSO's performance is robust to the choice of different polarizers. In practice, it is more preferable to set $\varepsilon < 0.3$ and $\alpha < 0.1$.

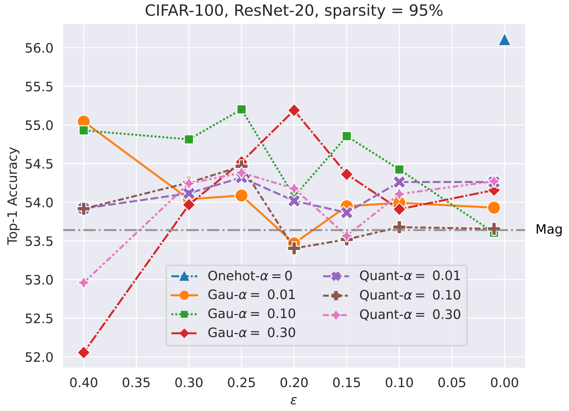

*Figure 5.* Ablation study on polarizers, where 'Quant' and 'Gau' refers to the quantile polarizer and Gaussian polarizer respectively.

**Ablation on SpODE discretization schemes.** As the PSO pruning score is obtained by solving the SpODE, the performance of PSO highly relies on the SpODE discretization scheme used in Algorithm 1. The SpODE discretization scheme is determined by the number of discretization steps $N$, the localization parameter $r_t$ and the schedule of sparsity increase $\Delta t$. For simplicity, $r_t$ is set as a constant with value $r$. We then implement PSO with various $(N, r)$ and $\Delta t$ schedules. Empirically, PSO is robust to the choice of $(N, r)$, as shown in Figure 6. For CIFAR-100 / VGG16-bn, PSO with only 20 discretization steps can compress the model to 95% sparsity with a 2.5% performance drop.

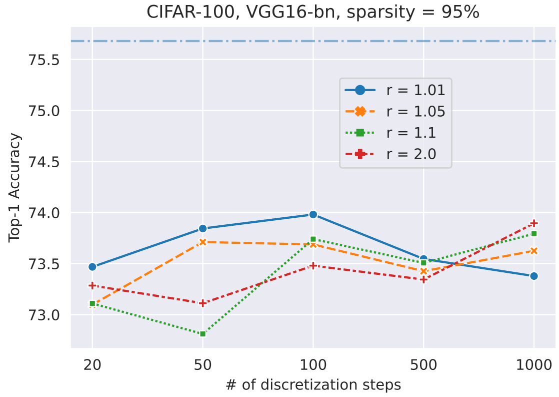

*Figure 6.* Ablation study on choices of $(N, r)$.

In Figure 7 we visualize three types of sparsity schedules (left) and we track the loss function value along the associated SpODE paths (middle). In high sparsity regimes, the exponential schedule demonstrates its ability to effectively preserve the loss value, whereas the linear and inverse-exponential schedules cause an evident increase in the loss function. Furthermore, when using an exponential $\Delta t$ schedule, PSO attains the highest top-1 accuracy across multiple sparsity levels. We hypothesize that, in high sparsity regimes, the population of the optimal masks is scarcer, making the SpODE path more irregular and harder to be approximated numerically. Therefore, it is recommended to allocate more computational resources to the high sparsity regimes by using decaying $\Delta t$ schedules like the exponential schedule.

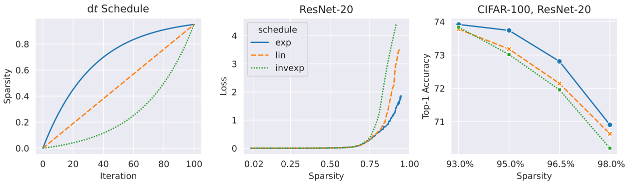

*Figure 7.* Ablation study on three SpODE $\Delta t$ schedules, including exponential, linear, and inverse-exponential schedules.

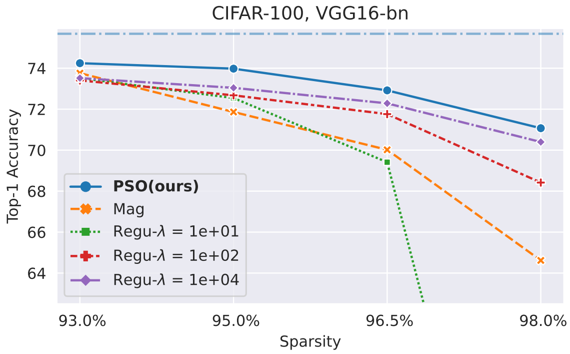

*Figure 8.* Ablation on the SpODE explicit update. 'Regu-$\lambda$' represents the results of the regularized counterpart of PSO with a $\lambda$ penalty weight.

**Ablation on explicit SpODE update.** To confirm the superiority of using the explicit SpODE update, we compare the performance of the standard PSO with that of the regularized-based counterpart of PSO with a penalized objective function $\mathcal{E}_\varepsilon + \lambda/2(G-(1-d'/d))^2$, $\lambda > 0$ is the penalty weight. The regularized counterpart updates the mask by standard SGD until the target sparsity is reached. As shown in Figure 8, the regularized-based pruner struggles to achieve the same level of performance as the standard PSO, highlighting the advantage of using explicit SpODE update. However, with a fine-tuned regularization parameter (e.g. $\lambda = 10^4$), the regularized-based pruner can still surpass other baseline pruners. This can be attributed to the fact that our polarized soft pruning model (3) effectively approximates the hard pruning problem.

**Implicit Mask Regrowing.** Conventional pruners typically compute pruning scores based on the current unpruned parameters, making them vulnerable to premature and permanent mask removal, unless a well-designed mask-regrowing strategy is implemented. In contrast, PSO travels along the SpODE path that is composed of optimal masks, thus enabling implicit mask regrowing. We empirically verify this property by 1) collecting PSO checkpoints of different sparsity levels along a single SpODE path; 2) pruning each checkpoint and retraining it until convergence; 3) computing the number of regrown masks for each sparsity. As shown in Figure 9, when the sparsity is high, the mask regrowing is intense and the PSO checkpoints exhibit significantly better performance than baseline. This shows that in high sparsity regimes, where the risk of premature removal is high, PSO's improved performance can be partly attributed to its implicit mask-regrowing feature. Moreover, this supports the claim of Proposition 4 that the **SpODE path intersects with optimal masks at various sparsity levels**, providing a robust solution for neural pruning.

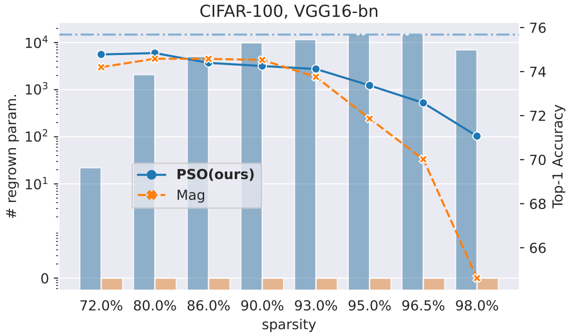

*Figure 9.* PSO exhibits intense implicit mask regrowing in high sparsity regimes.

## 6. Conclusion

In this paper, we proposed a novel Sparsity-indexed ODE (SpODE) pruning framework (Proposition 3) that illuminates the evolution of optimal masks as the sparsity level increases continuously. Then we develop the Pruning via SpODE (PSO) algorithm (Algorithm 1) that enables effective pruning by traveling along the SpODE path. Our PSO achieves SOTA performance across various of pruning settings. Noteworthily, our oneshot-PSO is able to achieve better or comparable performance than the expensive iterative pruners (Table 1); our PSO significantly narrows the performance gap between unstructured pruning and structured pruning (Table 2); PSO allows for implicit mask regrowing, making it more robust in high sparsity regimes (Figure 9).

## Acknowledgement

This study is partially supported under the Hong Kong Global STEM Professorship, the 2020 Microsoft Research Asia collaborative research grant, and Singapore's RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s).

## References

Babu, A., Wang, C., Tjandra, A., Lakhotia, K., Xu, Q., Goyal, N., Singh, K., von Platen, P., Saraf, Y., Pino, J., Baevski, A., Conneau, A., and Auli, M. XLS-R: self-supervised cross-lingual speech representation learning at scale. *CoRR*, abs/2111.09296, 2021. URL https://arxiv.org/abs/2111.09296.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Chen, Q., Zhao, H., Li, W., Huang, P., and Ou, W. Behavior sequence transformer for e-commerce recommendation in alibaba. In *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data*, DLP-KDD '19, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367837. doi: 10.1145/3326937.3341261. URL https://doi.org/10.1145/3326937.3341261.

Chen, T., Ji, B., DING, T., Fang, B., Wang, G., Zhu, Z., Liang, L., Shi, Y., Yi, S., and Tu, X. Only train once: A one-shot neural network training and pruning framework. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=p5rMPjrcCZq.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL http://arxiv.org/abs/1810.04805.

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

Frankle, J. and Carbin, M. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJl-b3RcF7.

Han, S., Pool, J., Tran, J., and Dally, W. Learning both weights and connections for efficient neural network. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/file/ae0eb3eed39d2bcef4622b2499a05fe6-Paper.pdf.

Han, S., Mao, H., and Dally, W. J. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. In Bengio, Y. and LeCun, Y. (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1510.00149.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hoefler, T., Alistarh, D., Ben-Nun, T., Dryden, N., and Peste, A. Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks. *J. Mach. Learn. Res.*, 22(241):1–124, 2021.

Hu, H., Peng, R., Tai, Y.-W., and Tang, C.-K. Network trimming: A data-driven neuron pruning approach towards efficient deep architectures. *arXiv preprint arXiv:1607.03250*, 2016.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/5a4be1fa34e62bb8a6ec6b91d2462f5a-Paper.pdf.

Janner, M., Li, Q., and Levine, S. Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems*, 34: 1273–1286, 2021.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Krizhevsky, A. Learning multiple layers of features from tiny images. pp. 32–33, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Kusupati, A., Ramanujan, V., Somani, R., Wortsman, M., Jain, P., Kakade, S., and Farhadi, A. Soft threshold weight reparameterization for learnable sparsity. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5544–5555. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/kusupati20a.html.

Le, Y. and Yang, X. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Lee, N., Ajanthan, T., and Torr, P. SNIP: SINGLE-SHOT NETWORK PRUNING BASED ON CONNECTION SENSITIVITY. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=B1VZqjAcYX.

Liang, T., Glossner, J., Wang, L., Shi, S., and Zhang, X. Pruning and quantization for deep neural network acceleration: A survey. *Neurocomputing*, 461:370–403, 2021.

Lin, T., Stich, S. U., Barba, L., Dmitriev, D., and Jaggi, M. Dynamic model pruning with feedback. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJem8lSFwB.

Louizos, C., Welling, M., and Kingma, D. P. Learning sparse neural networks through l0 regularization. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1Y8hhg0b.

Lubana, E. S. and Dick, R. A gradient flow framework for analyzing network pruning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=rumv7QmLUue.

Morcos, A., Yu, H., Paganini, M., and Tian, Y. One ticket to win them all: generalizing lottery ticket initializations across datasets and optimizers. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/a4613e8d72a61b3b69b32d040f89ad81-Paper.pdf.

Mostafa, H. and Wang, X. Parameter efficient training of deep convolutional neural networks by dynamic sparse reparameterization. In *International Conference on Machine Learning*, pp. 4646–4655. PMLR, 2019.

Papadimitriou, C. H. and Steiglitz, K. *Combinatorial optimization: algorithms and complexity*. Courier Corporation, 1998.

Rachwan, J., Zügner, D., Charpentier, B., Geisler, S., Ayle, M., and Günnemann, S. Winning the lottery ahead of time: Efficient early network pruning. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 18293–18309. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/rachwan22a.html.

Radosavovic, I., Kosaraju, R. P., Girshick, R., He, K., and Dollár, P. Designing network design spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10428–10436, 2020.

Schwarz, J., Jayakumar, S. M., Pascanu, R., Latham, P. E., and Teh, Y. W. Powerpropagation: A sparsity inducing weight reparameterisation. In *Neural Information Processing Systems*, 2021.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition, 2015.

Singh, S. P. and Alistarh, D. Woodfisher: Efficient second-order approximation for neural network compression. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Srinivas, S. and Babu, R. V. Data-free parameter pruning for deep neural networks. *arXiv preprint arXiv:1507.06149*, 2015.

Tanaka, H., Kunin, D., Yamins, D. L., and Ganguli, S. Pruning neural networks without any data by iteratively conserving synaptic flow. *Advances in Neural Information Processing Systems*, 33:6377–6389, 2020.

Wang, C., Zhang, G., and Grosse, R. Picking winning tickets before training by preserving gradient flow. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkgsACVKPH.

Xiao, X., Wang, Z., and Rajasekaran, S. Autoprune: Automatic network pruning by regularizing auxiliary parameters. *Advances in neural information processing systems*, 32, 2019.

Zagoruyko, S. and Komodakis, N. Wide residual networks, 2016. URL https://arxiv.org/abs/1605.07146.

Zhang, T., Ye, S., Zhang, K., Tang, J., Wen, W., Fardad, M., and Wang, Y. A systematic dnn weight pruning framework using alternating direction method of multipliers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 184–199, 2018.

Zhou, X., Zhang, W., Xu, H., and Zhang, T. Effective sparsification of neural networks with global sparsity constraint. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3598–3607, 2021. doi: 10.1109/CVPR46437.2021.00360.

Zhu, M. H. and Gupta, S. To prune, or not to prune: Exploring the efficacy of pruning for model compression. In *International Conference on Learning Representations Workshop*, 2018. URL `https://openreview.net/forum?id=S1lN69AT-`.

Zhuang, T., Zhang, Z., Huang, Y., Zeng, X., Shuang, K., and Li, X. Neuron-level structured pruning using polarization regularizer. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9865–9877. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/703957b6dd9e3a7980e040bee50ded65-Paper.pdf`.

Zimmer, M., Pokutta, S., and Spiegel, C. Back to basics: Efficient network compression via imp, 2022. URL `https://openreview.net/forum?id=AsDSpwXYGeT`.

# A. Implementation Details

## A.1. Pruning Configurations

A general pruning evaluation process consists of three key steps:

1. Compressing a pretrained model $\boldsymbol{\theta}^*$ to the target sparsity level using a specific **pruner** and **pruning configuration**.
2. Retraining the pruned sparse model for some epochs to achieve convergence.
3. Evaluate the average top-1 classification accuracy of the associated pruned model in the last several retraining epochs.

Different pruning configurations **vary only in step 1**, while the subsequent retraining (Table 6) and evaluating procedures remain consistent across all settings, enabling a fair comparison of the pruning methods. Specifically, a pruning configuration is determined by the pruning scope (unstructured / structured) and the pruning scheme (one-shot/iterative prune-and-retrain).

**Pruning scopes.** In this work, we conduct experiments on both types of pruning scopes: unstructured and structured pruning. For unstructured (structured) pruning, we first compute the pruning score of each individual parameter (each neuron) and cancel out those with the lowest pruning scores. For unstructured pruning methods, we extend them to structured pruning by defining the neuron score as the sum of the scores of the individual weights associated with that neuron. Following the pruning conventions, we leave the first convolutional layer and all batch normalization layers unpruned.

**Pruning schemes.** Experiments are also conducted on both types of pruning schemes: one-shot and iterative pruning. Generally, one-shot pruning emphasizes that the parameter elimination (but the pruning score evaluation can take place multiple times) is performed only once before final retraining; on the contrary iterative pruning allows for multiple prune-and-retrain cycles. Thus, an iterative pruning configuration is characterized by a series of intermediate sparsity levels (also known as a pruning schedule) and the optimization strategy for retraining at each intermediate stage. For all pruners, we adopt the traditional exponential pruning schedule. In each prune-and-retrain iteration, the intermediate sparse model is tuned for 5 epochs with the same training strategy as the final retraining stage (Table 6). In addition, the batch size of dataset (if needed) used for pruning score evaluation is 256 for CIFAR-10/100, and 64 for Tiny ImageNet and ImageNet.

*Table 6.* Hyperparameter configurations for retraining procedure.

| | VGG16-bn | | VGG19-bn | WRN20 | WRN34 | ResNet-20 | ResNet-50 | |
|---|---|---|---|---|---|---|---|---|
| | CIFAR-10/100 | ImageNet | Tiny ImageNet | CIFAR-10/100 | Tiny ImageNet | CIFAR-10/100 | Tiny ImageNet | ImageNet |
| **Optimizer** | SGD-Momentum | SGD-Momentum | SGD-Momentum | SGD-Momentum | SGD-Momentum | SGD-Momentum | SGD-Momentum | SGD-Momentum |
| **Training Epochs** | 100 | 45 | 70 | 100 | 70 | 100 | 70 | 45 |
| **Batch Size** | 64 | 128 | 128 | 64 | 128 | 64 | 128 | 128 |
| **Learning Rate** | 1e-2 | 1e-2 | 1e-2 | 1e-2 | 1e-2 | 1e-2 | 1e-2 | 1e-2 |
| **Learning Rate Schedule** | CosineAnnealing | CosineAnnealing | CosineAnnealing | CosineAnnealing | CosineAnnealing | CosineAnnealing | CosineAnnealing | CosineAnnealing |
| **Minimal Learning Rate and When to Reach** | 1e-6<br>last 10 epochs | 1e-6<br>last 0 epochs | 1e-6<br>last 10 epochs | 1e-6<br>last 10 epochs | 1e-6<br>last 10 epochs | 1e-6<br>last 10 epochs | 1e-6<br>last 10 epochs | 1e-6<br>last 0 epochs |
| **Evaluated Epochs** | last 10 epochs | last 3 epochs | last 10 epochs | last 10 epochs | last 10 epochs | last 10 epochs | last 10 epochs | last 3 epochs |

## A.2. Implementation of Pruning via Sparsity-indexed ODE

A detailed version of Pruning via Sparsity-indexed ODE (PSO) algorithm is shown in Algorithm 2.

## A.3. Implementation of Empirical Polarizers

In this section, we elaborate the implementation of three empirical mask polarizers, including the one-hot polarizer (Algorithm 3), the quantile polarizer (Algorithm 4), and the Gaussian polarizer (Algorithm 5).

**Algorithm 2** Pruning via Sparsity-indexed ODE (PSO)

```
**Input:** reference model $f_{\boldsymbol{\theta}^*}(\cdot)$, loss function $l(\cdot,\cdot)$, target parameter budget $d'$, soft sparsity function $G(\cdot)$, empirical mask polarizer $\widehat{\mathcal{P}}_{\varepsilon,\alpha}(\cdot)$, localization shceme $r_t$, SpODE discretization steps number $N$.
**Output:** a hard mask pruned by SpODE $\widehat{\mathbf{m}} \in \{0,1\}^d$.
Initialization $t \leftarrow 0$, $\widetilde{\mathbf{m}}_t \leftarrow \mathbf{1}$, $\Delta t \leftarrow (1 - d'/d)/N$
**for** $i = 1$ **to** $N$ **do**
    $\widehat{\mathbf{m}}_t \leftarrow \widehat{\mathcal{P}}_{\varepsilon,\alpha}(\widetilde{\mathbf{m}}_t)$ {Mask polarization}
    $(\mathbf{x}, \mathbf{y}) \leftarrow \mathrm{Mini\_Batch}$
    $\mathcal{E}_\varepsilon(\widehat{\mathbf{m}}_t) \leftarrow l(f_{\boldsymbol{\theta}^*\odot\widehat{\mathbf{m}}_t}(\mathbf{x}), \mathbf{y})$
    $\mathbf{e} \leftarrow \nabla\mathcal{E}_\varepsilon(\widehat{\mathbf{m}}_t)$ {Calculate gradient w.r.t energy (mini-batch lossa)}
    $\mathbf{g} \leftarrow \nabla G(\widehat{\mathbf{m}}_t)$ {Calculate gradient w.r.t soft sparsity}
    **if** $\|\mathbf{g}\|\|\mathbf{e}\| \neq (\mathbf{g}^\top\mathbf{e})^2$ **then**
        $x \leftarrow \sqrt{(r^2-1)/((\|\mathbf{g}\|\|\mathbf{e}\|)^2 - (\mathbf{g}^\top\mathbf{e})^2)}$
        $y \leftarrow (1 - \mathbf{g}^\top\mathbf{e}x)/\|\mathbf{g}\|^2$
        $F(\widetilde{\mathbf{m}}_t) \leftarrow x\mathbf{e} + y\mathbf{g}$
    **else**
        $F(\widetilde{\mathbf{m}}_t) \leftarrow \mathbf{g}/\|\mathbf{g}\|^2$
    **end if**
    $\widetilde{\mathbf{m}}_{t+\Delta t} \leftarrow \widetilde{\mathbf{m}}_t + F(\widetilde{\mathbf{m}}_t)\Delta t$ {SpODE discretization by Eq. (8)}
    $t \leftarrow t + \Delta t$
**end for**
$\widehat{\mathbf{m}} \leftarrow \mathrm{top}_{d'}(\widehat{\mathcal{P}}_{\varepsilon,\alpha}(\widetilde{\mathbf{m}}_{d'/d}))$ {Empirical mask polarization}
**return** $\widehat{\mathbf{m}}$
```

**Algorithm 3** One-hot Polarizer

```
**Input:** a soft dense mask $\mathbf{m} \in \mathbb{R}^d$, soft sparsity function $G(\cdot)$.
**Output:** a nearly sparse $\widehat{\mathcal{P}}^{\mathrm{oh}}(\mathbf{m}) \in \{0,1\}^d$.
$t \leftarrow G(\mathbf{m})$
$k \leftarrow \lceil td \rceil$
$\widehat{\mathcal{P}}^{\mathrm{oh}}(\mathbf{m}) \leftarrow \mathrm{top}_k(\mathbf{m})$
```

**Algorithm 4** Quantile Polarizer

```
**Input:** a soft dense mask $\mathbf{m} \in \mathbb{R}^d$, soft sparsity function $G(\cdot)$, hyperparameters $\varepsilon, \alpha \in (0,1)$.
**Output:** a nearly sparse $\widehat{\mathcal{P}}^{\mathrm{quant}}_{\varepsilon,\alpha}(\mathbf{m}) \in [0,1]^d$.
$t \leftarrow G(\mathbf{m})$
$C_l, C_u \leftarrow \mathrm{quantile}_{\alpha*(1-t)}(\mathbf{m}), \mathrm{quantile}_{(1-t)}(\mathbf{m})$ {Calculate the quantiles of $\mathbf{m}$}
$C_l', C_u' \leftarrow \mathrm{logit}(\varepsilon), \mathrm{logit}(1-\varepsilon)$
$\widehat{\mathcal{P}}^{\mathrm{quant}}_{\varepsilon,\alpha}(\mathbf{m}) \leftarrow \mathrm{sigmoid}(\frac{C_u'-C_l'}{C_u-C_l}(\mathbf{m}-C_l)+C_l')$ {Match to sigmoid distribution via quantiles alignment}
```

**Algorithm 5** Gaussian Polarizer

```
**Input:** a soft dense mask $\mathbf{m} \in \mathbb{R}^d$, soft sparsity function $G(\cdot)$, hyperparameters $\varepsilon, \alpha \in (0,1)$.
**Output:** a nearly sparse $\widehat{\mathcal{P}}^{\mathrm{gau}}_{\varepsilon,\alpha}(\mathbf{m}) \in [0,1]^d$.
$t \leftarrow G(\mathbf{m})$
$c_l, c_u \leftarrow \mathrm{quantile}_{\alpha*(1-t)}(N(0,1)), \mathrm{quantile}_{(1-t)}(N(0,1))$ {Approximate the quantiles of $\mathbf{m}$ with Gaussian quantiles}
$\mu, \sigma \leftarrow \mathrm{mean}(\mathbf{m}), \mathrm{std}(\mathbf{m})$
$C_l, C_u \leftarrow \mu - c_l\sigma, \mu + c_u\sigma$
$C_l', C_u' \leftarrow \mathrm{logit}(\varepsilon), \mathrm{logit}(1-\varepsilon)$
$\widehat{\mathcal{P}}^{\mathrm{gau}}_{\varepsilon,\alpha}(\mathbf{m}) \leftarrow \mathrm{sigmoid}(\frac{C_u'-C_l'}{C_u-C_l}(\mathbf{m}-C_l)+C_l')$ {Match to sigmoid distribution via quantiles alignment}
```

## B. Theoretical Results

| Method | $\mathcal{E}(\boldsymbol{\theta})$ | Energy Type |
|---|---|---|
| $\ell_1$ Magnitude (Han et al., 2015) | $\Vert\boldsymbol{\theta}\Vert_2$ | Capacity mearsure |
| SNIP (Lee et al., 2019) | $L(\boldsymbol{\theta})$ | Evaluation loss |
| GraSP (Wang et al., 2020) | $\Vert\nabla L(\boldsymbol{\theta})\Vert_2$ | Model optimality |
| SynFlow (Tanaka et al., 2020) | $\mathbf{1}^\top \prod_l \lvert\boldsymbol{\theta}_l\rvert\mathbf{1}$ | Capacity mearsure |

*Table 7.* Unifying various neural pruning methods via the energy preservation viewpoint. $\boldsymbol{\theta}_l$ denotes the weight matrix of the $l$-th layer of deep model $\boldsymbol{\theta}$.

**Proposition 5** (Greedy Path of Optimal Masks (formal version of Proposition 1))**.** *For any sparsity level $t \in [0,1]$, we define $\Gamma_t \triangleq \{\mathbf{m} : \ G(\mathbf{m}) = t\}$, $\mathcal{M}_t$ as the set of all optima of* (3) *at sparsity level $t$ and $\mathcal{E}_t^*$ as the associated optimal energy value. Assume that*

- *Locally regularity of $\mathcal{M}_t$: $\forall\, \mathbf{m}_t \in \mathcal{M}_t$, $\forall\, r > 0$ there exists $\delta > 0$ s.t. $\mathcal{M}_{t-\delta} \cap B_r(\mathbf{m}_t) \neq \varnothing$.*
- *Finite critical sparsity levels: the set of critical sparsity levels*

  $$\mathrm{crit}(\mathcal{E}_\varepsilon, G) \triangleq \{t : \ \exists\, \mathbf{m}_t \in \mathcal{M}_t \text{ s.t. } \mathcal{E}_\varepsilon(\cdot) \textit{ is not differentiable at } \mathbf{m}\}$$

  *is finite, i.e. $\exists\, K \in \mathbb{Z}$, s.t. $\mathrm{crit}(\mathcal{E}_\varepsilon, G) = \{\tau_i\}_{i=1}^K$, $\tau_i < \tau_j$ if $i < j$.*
- *Locally regularity of $G(\cdot)$: for any $\mathbf{m} \notin \bigcup_{t\in\mathrm{crit}(\mathcal{E}_\varepsilon,G)} \Gamma_t$, there exists $R, C > 0$, s.t. for any $\mathbf{m}'$ satisfying $\Vert\mathbf{m}-\mathbf{m}'\Vert \leqslant R$, it holds that*

  $$\Vert\mathbf{m} - \mathbf{m}'\Vert \leqslant C|G(\mathbf{m}) - G(\mathbf{m}')|.$$

*For any fixed $\Delta t$ and any total order $\preceq$ on $\mathbb{R}^d$, we are able to construct a discretely indexed series of optimal masks $\{\mathbf{m}_t\}_{t\in\mathcal{T}}$, where $\tau_0 \triangleq 0, \tau_{K+1} \triangleq 1$ and*

$$\mathcal{T} \triangleq \bigcup_{i=0}^{K-1} \mathcal{T}_{\tau_i,\tau_{i+1}},\ \mathcal{T}_{a,b} \triangleq \{a, b\} \cup \{a + k\Delta t\}_{k=1}^{\lceil (b-a)/\Delta t\rceil},$$

*such that*

$$\mathbf{m}_0 \triangleq \mathbf{1},\ \mathbf{m}_{t_{i+1}} \triangleq \min_{\preceq}\left(\arg\min_{\mathbf{m}'\in\mathcal{M}_{t_{i+1}}} \Vert\mathbf{m}' - \mathbf{m}_{t_i}\Vert\right),$$

*where $t_{i+1}$ is the successive element of $t_i \in \mathcal{T}$. Then, when we take $\Delta t \to 0$, $\{\mathbf{m}_t\}_{t\in\mathcal{T}}$ converges to $(\mathbf{m}_t^*)_{t\in[0,1]}$, a piecewise locally Lipschitz function on* $[0,1]$.

*Proof.* We only need to show that $\lim_{\Delta t\to 0} \Vert\mathbf{m}_{t+\Delta t} - \mathbf{m}_t\Vert = 0$, where $t, t+\Delta t \in \mathcal{T}\backslash\mathrm{crit}(\mathcal{E}_\varepsilon, G)$. This implies that, as the resolution $\Delta t$ tends to zero, any $\mathbf{m}_t^*$ can be approximated by elements in $\{\mathbf{m}_t\}_{t\in\mathcal{T}}$ at any precision.

Suppose $t \notin \mathrm{crit}(\mathcal{E}_\varepsilon, G)$, for any $\epsilon > 0$, we need to find a $\Delta t$ s.t. $\Vert\mathbf{m}_{t+\Delta t} - \mathbf{m}_t\Vert \leqslant \epsilon$. By the locally regularity of $G$, there exists $R, C > 0$ s.t. $\Vert\mathbf{m} - \mathbf{m}_t\Vert \leqslant C|G(\mathbf{m}) - t|$, $\forall\, \mathbf{m} \in B_R(\mathbf{m}_t)$. Since for sufficiently small $\Delta t$, $\mathcal{M}_{t+\Delta t} \cap B_R(\mathbf{m}_t) \neq \varnothing$, the proof is completed by

$$\Vert\mathbf{m}_{t+\Delta t} - \mathbf{m}_t\Vert = \min_{\mathbf{m}\in\mathcal{M}_{t+\Delta t}} \Vert\mathbf{m} - \mathbf{m}_t\Vert \leqslant \min_{\mathbf{m}\in\mathcal{M}_{t+\Delta t}\cap B_R(\mathbf{m}_t)} \Vert\mathbf{m} - \mathbf{m}_t\Vert \leqslant C\Delta t.$$

□

**Example B.1** (An example of the total order $\preceq$ in Proposition 5)**.** *We are able to define a total order $\preceq$ over $\mathbb{R}^d$ based on a reference model $\boldsymbol{\theta}^* \in \boldsymbol{\Theta}$. Suppose there are no value collisions in the entries of $|\boldsymbol{\theta}^*|$, i.e. $|\boldsymbol{\theta}[i]| \neq |\boldsymbol{\theta}[j]|$ if $i \neq j$. We sort $|\boldsymbol{\theta}|$ in an descending order such that $|\boldsymbol{\theta}[i_1]| \geqslant |\boldsymbol{\theta}[i_2]| \geqslant \cdots \geqslant |\boldsymbol{\theta}[i_d]|$. Then, the total order is defined by:*

$$\mathbf{m} \preceq \mathbf{m}' \iff \exists\, k \leqslant d,\ \text{s.t. } |\mathbf{m}[j]| = |\mathbf{m}'[j]|,\ \forall\, j < i_k,\ |\mathbf{m}[i_d]| \leqslant |\mathbf{m}'[i_d]|. \tag{10}$$

*The total order is merely an auxiliary to ensure that we can choose a unique successive state when constructing $\{\mathbf{m}_t\}_{\mathcal{T}}$ and calculating the projection $\mathcal{E}_\varepsilon(\cdot)$.*

**Definition 4** ((Formal) Mask Polarizer)**.** The mask polarizer $\mathcal{P}_\varepsilon(\cdot)$ is defined as the minimum element (w.r.t a total order $\preceq$) projection from $\mathbb{R}^d$ to $I_\varepsilon \cap \Gamma_t$, i.e.

$$\mathcal{P}_\varepsilon(\mathbf{m}) \triangleq \min_{\preceq} \arg \min_{\mathbf{m}' \in I_\varepsilon \cap \Gamma_t} \|\mathbf{m} - \mathbf{m}'\|.$$

Thus, the output of a polarizer is always unique.

*Proof of Proposition 2.* The proof directly follows the lemma below.

**Lemma 1.** *The following optimization problem*

$$\max_{\boldsymbol{\delta}} \ \mathbf{e}^\top \boldsymbol{\delta}, \text{ s.t. } \mathbf{g}^\top \boldsymbol{\delta} = a, \ \|\boldsymbol{\delta}\| \leqslant r, \tag{11}$$

*admits a closed-form solution*

$$\boldsymbol{\delta}^* = \begin{cases} a\mathbf{g}/\|\mathbf{g}\|^2, \text{ if } \|\mathbf{g}\|\|\mathbf{e}\| = |\mathbf{g}^\top \mathbf{e}|^2, \\ x\mathbf{e} + y\mathbf{g}, \text{ else,} \end{cases}$$

*where*

$$x \triangleq \sqrt{\frac{(r\|\mathbf{g}\|)^2 - a^2}{(\|\mathbf{g}\|\|\mathbf{e}\|)^2 - (\mathbf{g}^\top \mathbf{e})^2}}, \ y \triangleq (a - (\mathbf{g}^\top \mathbf{e})x)/\|\mathbf{g}\|^2. \tag{12}$$

*Proof of Lemma 1.*

By convex optimization theory, the optimal is attained at the decision boundary $\Gamma$, where $\Gamma \triangleq \{\boldsymbol{\delta} \mid \ \mathbf{g}^\top \boldsymbol{\delta} = a, \ \|\boldsymbol{\delta}\| = r\}$. Notice that, the solution of $\max_{\boldsymbol{\delta} \in \Gamma} \mathbf{e}^\top \boldsymbol{\delta}$ resides at the hyperplane spanned by $\{\mathbf{g}, \mathbf{e}\}$, i.e. $\boldsymbol{\delta}^* = x\mathbf{e} + y\mathbf{g}$, $x, y \in \mathbb{R}_+$. Thus, we can solve the optimal $x, y$ by

$$x\mathbf{g}^\top \mathbf{e} + y\|\mathbf{g}\|^2 = a, \tag{13}$$

$$x^2\|\mathbf{e}\|^2 + 2xy\mathbf{e}^\top \mathbf{g} + y^2\|\mathbf{g}\|^2 = r^2. \tag{14}$$

Thus we have

$$y = (a - (\mathbf{g}^\top \mathbf{e})x)/\|\mathbf{g}\|^2, \tag{15}$$

$$\|\mathbf{e}\|^2 x^2 + \frac{2x}{\|\mathbf{g}\|^2}(a - (\mathbf{g}^\top \mathbf{e})x) \cdot (\mathbf{e}^\top \mathbf{g}) + \frac{1}{\|\mathbf{g}\|^2}(a - (\mathbf{g}^\top \mathbf{e})x)^2 = r^2, \tag{16}$$

which further implies $Ax^2 + 2Bx + C = 0$, where

$$A = \|\mathbf{e}\|^2 - (\mathbf{g}^\top \mathbf{e})^2/\|\mathbf{g}\|^2, \ B = 0, \ C = a^2/\|\mathbf{g}\|^2 - r^2. \tag{17}$$

The proof is hence completed by taking

$$x = \sqrt{\frac{(r\|\mathbf{g}\|)^2 - a^2}{(\|\mathbf{g}\|\|\mathbf{e}\|)^2 - (\mathbf{g}^\top \mathbf{e})^2}}. \tag{18}$$

□

□

**Proposition 6** (Sparsity-Indexed ODE (formal version of Proposition 3))**.** *Following the notations and conditions in 5, taking $\Delta t \to 0$, the series $\{\widetilde{\mathbf{m}}_{k\Delta t}\}$ constructed by $\widetilde{\mathbf{m}}_{t+\Delta t} \triangleq \widetilde{\mathbf{m}}_t + \widetilde{\boldsymbol{\delta}}_t$, where*

$$\widetilde{\boldsymbol{\delta}}_t \triangleq \begin{cases} \nabla G(\mathbf{m}_t)\Delta t, & \text{if } [t - \Delta t, t + \Delta t] \cap \operatorname{crit}(\mathcal{E}_\varepsilon, G) \neq \varnothing \\ F(\mathbf{m}_t)\Delta t, & \text{otherwise} \end{cases} \tag{19}$$

*converges to a piecewise smooth Sparsity-Indexed ODE (SpODE), which is given by*

$$\begin{aligned} \mathrm{d}\widetilde{\mathbf{m}}_t =& F(\widetilde{\mathbf{m}}_t)\mathrm{d}t,\ t \in [0,1], \\ \widetilde{\mathbf{m}}_0 =& \mathbf{1}, \end{aligned} \tag{20}$$

*where $F(\cdot)$ is defined in* (8)*.*

*Proof.* Following the notations in the proof of Proposition 5, let $\mathrm{crit}(\mathcal{E}_\varepsilon, G) = \{\tau_i\}_{i=1}^K$, we can see both $\mathcal{E}_\varepsilon(\cdot)$ and $G(\cdot)$ are smooth at $\{(\mathcal{M}_t)_{t\in(\tau_{i-1},\tau_i)}\}_{i=1}^K$. Thus, the constructed subsequence $\{\widetilde{\mathbf{m}}_{k\Delta t}\}_{k:(k\Delta t)\in(\tau_{i-1},\tau_i)}$ is a Euler discretization sequence which converges to the SpODE in the sparsity segment $(\tau_{i-1}, \tau_i)$. To complete the proof, it is sufficient to demonstrate that the convergence of the constructed sequence remains unaltered at the critical sparsity levels. By the locally regularity of $G$, for $k$ such that $k\Delta t \leqslant \tau_1 \leqslant (k+1)\Delta t$, we have

$$\|\widetilde{\mathbf{m}}_{(k+1)\Delta t} - \widetilde{\mathbf{m}}_{k\Delta t}\| \leqslant 2 \max_{\kappa\in\{k,k+1\}} \|\nabla G(\widetilde{\mathbf{m}}_{\kappa\Delta t})\|\Delta t,$$

which implies the limitation $\widetilde{\mathbf{m}}_{\tau_i} \triangleq \lim_{\Delta t\to 0} \widetilde{\mathbf{m}}_{k(\Delta t)\Delta t}$ exists. This demonstrates that $\{\widetilde{\mathbf{m}}_{k\Delta t}\}$ converges to a smooth SpODE path on the interval $[0, \tau_1]$. The proof is completed by iteratively applying this argument to the remaining $K$ intervals $\{(\tau_i, \tau_{i+1})\}_{i=1}^K$. Since $K < +\infty$, the constructed sequence converges to the piecewise smooth SpODE on $[0, 1]$ as $\Delta t \to 0$. □

**Proposition 7** (SpODE Estimation (formal version of Proposition 4))**.** *Following the notations in Proposition 5 and Proposition 6, let $(\mathbf{m}_t^*)_{t\in[0,1]}$ be the greedy path defined in Proposition 5, and $(\widetilde{\mathbf{m}}_t)_{t\in[0,1-d'/d]}$ follows the SpODE with an oracle localization scheme $r_t$, where $t \mapsto r_t$ is an a.e. smooth function defined by*

$$r_t \triangleq \|\nabla G(\mathbf{m}_t^*)\|^{-1} \left\| \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{m}_t^* \right\|. \tag{21}$$

*If we further assume*

- *Inevitable energy explosion: $\exists\ \phi > 0$, s.t. $\cos(-\nabla\mathcal{E}_\varepsilon(\mathbf{m}), \nabla G(\mathbf{m})) > \phi$ holds for any $\mathbf{m} \in \{\mathbf{m} : G(\mathbf{m}) > 0\}$.*
- *Dominant first-order information: for any $t \in [0,1]\backslash\mathrm{crit}(\mathcal{E}_\varepsilon, G)$, the projection to $\mathrm{span}(\{\nabla\mathcal{E}_\varepsilon(\mathbf{m}_t^*)\})^\perp$ is a contraction mapping at the vicinity of $\mathbf{m}_t^*$.*
- *Both $\nabla\mathcal{E}_\varepsilon(\cdot)$ and $\nabla G(\cdot)$ are L-locally Lipschitz at $\mathbf{m}_t^*$ for any $t \in [0,1]\backslash\mathrm{crit}(\mathcal{E}_\varepsilon^*, G)$.*

*Then it holds that $\widetilde{\mathbf{m}}_{1-d'/d} = \mathbf{m}_{1-d'/d}^*$.*

*Proof.* We only need to show $\|\widetilde{\mathbf{m}}_t - \mathbf{m}_t^*\| = 0$ holds for any $t$ in the first smooth segment of the SpODE, i.e. $[1, \tau_1]$. Then the proof is completed by repeating this argument on the remaining finite many smooth segments.

We tackle the proof via a discrete argument, then we take the limit $\Delta t \to 0$ for desired conclusions. For any $t \in [1, \tau_1]$ and $\Delta t > 0$, it holds that

$$\|\widetilde{\mathbf{m}}_{t+\Delta t} - \mathbf{m}_{t+\Delta t}^*\| - \|\widetilde{\mathbf{m}}_t - \mathbf{m}_t^*\| \tag{22}$$

$$\leqslant \underbrace{\|F^*(\mathbf{m}_t^*) - F(\mathbf{m}_t^*)\|}_{\text{Localization error}} + \underbrace{\|F(\mathbf{m}_t^*) - F(\widetilde{\mathbf{m}}_t)\|}_{\text{Displacement error}}, \tag{23}$$

where $F^*(\cdot)$ is the oracle displacement function, i.e. $\mathbf{m}_t^* + F^*(\mathbf{m}_t^*)$ is the optimal solution of

$$\min_{\mathbf{m}} \mathcal{E}_\varepsilon(\mathbf{m}),\ \text{s.t. } G(\mathbf{m}) = t + \Delta t,\ \|\mathbf{m} - \mathbf{m}_t^*\| \leqslant r_t\Delta t. \tag{24}$$

To cope with the localization error term, we further introduce an auxiliary displacement function $\widehat{F}(\cdot)$ such that $\mathbf{m}_t^* + \widehat{F}(\mathbf{m}_t^*)$ attains the minimal energy of the following problem

$$\min_{\mathbf{m}} \mathcal{E}_\varepsilon(\mathbf{m}),\ \text{s.t. } \nabla G(\mathbf{m}_t^*)^\top \mathbf{m} = \Delta t,\ \|\mathbf{m} - \mathbf{m}_t^*\| \leqslant r_t\Delta t. \tag{25}$$

Suppose the optimal solution of (25) is $\widehat{\mathbf{m}}_t$, by a second order Taylor expansion argument at $\mathbf{m}_t^*$, it holds that

$$\begin{aligned} &\nabla\mathcal{E}_\varepsilon(\mathbf{m}_t^*)^\top(\widehat{\mathbf{m}}_{t+\Delta t} - \widetilde{\mathbf{m}}_{t+\Delta t}) && (26)\\ \leqslant&\frac{1}{2}(C + \lambda_{\max}(\nabla^2\mathcal{E}_\varepsilon(\mathbf{m}_t^*)))\left(\|(\widehat{\mathbf{m}}_{t+\Delta t} - \mathbf{m}^*_{t+\Delta t})\|^2 + \|(\widetilde{\mathbf{m}}_{t+\Delta t} - \mathbf{m}^*_{t+\Delta t})\|^2\right) && (27)\\ \leqslant& C r_t^2 \Delta t^2, && (28)\end{aligned}$$

where $\lambda^*(\cdot)$ denotes the largest eigen-value of a matrix and $C$ represents the absolute constant. Recall that the projection to $\mathrm{span}(\{\nabla\mathcal{E}_\varepsilon(\mathbf{m}_t^*)\})^\perp$ is a contraction mapping near $\mathbf{m}_t^*$, for $\Delta t$ that is sufficiently small, we have

$$\|\mathbf{P}(\widehat{\mathbf{m}}_{t+\Delta t} - \widetilde{\mathbf{m}}_{t+\Delta t})\| \leqslant \gamma\|\widehat{\mathbf{m}}_{t+\Delta t} - \widetilde{\mathbf{m}}_{t+\Delta t}\|, \tag{29}$$

with $\gamma < 1$ and $\mathbf{P} \triangleq \mathbf{I} - \|\nabla\mathcal{E}_\varepsilon(\mathbf{m}_t^*)\|^{-2}\cdot\nabla\mathcal{E}_\varepsilon(\mathbf{m}_t^*)\nabla\mathcal{E}_\varepsilon(\mathbf{m}_t^*)^\top$. This further implies

$$\begin{aligned} \|(\mathbf{I}-\mathbf{P})(\widehat{\mathbf{m}}_{t+\Delta t} - \widetilde{\mathbf{m}}_{t+\Delta t})\| \geqslant&(1-\gamma)\|\widehat{\mathbf{m}}_{t+\Delta t} - \widetilde{\mathbf{m}}_{t+\Delta t}\|, && (30)\\ \Longrightarrow\ \|\widehat{F}(\mathbf{m}_t^*) - F(\mathbf{m}_t^*)\| = \|\widehat{\mathbf{m}}_{t+\Delta t} - \widetilde{\mathbf{m}}_{t+\Delta t}\| \leqslant& \frac{1}{(1-\gamma)\|\nabla\mathcal{E}_\varepsilon(\mathbf{m}_t^*)\|}|\nabla\mathcal{E}_\varepsilon(\mathbf{m}_t^*)^\top(\widehat{\mathbf{m}}_{t+\Delta t} - \widetilde{\mathbf{m}}_{t+\Delta t})|, && (31)\\ \leqslant& \frac{C r_t^2}{(1-\gamma)\|\nabla\mathcal{E}_\varepsilon(\mathbf{m}_t^*)\|}\Delta t^2 \triangleq C'\Delta t^2. && (32)\end{aligned}$$

This shows $\|\widehat{F}(\mathbf{m}_t^*) - F(\mathbf{m}_t^*)\|$ is a second order term w.r.t $\Delta t$. In addition, by the locally regularity of $G(\cdot)$, it holds that

$$\begin{aligned} &\|\widehat{F}(\mathbf{m}_t^*) - F^*(\mathbf{m}_t^*)\| = \|\widehat{\mathbf{m}}_{t+\Delta t} - \mathbf{m}^*_{t+\Delta t}\| && (33)\\ \leqslant& C|G(\widehat{\mathbf{m}}_{t+\Delta t}) - G(\mathbf{m}^*_{t+\Delta t})| \leqslant \frac{C}{2}(C + \lambda_{\max}(\nabla^2 G(\mathbf{m}_t^*)))\|\widehat{\mathbf{m}}_{t+\Delta t} - \mathbf{m}^*_{t+\Delta t}\|^2 \leqslant C'' r_t^2\Delta t^2. && (34)\end{aligned}$$

The combination of (32) and (34) indicates that the localization error term is bounded by a second-order term w.r.t $\Delta t$, i.e.

$$\|F^*(\mathbf{m}_t^*) - F(\mathbf{m}_t^*)\| \leqslant \mathcal{O}(\Delta t^2).$$

At this point, we only need to upper bound the displacement error term. Following the notations in Propostion 2, we have

$$\begin{aligned} &\|F(\mathbf{m}_t^*) - F(\widetilde{\mathbf{m}}_t)\| = \|\boldsymbol{\delta}_t^* - \widetilde{\boldsymbol{\delta}}_t\|\Delta t && (35)\\ \leqslant& \left(\underbrace{\|x^*\nabla\mathcal{E}_\varepsilon(\mathbf{m}_t^*) - \widetilde{x}\nabla\mathcal{E}_\varepsilon(\widetilde{\mathbf{m}}_t)\|}_{\text{(A)}} + \underbrace{\|y^*\nabla G(\mathbf{m}_t^*) - \widetilde{y}\nabla G(\widetilde{\mathbf{m}}_t)\|}_{\text{(B)}}\right)\Delta t. && (36)\end{aligned}$$

By the $L$-locally Lipschitzness of $\nabla\mathcal{E}_\varepsilon$ and $\nabla G$, it holds that

$$\begin{aligned} &\text{(A)} \leqslant \|\nabla\mathcal{E}_\varepsilon(\widetilde{\mathbf{m}}_t)\|\cdot|x^* - \widetilde{x}| + |\widetilde{x}|\cdot\|\nabla\mathcal{E}_\varepsilon(\mathbf{m}_t^*) - \nabla\mathcal{E}_\varepsilon(\widetilde{\mathbf{m}}_t)\|, && (37)\\ &\text{(B)} \leqslant \|\nabla G(\widetilde{\mathbf{m}}_t)\|\cdot|y^* - \widetilde{y}| + |\widetilde{y}|\cdot\|\nabla G(\mathbf{m}_t^*) - \nabla G(\widetilde{\mathbf{m}}_t)\|, && (38)\\ &\|\nabla\mathcal{E}_\varepsilon(\mathbf{m}_t^*) - \nabla\mathcal{E}_\varepsilon(\widetilde{\mathbf{m}}_t)\| \leqslant L\cdot\|\mathbf{m}_t^* - \widetilde{\mathbf{m}}_t\|, && (39)\\ &\|\nabla G(\mathbf{m}_t^*) - \nabla G(\widetilde{\mathbf{m}}_t)\| \leqslant L\cdot\|\mathbf{m}_t^* - \widetilde{\mathbf{m}}_t\|. && (40)\end{aligned}$$

Since that $|\nabla\mathcal{E}_\varepsilon(\mathbf{m})^\top\nabla G(\mathbf{m})|/(\|\nabla\mathcal{E}_\varepsilon(\mathbf{m})\|\|\nabla G(\mathbf{m})\|) = \cos(\nabla G(\mathbf{m}), \nabla\mathcal{E}_\varepsilon(\mathbf{m})) \geqslant \beta > 0$ holds for any $\mathbf{m}$ with $G(\mathbf{m}) < 1$, both $\widetilde{x}, \widetilde{y}$ are bounded. Moreover, following the notations in Proposition 2, by using (39) and (40), when $\Delta t$ is sufficiently small, it holds that

$$\begin{aligned} &\left|\left((\|\widetilde{\mathbf{g}}_t\|\|\widetilde{\mathbf{e}}_t\|)^2 - (\widetilde{\mathbf{g}}_t^\top\widetilde{\mathbf{e}}_t)^2\right) - \left((\|\mathbf{g}_t^*\|\|\mathbf{e}_t^*\|)^2 - (\mathbf{g}_t^{*\top}\mathbf{e}_t^*)^2\right)\right| && (41)\\ \leqslant& \left|(\|\widetilde{\mathbf{g}}_t\|\|\widetilde{\mathbf{e}}_t\|)^2 - (\|\mathbf{g}_t^*\|\|\mathbf{e}_t^*\|)^2\right| + \left|(\widetilde{\mathbf{g}}_t^\top\widetilde{\mathbf{e}}_t)^2 - (\mathbf{g}_t^{*\top}\mathbf{e}_t^*)^2\right|\\ \leqslant& M(L)\|\widetilde{\mathbf{m}}_t - \mathbf{m}_t^*\|, && (42)\end{aligned}$$

where $M(L)$ is an absolute constant that only depends on $L$. This implies that both $|x^* - \widetilde{x}|$ and $|y^* - \widetilde{y}|$ are upper bounded by $M'(L, C, \beta, t)\|\widetilde{\mathbf{m}}_{t+\Delta t} - \mathbf{m}^*_{t+\Delta t}\|$, where $M'$ is an absolute constant that only depends on $M(L)$, $C$, $\beta$ and $t$.

Now that we have obtained

$$\|\widetilde{\mathbf{m}}_{t+\Delta t} - \mathbf{m}^*_{t+\Delta t}\| - \|\widetilde{\mathbf{m}}_t - \mathbf{m}^*_t\| \leqslant \mathcal{O}(\Delta t^2) + M'(L, C, \beta, t)\|\widetilde{\mathbf{m}}_t - \mathbf{m}^*_t\|\Delta t, \tag{43}$$

by taking $\Delta t \to 0$, we have

$$\mathrm{d}\|\widetilde{\mathbf{m}}_t - \mathbf{m}^*_t\| \leqslant M'(L, C, \beta, t)\|\widetilde{\mathbf{m}}_t - \mathbf{m}^*_t\|\mathrm{d}t. \tag{44}$$

Thus, the initial argument is proven by applying Gröwnwall inequality to the mapping $t \mapsto \|\widetilde{\mathbf{m}}_t - \mathbf{m}^*_t\|$. □