Optimal Transport Depth Up-Scaling

Anonymous ACL submission

Abstract

Scaling up Large Language Models (LLMs) in parameter size yields remarkable performance gains but incurs exponentially increasing training costs. Depth up-scaling provides an efficient alternative by inserting new layers into a pre-trained LLM, avoiding training from scratch. However, most existing methods copying or averaging base layers for new layer, which misalign functionally corresponding neurons, leading to neuron permutation mismatch that harms performance. To address this issue, we propose Optimal Transport Depth Up-Scaling (OT-DUS), which leverages Optimal Transport (OT) theory to align and fuse functionally corresponding neurons module by module in adjacent base layers for new layer construction. OT-DUS achieves better overall performance in both general and specialized domains than existing methods for continual pre-training and supervised fine-tuning across different model sizes. Furthermore, our systematic analysis of insertion strategies finds that inserting new layers at higher positions yields not only stronger performance but also improved training efficiency, an aspect often neglected [1].

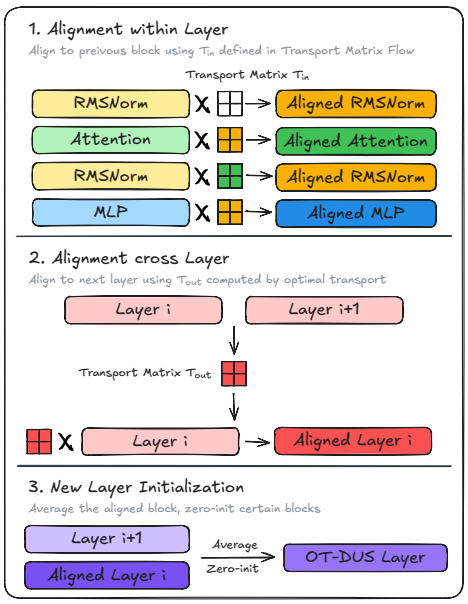

Figure 1: Illustration of our proposed OT-DUS. OT-DUS uses optimal transport to initialize new layers, each derived from two adjacent base layers f_i and f_{i+1}. It first aligns each module b to previous module $b-1$ in f_i (i.e. Alignment within Layer), then aligns it to b in f_{i+1} (i.e. Alignment across Layer).

1 Introduction

Large Language Models (LLMs) performance is largely attributed to scaling laws, where capabilities often improve with increased model and data size (Brown et al., 2020; Kaplan et al., 2020; Wei et al., 2022; Chung et al., 2024). However, scaling poses sustainability challenges, stemming from increased computational (Thompson et al., 2022; Wu et al., 2022; Luccioni et al., 2023) and data (Bender et al., 2021; Birhane et al., 2023; Villalobos et al., 2024) constraints.To address these challenges, model expansion approaches have been proposed, which increase the parameter size of a pre-trained model without altering its original architecture. This can be achieved by increasing the number of neurons per layer (i.e., width up-scaling) or the number of layers (i.e., depth up-scaling).

Depth up-scaling has shown better training efficiency and downstream performance than width up-scaling (Du et al., 2024). Recent depth up-scaling approaches update only the new layers for training efficiency while mitigating catastrophic forgetting (Kim et al., 2024; Yang et al., 2025). Depth up-scaling largely relies on copying (Wu et al., 2024; Kim et al., 2024; Du et al., 2024) or averaging (Yano et al., 2025) for new layer construction. While effective, directly copying or averaging from base layers ignores neuron permutation differences across layers, which can incorrectly

[1] Our code is available at https://anonymous.4open.science/r/OT-DUS.

align neurons, leading to neuron permutation mismatches that harm performance (Li et al., 2015; Yurochkin et al., 2019a,b). An alternative approach (Yang et al., 2025) initializes new layers by training an auxiliary neural network, but it introduces additional computational overhead that increases rapidly with model scale. These challenges motivate our main research question: *How can new layers in depth up-scaling be effectively initialized to mitigate neuron permutation mismatches without substantial additional computation?*

Inspired by applying Optimal Transport (OT) (Singh and Jaggi, 2020; Imfeld et al., 2024), we propose Optimal Transport Depth Up-Scaling (OT-DUS). As shown in Figure 1, OT-DUS aligns and fuses adjacent layers module-wise to create neuron-aligned new layers. Newly added layers are initialized via OT and inserted into the top half of the base model. Certain module weights are set to zero for better neuron alignment and function preservation. Our contributions are as follows:

- We introduce OT-DUS, which creates intermediate layer from adjacent layers by neuron alignment via OT. Experiments show that OT-DUS outperforms existing baselines on both continual pre-training and supervised fine-tuning training stages across various model sizes and diverse tasks.
- OT-DUS achieves the highest overall efficiency, requiring minimal additional computation. Our comprehensive study on layer interpolation position shows that inserting new layers at higher positions leads to higher training efficiency while obtaining better performance.
- Extensive ablation studies validate the contributions of the OT-DUS components, confirming that our design effectively mitigates neuron permutation mismatches. Analysis of neuron functionality reveals a strong correlation between depth up-scaling performance and high transport matrix entropy.

2 Related Work

2.1 Model Expansion

Model expansion accelerates neural network training by expanding a base model to reduce training time and computational overhead (Chen et al., 2016; Wei et al., 2016; Chang et al., 2018; Rusu et al., 2022). Network architecture preservation has proven effective for iterative expansion in encoder-only LLMs (Gong et al., 2019; Yang et al., 2020; Chen et al., 2022). More recently, various model expansion approaches have been explored for decoder-only LLMs. Du et al. (2024) showed depth up-scaling yields greater training efficiency and stronger downstream performance compared to width up-scaling. However, prior work primarily focuses on expansion during the pre-training stage with a relatively large pre-training corpus (Shen et al., 2022; Wang et al., 2023, 2024; Yao et al., 2024; Yano et al., 2025), resulting in high overall computational costs. Limited work focuses on post-training expansion (Kim et al., 2024; Wu et al., 2024; Yang et al., 2025), using a substantially smaller corpus compared to the original pre-training corpus for training efficiency.

2.2 Depth Up-Scaling

Stacking. Stacking methods insert a series of new layers, typically on top by copying layers from the base model (Du et al., 2024; Kim et al., 2024). Du et al. (2024) proposed stacking entire base layers for stronger downstream performance during pre-training. Kim et al. (2024) introduced SOLAR, a partial stacking approach that omits bottom and top base layers for continual pre-training. However, stacking requires updating the entire model, incurring extra computational costs.

Interpolation. Interpolation methods insert new layers inside the base model. Previous work focuses on creating function preservation layers, where the expanded model performs identically to the base model before further training. Achieving function preservation leads to steadier learning processes and better performance. This is achieved by setting the LayerNorm weights to zero for new layer initialization (Shen et al., 2022), initializing the entire new layer to zero (Wang et al., 2024), or employing dynamic masking mechanisms (Yao et al., 2024). Wu et al. (2024) proposed LLaMA PRO, which initializes the inserted new layers by copying weights from the base model. For function preservation, the output matrices of attention and MLP in these new Transformer layers are set to zero, termed zero-init. Yano et al. (2025) initialized new layers by averaging weights from adjacent base layers for pre-training. LESA (Yang et al., 2025) initializes new layers using an auxiliary network given adjacent layers at interpolation positions as input. However, existing methods largely rely on copying (Kim et al., 2024; Wu et al., 2024)

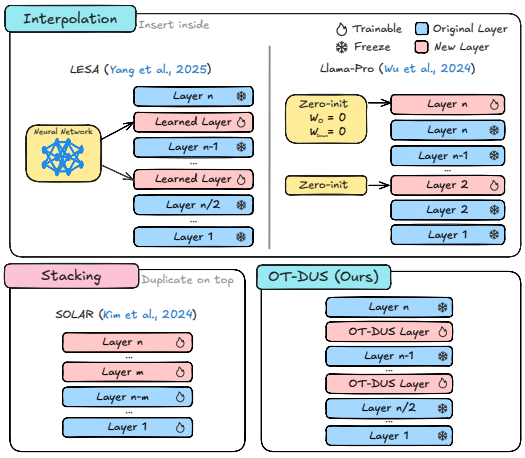

Figure 2: State-of-the-art depth up-scaling methods and our proposed OT-DUS.

or averaging (Yano et al., 2025) to initialize new layers, neglecting neuron permutation differences.

3 Preliminaries

3.1 Depth Up-scaling

Let $\mathcal{M}$ be a *base* LLM with n Transformer layers $\{f_i\}_{i=1}^{n}$, parametrized by θ. The aim is to obtain an *expanded* model $\mathcal{M}'$ with parameters θ' by introducing k additional Transformer layers $\{f'_i\}_{i=1}^{k}$. $\mathcal{M}'$ retains the same layer type and hidden size.

Each Transformer layer f_i is composed of a sequence of modules. We denote the parameters of layer f_i by $\{\mathbf{W}_b^{(i)}\}_{b=1}^{B}$, where $\mathbf{W}_b^{(i)}$ represents the weight matrix of module b in layer f_i. Accordingly, each new layer f'_i is parameterized by $\{\mathbf{W'}_b^{(i)}\}_{b=1}^{B}$.

Stacking. $\mathcal{M}$ is expanded by adding a set of new layers on top of the base layers to obtain $\mathcal{M}'$. $\circ$ denotes the connection between Transformer layers:

$$\mathcal{M}'(x;\theta') = f'_k \circ \cdots \circ f'_1 \circ f_n \circ \cdots \circ f_1(x).$$

Each new layer f'_i is typically initialized by duplicating f_i (Du et al., 2024; Kim et al., 2024). Corresponding to module-wise weight copying:

$$\mathbf{W'}_b^{(i)} \leftarrow \mathbf{W}_b^{(i)}, \quad \forall b \in \{1,\ldots,B\}.$$

Interpolation. Figure 2 illustrates different interpolation strategies adopted by existing depth up-scaling methods. $\mathcal{M}$ is expanded by inserting new layers between base layers as follows:

$$\mathcal{M}'(x;\theta') = \begin{cases} f'_i \circ f_i, & \text{if a new layer is inserted,} \\ f_i, & \text{otherwise.} \end{cases}$$

Interpolation methods initialize f'_i from adjacent base layers f_i and f_{i+1}. This corresponds to initializing each module weight matrix $\mathbf{W'}_b^{(i)}$ using $\mathbf{W}_b^{(i)}$ and $\mathbf{W}_b^{(i+1)}$. Prior work initialize f'_i by copying (Wu et al., 2024; Kim et al., 2024; Du et al., 2024), averaging (Yano et al., 2025), or prediction via an auxiliary network (Yang et al., 2025):

$$\mathbf{W'}_b^{(i)} \leftarrow \begin{cases} \mathbf{W}_b^{(i)}, & \text{copying,} \\ \text{Avg}\left(\mathbf{W}_b^{(i)}, \mathbf{W}_b^{(i+1)}\right), & \text{averaging,} \\ \text{NN}\left(\mathbf{W}_b^{(i)}, \mathbf{W}_b^{(i+1)}\right), & \text{prediction,} \\ \text{OT-DUS}\left(\mathbf{W}_b^{(i)}, \mathbf{W}_b^{(i+1)}\right), & \text{our method,} \end{cases}$$

3.2 Optimal Transport

Optimal Transport is a mathematical framework determining the most cost-effective way to transform one probability distribution into another, given a defined transport cost. Formally, let $\mu = \sum_{i=1}^{n} \alpha_i \delta(x^{(i)})$ and $\nu = \sum_{j=1}^{m} \beta_j \delta(y^{(j)})$ be two discrete probability measures supported on $\{x^{(i)}\}_{i=1}^{n}$ and $\{y^{(j)}\}_{j=1}^{m}$ with probability distribution $\boldsymbol{\alpha} = (\alpha_1,\ldots,\alpha_n)$ and $\boldsymbol{\beta} = (\beta_1,\ldots,\beta_m)$, respectively, where $\delta(s)$ denotes the unit mass at point s. Given a cost matrix $\mathbf{C} \in \mathbb{R}^{n\times m}$ where $\mathbf{C}_{ij}$ is the cost of transporting mass from $x^{(i)}$ to $y^{(j)}$, the Optimal Transport (OT) problem is defined as:

$$\text{OT}(\mu,\nu,\mathbf{C}) = \min_{\mathbf{T}\in\mathbb{R}_+^{n\times m}} \sum_{i,j} \mathbf{T}_{ij}\mathbf{C}_{ij}$$
$$\text{s.t. } \mathbf{T}\mathbf{1}_m = \boldsymbol{\alpha}, \mathbf{T}^\top \mathbf{1}_n = \boldsymbol{\beta}.$$

4 Optimal Transport Depth Up-Scaling

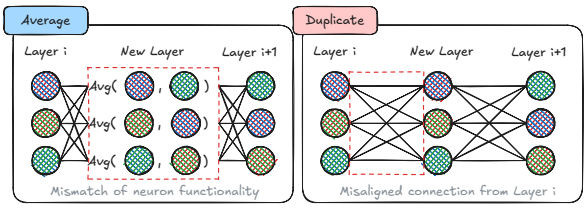

Figure 3: Neuron permutation mismatch caused by average and duplicate. Each column represents the order of neurons within layer. Multiple colours of each neuron represent multiple functions it contributes to. Direct averaging weights for new layer f'_i align neurons with mismatched functionality. Duplicating f_i for initializing f'_i preserves the connection between f'_i and f_{i+1}, while the connection between f_i and f'_i is misaligned.

4.1 Motivation

Copying and averaging from original layers are commonly used methods for creating new layers (Wu et al., 2024; Kim et al., 2024; Du et al.,

2024; Yano et al., 2025). However, such approach neglects neuron permutation mismatch, which is widely present in deep neural networks and Transformers (Li et al., 2015; Yurochkin et al., 2019a,b). During training, a single neuron may contribute to multiple functions (Nguyen et al., 2016), and same-index neurons in different layers may not be functionally corresponding (Sajjad et al., 2022; Klabunde et al., 2025). As shown in Figure 3, averaging from base layers to initialize f_i' can incorrectly merge neurons with different functionalities, while copying from f_i to initialize f_i' breaks the connectivity between f_i and f_i'. Thus, directly copying or averaging can cause misalignment between f_i and f_{i+1}, potentially harming performance (Li et al., 2015; Yurochkin et al., 2019a,b).

Neuron permutation mismatch can be mitigated by aligning neurons between f_i and f_{i+1} using OT, which models functional similarity per neuron. Singh and Jaggi (2020) and Imfeld et al. (2024) show that aligning neurons layer-wise via OT leads to better-initialized new layers for model merging. Recent research further shows that using information from adjacent layers provides stronger initialization than random initialization in depth up-scaling (Du et al., 2024; Yano et al., 2025; Yang et al., 2025). This inspires our Optimal Transport Depth Up-scaling (OT-DUS), illustrated in Figure 1. OT-DUS is a progressive interpolation approach that updates only f'. It aligns and fuses layers f_i and f_{i+1} module by module to create f_i' via OT. OT-DUS inserts new layers f_i' in the top half of $\mathcal{M}$, between base layers f_i and f_{i+1}. This layer interpolation strategy provides better performance (Section 7.1) and training efficiency (Section 7.2).

4.2 Transport Matrix Flow for OT-DUS

OT-DUS relies on two types of transport matrices: $\mathbf{T_{in}}$ and $\mathbf{T}_{\text{out}}$. Each module weight matrix $\mathbf{W}_b^{(i)}$ in f_i' is assigned a $\mathbf{T}_{\text{in}}$. $\mathbf{T}_{\text{in}}$ aligns $\mathbf{W}_b^{(i)}$ to $\mathbf{W}_{b-1}^{(i)}$ within the layer. $\mathbf{T}_{\text{out}}$ aligns $\mathbf{W}_b^{(i)}$ to $\mathbf{W}_b^{(i+1)}$ across layers. $\mathbf{T}_{\text{in}}$ for $\mathbf{W}_b^{(i)}$ is initialized by reusing the $\mathbf{T}_{\text{out}}$ from the previous module $\mathbf{W}_{b-1}^{(i)}$. $\mathbf{T}_{\text{out}}$ is computed by solving an OT problem (Section 4.3).

We use Transport Matrix Flow (TMF) to define the assignment of $\mathbf{T}_{\text{in}}$ for each module (Table 1). Following Imfeld et al. (2024), at the layer entrance of f_i', $\mathbf{T}_{\text{in}}$ is initialized as the identity matrix $\mathbf{I}$. For residual connections (i.e., Pre-MLP Normalization), $\mathbf{T}_{\text{in}}$ is set by averaging the $\mathbf{T}_{\text{out}}$ from both paths (i.e. layer entrance and attention output).

	Normalization		Attention				MLP		
Transport Matrix	Pre-Attn	Pre-MLP	Query	Key	Value	Output	Gate	Up	Down
$\mathbf{T}_{\text{in}}$	$\mathbf{I}$	$\frac{1}{2}(\mathbf{T}_O+\mathbf{I})$	$\mathbf{I}$	$\mathbf{I}$	$\mathbf{I}$	$\underline{\mathbf{I}}$	$\underline{\mathbf{T}_O}$	$\underline{\mathbf{T}_O}$	$\underline{\mathbf{I}}$
$\mathbf{T}_{\text{out}}$	$\mathbf{I}$	$\frac{1}{2}(\mathbf{T}_O+\mathbf{I})$	$\mathbf{T}_Q$	$\mathbf{T}_K$	$\mathbf{T}_V$	$\mathbf{T}_O$	$\mathbf{T}_{\text{Gate}}$	$\mathbf{T}_{\text{Up}}$	$\mathbf{T}_{\text{Down}}$

Table 1: Transport Matrix Flow. We manually set $\mathbf{T}_{\text{in}}$ to each module for alignment within layer. $\mathbf{T}_{\text{out}}$ is calculated through OT for alignment across layers (except Normalization component). Modules highlighted in underline deviate from Imfeld et al. (2024).

However, vanilla TMF is no longer applicable. We therefore modify the TMF design, theses modifications are further ablated in Section 7.3.

Architectural Advancements. Group-Query Attention (Ainslie et al., 2023) and SwiGLU MLP (Shazeer, 2020) introduce dimensional mismatches that makes vanilla TMF inapplicable. We thus set $\mathrm{T}_{\text{in}} = \mathbf{I}$ for Attention Output and MLP Down.

Non-consecutive Alignment. Alignment are applied only to non-consecutive f_i'. $\mathbf{T}_{in}$ at layer entrance does not carry meaningful cross-layer information. Incorporating this will dilute the effective of alignment. Thus, we set $\mathbf{T}_{in} = \mathbf{T}_O$ for MLP Gate and Up that after the residual connection.

4.3 Weight Initialization with OT

Given the parameters of layers f_i and f_{i+1} and the pre-defined TMF, Algorithm 1 demonstrates how to initialize the new layer f_i' via OT. The overall procedure consists of five steps, described below.

Algorithm 1 Optimal Transport Depth Up-Scaling

Input: $\mathbf{W}_b^{(i)}, \mathbf{W}_b^{(i+1)}; \mathcal{TMF} = \{\mathbf{T}_{\text{in}}^{(b)}\}_{b=1}^{B}$ (Table 1)
Output: $\mathbf{W'}_b^{(i)}$
1: **for** base layer f_i ($\frac{n}{2} \leq i < n$) **do**
2: **for** each module b **do**
3: Initialize μ and ν ▷ OT instantiation
4: $\mathbf{C}_{kj} = \|\delta(x^{(k)}) - \delta(y^{(j)})\|_2$ ▷ OT instantiation
5: $\mathbf{T}_{\text{in}} \leftarrow \mathcal{TMF}[b]$ ▷ Retrieve $\mathbf{T}_{\text{in}}$ from $\mathcal{TMF}$
6: $\mathbf{W}_b^{(i)} \leftarrow \mathbf{W}_b^{(i)} \cdot \mathbf{T}_{\text{in}}$ ▷ Alignment within layer
7: $\mathbf{T}_{\text{out}} = \mathrm{OT}(\mu, \nu, \mathbf{C})$ ▷ Solve OT problem
8: $\mathbf{W}_b^{(i)} \leftarrow \mathbf{T}_{\text{out}}^{\top} \cdot \mathbf{W}_b^{(i)}$ ▷ Align across layer
9: $\mathbf{W'}_b^{(i)} \leftarrow \frac{1}{2}(\mathbf{W}_b^{(i)} + \mathbf{W}_b^{(i+1)})$ ▷ New Weights
10: **end for**
11: $\mathbf{W'}_O^{(i)}, \mathbf{W'}_{\text{Down}}^{(i)} \leftarrow \mathbf{0}$ ▷ Zero-init
12: **end for**

Step-1: OT problem instantiation. Based on the definition of OT, we initialize μ and ν over $\mathbf{W}b^{(i)}$ and $\mathbf{W}b^{(i+1)}$. Their probability distributions $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are initialized uniformly, treating each neurons equally (cf. line 3). To measure neuron differences, we adopt the weight-based support function δ (Singh and Jaggi, 2020), representing each neuron directly by its weight value without auxiliary

constraints. The transport cost $\mathbf{C}kj$ is defined as the Euclidean distance between the k-th neuron in $\mathbf{W}b^{(i)}$ and the j-th neuron in $\mathbf{W}_b^{(i+1)}$ (cf. line 4).

Step-2: Alignment within layer. The permutation change caused by aligning $\mathbf{W}_{b-1}^{(i)}$ to $\mathbf{W}_{b-1}^{(i+1)}$ disrupts the original neuron correspondence between $\mathbf{W}_{b-1}^{(i)}$ and $\mathbf{W}_b^{(i)}$. Such permutation change information is stored in $\mathbf{T}_{\text{in}}$ for $\mathbf{W}_b^{(i)}$. To restore this, $\mathbf{W}_b^{(i)}$ needs to align with $\mathbf{W}_{b-1}^{(i)}$ using $\mathbf{T}_{\text{in}}$. $\mathbf{T}_{\text{in}}$ is defined by $\mathcal{TMF}$ for each module, shown in Table 1. After retrieving $\mathbf{T}_{\text{in}}$ (cf. line 5), the alignment within the layer is performed via $\mathbf{W}_b^{(i)} \leftarrow \mathbf{W}_b^{(i)} \cdot \mathbf{T}_{\text{in}}$ (cf. line 6).

Step-3: Alignment across layer We then solve $\text{OT}(\mu, \nu, \mathbf{C})$ to compute the transport matrix. Imfeld et al. (2024) found that the Sinkhorn-Knopp algorithm (Knight, 2008) is optimal for solving $\text{OT}(\mu, \nu, \mathbf{C})$ in Transformer fusion. We employ this approach to obtain $\mathbf{T}_{\text{out}}$ for $\mathbf{W}_b^{(i)}$ (cf. line 7). $\mathbf{W}_b^{(i)}$ is then aligned with $\mathbf{W}_b^{(i+1)}$ using the computed $\mathbf{T}_{\text{out}}$ via $\mathbf{W}_b^{(i)} \leftarrow \mathbf{T}_{\text{out}}^{\top} \cdot \mathbf{W}_b^{(i)}$ (cf. line 8).

Step-4: Compute new-layer weights. $\mathbf{W}'^{(i)}_b$ is the average of aligned $\mathbf{W}_b^{(i)}$ and $\mathbf{W}_b^{(i+1)}$ (cf. line 9).

Step-5: Zero-Init. We set $\mathbf{T}_{\text{in}} = \mathbf{I}$ for $\mathbf{W}_O$ and $\mathbf{W}_{down}$ in TMF due to architectural advancements (Section 4.2), which may cause misalignment problem. Inspired by zero-init (Wu et al., 2024), we set $\mathbf{W}_O = 0$ and $\mathbf{W}_{down} = 0$ (cf. line 11), which naturally resolves this issue while ensuring function preservation, a property crucial for retaining model performance (Wang et al., 2024; Wu et al., 2024).

5 Experimental Setup

5.1 Base Model

Following prior work (Wu et al., 2024; Kim et al., 2024; Yang et al., 2025), we use off-the-shelf 32-layer Llama-3.1-8B (Grattafiori et al., 2024) as our *base* model. We further conducted a smaller-scale experiment using off-the-shelf 16-layer Llama-3.2-1B. Following Yang et al. (2025), we insert a number of new layers equal to 50% of the base layers. The *expanded* model sizes are fixed at 11.5B parameters with 48 layers (adding 16 layers) and 1.72B with 24 layers (adding 8 layers) for all depth up-scaling methods.

5.2 Baselines

We experiment with state-of-the-art depth up-scaling methods, as shown in Figure 2. Hyper-parameter and implementation details are provided in Appendix A and B.

Base. We continue pre-training the *base* model without expansion. All layers are trained.

SOLAR. This method copies the bottom and top m layers from $\mathcal{M}$ to form $\mathcal{M}'$. We choose $m = 24$ and $m = 12$ for 11.5B and 1.72B *expanded* models. All layers are trained in line with Kim et al. (2024).

LLaMA PRO. It divides $\mathcal{M}$ into g groups of m layers. p new layers are created by copying the top-p base layers and inserted on top of each group with zero-init. We use $g = 16$ (11.5B *expanded* models) and $g = 8$ (1.72B *expanded* models); $m = 2$ and $p = 1$ are used throughout. Only f' are trained following Wu et al. (2024).

LESA. This approach uses an auxiliary network to initialize f'_i given f_i and f_{i+1}. LESA inserts f'_i in the top half of $\mathcal{M}$. We insert new layers between f_{16} and f_{32} (11.5B *expanded* models), and between f_8 to f_{16} (1.72B *expanded* models). Only f' are trained as in Yang et al. (2025).

5.3 Training Data

For Continual Pre-Training (CPT), we opt using data of same size as in Yang et al. (2025), published after the base model's knowledge cut-off. We sample 1.5B tokens from the CC-MAIN-2024-51 subset of FineWeb-Edu (Penedo et al., 2024). For supervised fine-tuning (SFT), we choose Alpaca GPT4 (Peng et al., 2023) and update the whole model following Yang et al. (2025).

5.4 Evaluation

Following previous studies (Wu et al., 2024; Yang et al., 2025), we conduct experiments focusing on general knowledge-related tasks. We further conduct extensive experiments on specialized domains: biomedical (Lee et al., 2019; Gu et al., 2021; Luo et al., 2022; Singhal et al., 2023) and legal (Chalkidis et al., 2019; Zheng et al., 2021; Henderson et al., 2022; Niklaus et al., 2024) as they are widely explored within LLMs.

General. We include ARC-Easy (Clark et al., 2018), LogiQA (Liu et al., 2020), Winogrande (Sakaguchi et al., 2021) for *Reasoning*; CSQA (Talmor et al., 2019), BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2020) for *Commonsense and Knowledge*; MMLU (Hendrycks et al., 2021) for *Examination*; and WikiText (Merity et al., 2017) for *Language Modeling*.

Biomedical. Following previous work (Williams et al., 2025), we include MultiMedQA benchmark

	Methods	Perplexity ↓	Zero-shot Performance ↑							
		Wiki-PPL	ARC	LogiQA	Wino	CSQA	BoolQ	PIQA	MMLU	Average
CPT	Base-8B	8.35	79.97	26.88	72.06	65.19	81.83	78.84	58.61	66.20
	SOLAR-11.5B	9.90	79.88	26.88	71.59	57.41	80.70	78.56	54.37	64.20
	LLaMA PRO-11.5B	7.81	81.61	**29.49**	73.72	70.93	81.65	79.98	62.56	68.56
	LESA-11.5B	7.73	**82.07**	27.96	74.11	**72.40**	81.93	80.30	62.63	68.77
	OT-DUS-11.5B (Ours)	**7.73**	**82.07**	27.34	**74.74**	71.91	**82.26**	**80.79**	**62.96**	**68.87**
SFT	Base-8B	8.32	81.10	24.58	72.14	68.30	82.14	79.71	59.17	66.73
	SOLAR-11.5B	9.68	80.68	25.19	71.19	61.18	81.19	79.16	55.03	64.80
	LLaMA PRO-11.5B	7.81	83.33	**27.19**	74.11	72.07	82.26	80.79	62.32	68.87
	LESA-11.5B	**7.72**	**83.84**	26.57	75.53	**73.05**	83.00	80.69	63.57	69.47
	OT-DUS-11.5B (Ours)	7.73	83.80	26.73	**76.09**	**73.05**	**83.36**	**80.85**	**63.84**	**69.67**
CPT	Base-1B	13.68	68.64	21.35	58.48	24.57	62.32	74.97	28.85	48.46
	SOLAR-1.72B	13.87	**68.90**	21.20	59.67	21.21	61.07	74.76	28.58	47.91
	LLaMA PRO-1.72B	12.43	67.26	21.04	**61.96**	34.48	62.91	**75.52**	31.85	50.72
	LESA-1.72B	12.28	66.71	21.20	59.75	41.03	**63.64**	74.76	**33.47**	51.51
	OT-DUS-1.72B (Ours)	**12.19**	67.00	**22.58**	60.77	**43.00**	62.72	75.03	33.02	**52.02**
SFT	Base-1B	13.57	69.87	**22.43**	59.43	26.29	62.81	75.57	29.91	49.47
	SOLAR-1.72B	13.68	**70.41**	22.27	59.27	24.90	60.83	75.84	29.40	48.99
	LLaMA PRO-1.72B	**12.36**	68.14	21.35	60.30	38.08	64.07	**76.12**	30.73	51.26
	LESA-1.72B	12.54	67.76	20.89	59.98	43.73	64.86	75.84	**34.47**	52.51
	OT-DUS-1.72B (Ours)	12.46	68.31	21.51	**60.46**	**44.47**	**65.84**	75.84	33.16	**52.80**

Table 2: CPT on 1.5B tokens and SFT (after CPT) performance of 11.5B and 1.72B *expanded* models.

	Biomedical				Legal		
Methods	MedMCQA	MedQA	Bio-MMLU	PubMedQA	Legal-MMLU	CaseHOLD	ECtHR
Base	48.77	53.26	68.45	76.20	64.49	47.42	56.19
SOLAR	45.09	47.29	57.54	76.20	62.60	41.17	39.11
LLaMA PRO	54.72	59.47	70.88	**77.80**	**68.41**	48.50	60.27
LESA	56.51	59.86	**71.73**	76.00	67.02	51.08	60.89
OT-DUS	**56.63**	**60.25**	71.03	76.20	67.93	**51.83**	**61.26**

Table 3: Domain-specific CPT Performance of 11.5B *expanded* models.

(Singhal et al., 2022), specifically the PubMedQA (Jin et al., 2019), MedQA (Jin et al., 2021), MedMCQA (Pal et al., 2022) tasks, and relevant subsets from MMLU (Hendrycks et al., 2021) (anatomy, clinical knowledge, college medicine, medical genetics, professional medicine, college biology).

Legal. We follow Williams et al. (2025) in using CaseHOLD (Zheng et al., 2021) and ECtHR (Task A) (Chalkidis et al., 2019) from LexGLUE (Chalkidis et al., 2022) and Legal-MMLU, covering jurisprudence, professional law, and international law (Hendrycks et al., 2021).

6 Experimental Results

6.1 General Performance

Table 2 (Top) presents the CPT and SFT results of 11.5B *expanded* models. For CPT, we observe that OT-DUS achieves top performance on six out of eight benchmarks, specifically Wiki-PPL (7.73), ARC (82.07) ,Winogrande (74.74), BoolQ (82.26), PIQA (80.79), MMLU (62.96). This strong performance across various downstream tasks, yielding the highest average score (68.87), highlights the effectiveness. We note that the strong performance continues in SFT. It achieves top performance on Winogrande, CSQA, BoolQ, PIQA, MMLU and second performance on Wiki-PPL, ARC and LogiQA, with the top average score (69.67).

To analyze performance with different training budgets, we save five checkpoints while training the 11.5B *expanded* models (every 20% training steps), shown in Appendix C. We observe that OT-DUS consistently achieves top performance on at least five out of eight benchmarks across all checkpoints.

6.2 Domain Specific Performance

Table 3 presents the performance on biomedical and legal domains. We observe OT-DUS achieves the best overall performance. In biomedical tasks, OT-DUS wins two out of four tasks (i.e. MedMCQA and MedQA), while offering the second best performance on the remaining two. Strong performance is also observed in the legal domain, where OT-DUS wins two out three legal tasks and achieves second-best on the remaining one.

6.3 Performance at Smaller Scales

Table 2 (Bottom) presents the CPT and SFT results of 1.72B *expanded* models. For CPT, OT-DUS achieves the best overall performance (52.02) and ranks first on Wiki-PPL (12.19), LogiQA (22.58), and CSQA (43.00), while ranking second on Winogrande, PIQA, and MMLU. Compared to LESA, the second-best method, OT-DUS obtains the highest average score (52.02 vs. 51.51) and achieves top-2 performance on most downstream tasks (6 vs. 4). For SFT, strong performance can still be

observed with the highest average score. OT-DUS wins on Winogrande, CSQA, and BoolQ, while being second on Wiki-PPL, PIQA and MMLU. Similar to the results of the 11.5B *expanded* models, OT-DUS is the best-performing method using a smaller *base* model. This consistency demonstrate OT-DUS's robustness to model sizes.

6.4 Up-scaling Stability at Larger Scales

We follow previous work (Yano et al., 2025; Yang et al., 2025) by reporting perplexity without any training to evaluate up-scaling stability on larger models. Appendix D presents the perplexity at different model scales. We observe that both LLaMA-Pro and OT-DUS match the base model's perplexity regardless of model parameters due to zero-init, demonstrating maximum expansion stability compared to other baselines. Surprisingly, we find that LESA's perplexity sharply increases when applied to Llama-3.2-1B (871.50). We hypothesize this is because smaller models have fewer layers. This leads to less training data for the auxiliary network, consequently causing it to underfit.

7 Experimental Analysis

7.1 Interpolation Positions

Interpolation Strategy	Average
OT-DUS-Btm	68.32
OT-DUS-Mid	68.29
OT-DUS-Top	**68.87**
OT-DUS-T&B	<u>68.42</u>

Table 4: Average performance of OT-DUS-11.5B using different interpolation strategies. Full results are presented in Table 10 (Appendix F).

We extensively investigate the best interpolation positions of OT-DUS. We evaluate the following strategies: inserting in the bottom half (Btm), in the middle portion (Mid), in the top half (Top), and at the top and bottom quarters (T&B). The layer index ranges are defined as follows:

$$\mathcal{M}'(x;\theta')\circ = \begin{cases} f'_i \circ f_i, & i \leq \frac{n}{2} & \text{if Btm} \\ f'_i \circ f_i, & \frac{n}{4} < i \leq \frac{3n}{4} & \text{if Mid} \\ f'_i \circ f_i, & \frac{n}{2} \leq i < n & \text{if Top} \\ f'_i \circ f_i, & i \leq \frac{n}{4} \text{ or } \frac{3n}{4} \leq i < n & \text{if T\&B} \end{cases}$$

Table 4 illustrates the performance of different interpolation strategies. We observe that OT-DUS-Top is the best performing strategy, overall. OT-DUS-Top yields the highest average performance (68.87), winning in six out of eight benchmarks (i.e. ARC, Winogrande, CSQA, BoolQ, PIQA, MMLU). The performance difference between interpolation strategies is consistent with previous work, where inserting new layers into the top part offers additional performance gains (Yang et al., 2025). This further supports previous findings showing that bottom layers in Transformers are more critical (Jawahar et al., 2019), while top layers are less sensitive to modification (Men et al., 2025).

7.2 Training Efficiency

Methods	Trainable	Training Time
SOLAR-11.5B	11B	22:54:11 (+78.0%)
LLaMA PRO-11.5B	4B	14:58:34 (+16.4%)
LESA-11.5B	4B	12:54:07 (+0.3%)
OT-DUS-Btm-11.5B	4B	14:56:00 (+16.1%)
OT-DUS-Mid-11.5B	4B	13:53:14 (+7.9%)
OT-DUS-Top-11.5B	4B	12:52:04
OT-DUS-T&B-11.5B	4B	14:45:38 (+14.7%)

Table 5: Training time for 11.5B models.

Previous work ignores the impact of interpolation strategy regarding training efficiency (Wu et al., 2024; Yang et al., 2025). Table 5 shows that progressive depth up-scaling methods considerably outperform SOLAR (22:54:11) in training efficiency. We observe a strong correlation between interpolation positions and efficiency: top-half insertions, exemplified by OT-DUS-Top (12:52:04) and LESA (12:54:07), are notably faster. Conversely, strategies inserting layers in the bottom half, such as OT-DUS-Btm (14:56:00) and LLaMA PRO (14:58:34), require longer training time. This pattern holds regardless of the initialization method. The efficiency differences mainly arise from higher backpropagation costs when updating layers inserted at lower positions in the model.

OT-DUS incurs additional computation for solving the OT problem. Compared with baselines that also require extra computation (Table 9, Appendix E), it introduces less overhead and achieves the best overall efficiency among the baselines.

7.3 Ablation Studies

Ablation on TMF design. We modify the vanilla TMF to account for architectural advancements and non-consecutive alignment (Section 4.2). We further compare our design with alternative variants for architectural advancements (Output: $\mathbf{T}_{\text{in}} = \mathbf{T}_Q$) and non-consecutive alignment (Gate/Up: $\mathbf{T}_{\text{in}} = \frac{1}{2}(\mathbf{T}_O + \mathbf{I})$). Table 6 shows that OT-DUS

	Ablation Variants	Average
TMF	Attention Output $\mathbf{T}_{\text{in}} = \mathbf{T}_Q$-1.72B	51.98
	MLP Gate and Up $\mathbf{T}_{\text{in}} = \frac{1}{2}(\mathbf{T}_O + \mathbf{I})$)-1.72B	51.80
OT	Average-1.72B	50.88
	Copy-1.72B	50.12
	Random-1.72B	51.32
Zero-Init	Average + Zero-Init-1.72B	51.91
	Copy + Zero-Init-1.72B	51.67
	Random + Zero-Init-1.72B	51.58
	OT-DUS-1.72B (Ours)	**52.02**
TMF	Attention Output $\mathbf{T}_{\text{in}} = \mathbf{T}_Q$-11.5B	68.73
	MLP Gate and Up $\mathbf{T}_{\text{in}} = \frac{1}{2}(\mathbf{T}_O + \mathbf{I})$)-11.5B	68.71
OT	Average-11.5B	68.39
	Copy-11.5B	68.59
	Random-11.5B	67.14
Zero-Init	Average + Zero-Init-11.5B	68.64
	Copy + Zero-Init-11.5B	68.86
	Random + Zero-Init-11.5B	68.40
	OT-DUS-11.5B (Ours)	**68.87**

Table 6: Average performance of ablation variants. Full Results are presented at Table 11 (Appendix G).

	Attention				MLP			
Methods	Query	Key	Value	Output	Gate	Up	Down	Average
LLaMA PRO	12.23	9.99	10.29	16.64	14.17	14.20	16.63	13.45
Average	11.69	7.32	10.90	15.60	17.06	18.71	8.56	12.84
OT-DUS	13.25	10.32	11.57	16.64	16.78	18.36	16.64	14.79

Table 7: Layer-averaged Entropy of the transport matrix between base and new layers in the 11.5B *expanded* models. Normalization components are excluded as they do not involve cross-neuron mixing.

achieves the best overall performance (Avg: 52.02), validating our TMF design.

Ablation on OT alignment. To validate OT (Section 4.1), we compare it with direct copy, average, and random (Table 6). OT-DUS achieves the best performance, demonstrating that OT alignment provides a more meaningful initialization that mitigates neuron permutation mismatch.

Ablation on Zero-Init. OT-DUS adopts zero-init (Section 4.3). We further compare it with non-OT variants using zero-init (Table 6). OT-DUS consistently outperforms these variants, while zero-init improves performance for non-OT variants. Together with the observed up-scaling stability (Section 6.4), these results support the use of zero-init.

7.4 Analysis of Neuron Functionality

To investigate neuron functionality mapping, we analyze the transport matrix between each new layer and its adjacent base layers. Following Algorithm 1 (Section 4.3), we instantiate the OT problem and solve it with the same hyperparameters, then compute the entropy (Shannon, 1948) of each transport matrix for each module (Table 7). The transport matrix reflects neuron-level functional correspondence, while entropy measures how diffuse the mapping is. Higher entropy indicates a smoother mapping (Cuturi, 2013). SOLAR and LESA are excluded since SOLAR inserts continuous new layers and LESA does not directly use base layers.

We observe a clear correlation between higher transport matrix entropy and stronger performance. OT-DUS achieves the highest entropy (14.79) and the best performance (68.87), followed by LLaMA PRO (13.45/68.56) and Average (12.84/68.39). Higher entropy indicates that information from each neuron is distributed across multiple neurons in the next layer, forming representations through a richer mixture of input features and a more diverse representational space (Tax et al., 2017; Yu et al., 2021), which leads to better performance.

Interestingly, Average yields low entropy for the Key Projection (7.32 vs. 9.99/10.32) and Down Projection (8.56 vs. 16.63/16.64), suggesting these modules are more sensitive to averaging and thus suffer greater performance degradation. Low entropy indicates that functionality from old neurons is concentrated on fewer neurons in the new layer, causing a small subset to dominate the module output. This observation aligns with prior works on the low-rank bottleneck in Query and Key (Bhojanapalli et al., 2020) and parameter redundancy in Down (Pires et al., 2023; Wei et al., 2024).

8 Conclusion

We introduce OT-DUS, a depth up-scaling approach that aligns neurons within and across layers via OT to mitigate neuron permutation mismatches. Empirical results demonstrate that OT-DUS achieves better performance in both general and specialized domains with improved efficiency compared to existing approaches across different model sizes and training stages. Our systematic analysis of interpolation positions reveals their impact on training efficiency, showing that inserting new layers closer to the top leads to higher training efficiency with improved performance. Our neuron functionality analysis further reveals a correlation between strong performance and high transport matrix entropy when initializing new layers.

Limitations

Following recent work (Kim et al., 2024; Wu et al., 2024; Yang et al., 2025), we focus on up-scaling already pre-trained LLMs rather than up-scaling during pre-training as in prior work (Shen et al., 2022; Wang et al., 2023, 2024). OT-DUS mitigates neuron permutation mismatch by aligning neurons with similar functionalities across adjacent base layers, which implicitly assumes that base layers have already learned relatively stable functionalities. This assumption may not hold during intermediate stages of pre-training, when neurons are still learning new features. Moreover, applying up-scaling during pre-training would require tremendous training budgets, which are beyond our current computational resources.

References

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. 2023. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901, Singapore. Association for Computational Linguistics.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Srinadh Bhojanapalli, Chulhee Yun, Ankit Singh Rawat, Sashank Reddi, and Sanjiv Kumar. 2020. Low-rank bottleneck in multi-head attention models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 864–873. PMLR.

Abeba Birhane, vinay prabhu, Sanghyun Han, Vishnu Boddeti, and Sasha Luccioni. 2023. Into the LAION's Den: Investigating Hate in Multimodal Datasets. In *Advances in Neural Information Processing Systems*, volume 36, pages 21268–21284. Curran Associates, Inc.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: Reasoning about Physical Commonsense in Natural Language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2019. Neural legal judgment prediction in English. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4317–4323, Florence, Italy. Association for Computational Linguistics.

Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Katz, and Nikolaos Aletras. 2022. LexGLUE: A benchmark dataset for legal language understanding in English. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4310–4330, Dublin, Ireland. Association for Computational Linguistics.

Bo Chang, Lili Meng, Eldad Haber, Frederick Tung, and David Begert. 2018. Multi-level Residual Networks from Dynamical Systems View. In *International Conference on Learning Representations*.

Cheng Chen, Yichun Yin, Lifeng Shang, Xin Jiang, Yujia Qin, Fengyu Wang, Zhi Wang, Xiao Chen, Zhiyuan Liu, and Qun Liu. 2022. bert2BERT: Towards Reusable Pretrained Language Models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2134–2148, Dublin, Ireland. Association for Computational Linguistics.

Tianqi Chen, Ian Goodfellow, and Jonathon Shlens. 2016. Net2Net: Accelerating Learning via Knowledge Transfer. In *International Conference on Learning Representations*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2024. Scaling Instruction-Finetuned Language models. *Journal of Machine Learning Research*, 25(70):1–53.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions. In *Proceedings of the 2019 Conference of the North American*

Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge. *Preprint*, arXiv:1803.05457.

Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Tri Dao. 2024. FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning. In *The Twelfth International Conference on Learning Representations*.

Wenyu Du, Tongxu Luo, Zihan Qiu, Zeyu Huang, Yikang Shen, Reynold Cheng, Yike Guo, and Jie Fu. 2024. Stacking Your Transformers: A Closer Look at Model Growth for Efficient LLM Pre-Training. In *Advances in Neural Information Processing Systems*, volume 37, pages 10491–10540. Curran Associates, Inc.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. A framework for few-shot language model evaluation.

Linyuan Gong, Di He, Zhuohan Li, Tao Qin, Liwei Wang, and Tieyan Liu. 2019. Efficient Training of BERT by Progressively Stacking. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2337–2346. PMLR.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and et al. 2024. The Llama 3 Herd of Models. *Preprint*, arXiv:2407.21783.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Trans. Comput. Healthcare*, 3(1).

Peter Henderson, Mark Krass, Lucia Zheng, Neel Guha, Christopher D Manning, Dan Jurafsky, and Daniel Ho. 2022. Pile of law: Learning responsible data filtering from the law and a 256gb open-source legal dataset. In *Advances in Neural Information Processing Systems*, volume 35, pages 29217–29234. Curran Associates, Inc.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring Massive Multitask Language Understanding. In *International Conference on Learning Representations*.

Moritz Imfeld, Jacopo Graldi, Marco Giordano, Thomas Hofmann, Sotiris Anagnostidis, and Sidak Pal Singh. 2024. Transformer Fusion with Optimal Transport. In *The Twelfth International Conference on Learning Representations*.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What Does BERT Learn about the Structure of Language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14).

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. PubMedQA: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577, Hong Kong, China. Association for Computational Linguistics.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling Laws for Neural Language Models. *Preprint*, arXiv:2001.08361.

Sanghoon Kim, Dahyun Kim, Chanjun Park, Wonsung Lee, Wonho Song, Yunsu Kim, Hyeonwoo Kim, Yungi Kim, Hyeonju Lee, Jihoo Kim, Changbae Ahn, Seonghoon Yang, Sukyung Lee, Hyunbyung Park, Gyoungjin Gim, Mikyoung Cha, Hwalsuk Lee, and Sunghun Kim. 2024. SOLAR 10.7B: Scaling Large Language Models with Simple yet Effective Depth Up-Scaling. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, pages 23–35, Mexico City, Mexico. Association for Computational Linguistics.

Max Klabunde, Tobias Schumacher, Markus Strohmaier, and Florian Lemmerich. 2025. Similarity of neural network models: A survey of functional and representational measures. *ACM Comput. Surv.*, 57(9).

Philip A. Knight. 2008. The Sinkhorn-Knopp Algorithm: Convergence and Applications. *SIAM Journal on Matrix Analysis and Applications*, 30(1):261–275.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang.

2019. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Yixuan Li, Jason Yosinski, Jeff Clune, Hod Lipson, and John Hopcroft. 2015. Convergent Learning: Do different neural networks learn the same representations? In *Proceedings of the 1st International Workshop on Feature Extraction: Modern Questions and Challenges at NIPS 2015*, volume 44 of *Proceedings of Machine Learning Research*, pages 196–212, Montreal, Canada. PMLR.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. LogiQA: A Challenge Dataset for Machine Reading Comprehension with Logical Reasoning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3622–3628. International Joint Conferences on Artificial Intelligence Organization. Main track.

Alexandra Sasha Luccioni, Sylvain Viguier, and Anne-Laure Ligozat. 2023. Estimating the Carbon Footprint of BLOOM, a 176B Parameter Language Model. *Journal of Machine Learning Research*, 24(253):1–15.

Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. 2022. Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, 23(6):bbac409.

Xin Men, Mingyu Xu, Qingyu Zhang, Qianhao Yuan, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. 2025. ShortGPT: Layers in large language models are more redundant than you expect. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20192–20204, Vienna, Austria. Association for Computational Linguistics.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer Sentinel Mixture Models. In *International Conference on Learning Representations*.

Anh Nguyen, Jason Yosinski, and Jeff Clune. 2016. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. *Preprint*, arXiv:1602.03616.

Joel Niklaus, Veton Matoshi, Matthias Stürmer, Ilias Chalkidis, and Daniel Ho. 2024. MultiLegalPile: A 689GB multilingual legal corpus. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15077–15094, Bangkok, Thailand. Association for Computational Linguistics.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pages 248–260. PMLR.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024. The Fineweb Datasets: Decanting the Web for the Finest Text Data at Scale. In *Advances in Neural Information Processing Systems*, volume 37, pages 30811–30849. Curran Associates, Inc.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *Preprint*, arXiv:2304.03277.

Telmo Pires, António Vilarinho Lopes, Yannick Assogba, and Hendra Setiawan. 2023. One wide feedforward is all you need. In *Proceedings of the Eighth Conference on Machine Translation*, pages 1031–1044, Singapore. Association for Computational Linguistics.

Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. 2022. Progressive Neural Networks. *Preprint*, arXiv:1606.04671.

Hassan Sajjad, Nadir Durrani, and Fahim Dalvi. 2022. Neuron-level interpretation of deep NLP models: A survey. *Transactions of the Association for Computational Linguistics*, 10:1285–1303.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. WinoGrande: an adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106.

C. E. Shannon. 1948. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423.

Noam Shazeer. 2020. Glu variants improve transformer. *Preprint*, arXiv:2002.05202.

Sheng Shen, Pete Walsh, Kurt Keutzer, Jesse Dodge, Matthew Peters, and Iz Beltagy. 2022. Staged Training for Transformer Language Models. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 19893–19908. PMLR.

Sidak Pal Singh and Martin Jaggi. 2020. Model Fusion via Optimal Transport. In *Advances in Neural Information Processing Systems*, volume 33, pages 22045–22055. Curran Associates, Inc.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S. Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, Perry Payne, Martin Seneviratne, Paul Gamble, Chris Kelly, Nathaneal Scharli, Aakanksha Chowdhery, Philip Mansfield, Blaise Aguera y Arcas, Dale Webster, Greg S. Corrado, Yossi Matias, Katherine Chou, Juraj Gottweis, Nenad Tomasev, Yun Liu, Alvin Rajkomar, Joelle Barral, Christopher Semturs,

Alan Karthikesalingam, and Vivek Natarajan. 2022. Large language models encode clinical knowledge. *Preprint*, arXiv:2212.13138.

Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Le Hou, Kevin Clark, Stephen Pfohl, Heather Cole-Lewis, Darlene Neal, Mike Schaekermann, Amy Wang, Mohamed Amin, Sami Lachgar, Philip Mansfield, Sushant Prakash, Bradley Green, Ewa Dominowska, Blaise Aguera y Arcas, Nenad Tomasev, Yun Liu, Renee Wong, Christopher Semturs, S. Sara Mahdavi, Joelle Barral, Dale Webster, Greg S. Corrado, Yossi Matias, Shekoofeh Azizi, Alan Karthikesalingam, and Vivek Natarajan. 2023. Towards expert-level medical question answering with large language models. *Preprint*, arXiv:2305.09617.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Tycho M.S. Tax, Pedro A.M. Mediano, and Murray Shanahan. 2017. The partial information decomposition of generative neural network models. *Entropy*, 19(9).

Neil C. Thompson, Kristjan Greenewald, Keeheon Lee, and Gabriel F. Manso. 2022. The Computational Limits of Deep Learning. *Preprint*, arXiv:2007.05558.

Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. 2024. Will we run out of data? Limits of LLM scaling based on human-generated data. *Preprint*, arXiv:2211.04325.

Peihao Wang, Rameswar Panda, Lucas Torroba Hennigen, Philip Greengard, Leonid Karlinsky, Rogerio Feris, David Daniel Cox, Zhangyang Wang, and Yoon Kim. 2023. Learning to Grow Pretrained Models for Efficient Transformer Training. In *International Conference on Learning Representations*.

Yite Wang, Jiahao Su, Hanlin Lu, Cong Xie, Tianyi Liu, Jianbo Yuan, Haibin Lin, Ruoyu Sun, and Hongxia Yang. 2024. LEMON: Lossless model expansion. In *The Twelfth International Conference on Learning Representations*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent Abilities of Large Language Models. *Transactions on Machine Learning Research*. Survey Certification.

Tao Wei, Changhu Wang, Yong Rui, and Chang Wen Chen. 2016. Network Morphism. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 564–572. PMLR.

Xiuying Wei, Skander Moalla, Razvan Pascanu, and Caglar Gulcehre. 2024. Investigating low-rank training in transformer language models: Efficiency and scaling analysis. *Preprint*, arXiv:2407.09835.

Miles Williams, George Chrysostomou, Vitor Jeronymo, and Nikolaos Aletras. 2025. Compressing language models for specialized domains. *Preprint*, arXiv:2502.18424.

Carole-Jean Wu, Ramya Raghavendra, Udit Gupta, Bilge Acun, Newsha Ardalani, Kiwan Maeng, Gloria Chang, Fiona Aga, Jinshi Huang, Charles Bai, Michael Gschwind, Anurag Gupta, Myle Ott, Anastasia Melnikov, Salvatore Candido, David Brooks, Geeta Chauhan, Benjamin Lee, Hsien-Hsin Lee, Bugra Akyildiz, Maximilian Balandat, Joe Spisak, Ravi Jain, Mike Rabbat, and Kim Hazelwood. 2022. Sustainable AI: Environmental Implications, Challenges and Opportunities. In *Proceedings of Machine Learning and Systems*, volume 4, pages 795–813.

Chengyue Wu, Yukang Gan, Yixiao Ge, Zeyu Lu, Jiahao Wang, Ye Feng, Ying Shan, and Ping Luo. 2024. LLaMA Pro: Progressive LLaMA with Block Expansion. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6518–6537, Bangkok, Thailand. Association for Computational Linguistics.

Cheng Yang, Shengnan Wang, Chao Yang, Yuechuan Li, Ru He, and Jingqiao Zhang. 2020. Progressively Stacking 2.0: A Multi-stage Layerwise Training Method for BERT Training Speedup. *Preprint*, arXiv:2011.13635.

Yifei Yang, Zouying Cao, Xinbei Ma, Yao Yao, Zhi Chen, Libo Qin, and Hai Zhao. 2025. LESA: Learnable LLM layer scaling-up. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22463–22476, Vienna, Austria. Association for Computational Linguistics.

Kazuki Yano, Takumi Ito, and Jun Suzuki. 2025. STEP: Staged Parameter-Efficient Pre-training for Large Language Models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 374–384, Albuquerque, New Mexico. Association for Computational Linguistics.

Yiqun Yao, Zheng Zhang, Jing Li, and Yequan Wang. 2024. Masked Structural Growth for 2x Faster Language Model Pre-training. In *The Twelfth International Conference on Learning Representations*.

Shujian Yu, Kristoffer Wickstrøm, Robert Jenssen, and José C. Príncipe. 2021. Understanding convolutional

neural networks with information theory: An initial exploration. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):435–442.

Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, and Nghia Hoang. 2019a. Statistical Model Aggregation via Parameter Matching. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. 2019b. Bayesian Nonparametric Federated Learning of Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7252–7261. PMLR.

Lucia Zheng, Neel Guha, Brandon R. Anderson, Peter Henderson, and Daniel E. Ho. 2021. When does pretraining help? assessing self-supervised learning for law and the casehold dataset of 53,000+ legal holdings. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, ICAIL '21, page 159–168, New York, NY, USA. Association for Computing Machinery.

A Hyper-parameter Details

We set the regularization parameter of Sinkhorn-Knopp algorithm to 0.06, as in Imfeld et al. (2024). We set the global batch size and sequence length to 64 and 2048. For CPT, we use a maximum learning rate of 1e-4 for 1.72B *expanded* models and 5e-5 for 11.5B *expanded* models. For SFT, the maximum learning rate is set to 1e-5 and 5e-6, respectively.

B Implementation Details

We employ Flash-Attention 2 (Dao, 2024) and mixed-precision `bf16` for accelerated training. We use Language Model Evaluation Harness (Gao et al., 2024) for evaluation. 11.5B *expanded* models are trained on four NVIDIA GH200 (96GB) GPUs while 1.72B *expanded* models are trained on a single NVIDIA A100 (80GB). We create all *expanded* models using AMD EPYC 7413 CPU and a single NVIDIA A100 (80GB).

C Performance across checkpoints

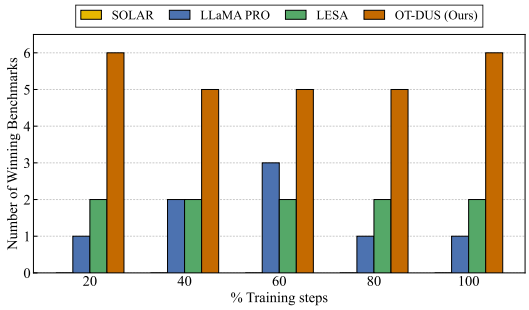

Figure 4: Number of benchmarks that achieve top performance during the training process of 11.5B *expanded* models. Sums may exceed 8 due to ties.

D Scaling Stability

Model	Base	SOLAR	LLaMA PRO	LESA	OT-DUS
Llama-3.2-1B	11.57	16.64	11.57	871.50	11.57
Llama-3.1-8B	7.33	9.01	7.33	9.35	7.33
Mistral-24B	4.43*	6.51*	4.43	5.17*	4.43
Qwen-2.5-32B	3.78*	INF*	3.78	5.67*	3.78
Llama-3-70B	1.98*	4.21*	1.98	2.62*	1.98

Table 8: PPL after 1.5x layer expansion initialization for different *base* models, along with PPL of base models. * denotes results from Yang et al. (2025), Results for LLaMA PRO and OT-DUS in bottom-half are obtained via reasonable extrapolation from top-half.

E Additional Computation Cost

Expanded Model	Creating Time
LESA-1.72B	00:26:15
OT-DUS-1.72B	00:02:34
LESA-11.5B	04:52:13
OT-DUS-11.5B	00:37:16

Table 9: Instantiation time for LESA and OT-DUS.

Both LESA and OT-DUS require additional computation. LESA necessitates extracting latent patterns using Singular Value Decomposition (SVD), while OT-DUS requires solving the OT problem module-wise. Table 9 presents the additional computation time required by LESA and OT-DUS to create the *expanded* model. We observe that LESA requires more time compared to OT-DUS (00:26:15 vs. 00:02:34). This time scales massively with larger models (04:52:13 vs. 00:37:16). We hypothesize that this increased time for LESA is mainly caused by the extra computation required for SVD when scaling up base models. Combining Table 5 and Table 9, our OT-DUS achieves the best time efficiency among the baselines.

F Interpolation Strategy

G Ablation Results

Methods	Perplexity ↓	Zero-shot Performance ↑							
	Wiki-PPL	ARC	LogiQA	Wino	CSQA	BoolQ	PIQA	MMLU	Average
OT-DUS-Btm	7.83	81.69	28.26	74.35	70.02	81.74	79.92	62.28	68.32
OT-DUS-Mid	**7.70**	**82.07**	27.65	74.35	70.11	81.07	80.25	62.56	68.29
OT-DUS-Top	7.73	**82.07**	27.34	**74.74**	**71.91**	**82.26**	**80.79**	**62.96**	**68.87**
OT-DUS-T&B	7.87	81.40	**28.57**	74.51	70.02	82.11	79.87	62.46	68.42

Table 10: Performance of 11.5B OT-DUS trained on 1.5B tokens using different interpolation strategies.

	Methods	Perplexity ↓	Zero-shot Performance ↑							
		Wiki-PPL	ARC	LogiQA	Wino	CSQA	BoolQ	PIQA	MMLU	Average
TMF	Attention Output $\mathbf{T}_{\text{in}} = \mathbf{T}_Q$-1.72B	**12.05**	67.13	**23.35**	60.30	42.67	62.87	75.41	32.10	51.98
TMF	MLP Gate and Up $\mathbf{T}_{\text{in}} = \frac{1}{2}(\mathbf{T}_O + \mathbf{I})$)-1.72B	12.09	66.84	22.89	60.38	42.59	62.23	**75.46**	32.24	51.80
OT	Average-1.72B	12.62	67.72	22.12	59.19	39.23	62.51	74.65	30.72	50.88
OT	Copy-1.72B	12.62	**68.01**	22.73	59.67	34.81	62.97	74.43	28.25	50.12
OT	Random-1.72B	16.40	66.62	22.27	59.67	40.79	62.72	74.97	32.19	51.32
Zero-Init	Average + Zero-Init-1.72B	12.16	66.88	22.43	59.91	42.75	62.91	74.81	**33.73**	51.91
Zero-Init	Copy + Zero-Init-1.72B	12.23	67.13	22.43	59.51	40.87	**64.28**	75.03	32.42	51.67
Zero-Init	Random + Zero-Init-1.72B	12.20	66.92	21.51	59.43	42.92	62.17	74.81	33.32	51.58
	OT-DUS-1.72B (Ours)	12.19	67.00	22.58	**60.77**	**43.00**	62.72	75.03	33.02	**52.02**
TMF	Attention Output $\mathbf{T}_{\text{in}} = \mathbf{T}_Q$-11.5B	**7.47**	81.90	27.19	73.72	**72.40**	82.26	80.69	62.98	68.73
TMF	MLP Gate and Up $\mathbf{T}_{\text{in}} = \frac{1}{2}(\mathbf{T}_O + \mathbf{I})$)-11.5B	**7.47**	81.61	27.19	74.59	71.58	82.32	80.58	**63.11**	68.71
OT	Average-11.5B	7.95	82.15	27.50	73.48	71.09	82.17	80.20	62.11	68.39
OT	Copy-11.5B	7.79	82.20	**27.65**	74.19	71.91	81.62	79.82	62.74	68.59
OT	Random-11.5B	11.69	79.71	27.19	72.53	69.62	80.55	78.62	61.77	67.14
Zero-Init	Average + Zero-Init-11.5B	7.53	82.15	27.04	73.95	72.07	82.08	80.47	62.68	68.64
Zero-Init	Copy + Zero-Init-11.5B	7.67	**82.28**	**27.65**	**74.90**	71.83	82.05	80.47	62.81	68.86
Zero-Init	Random + Zero-Init-11.5B	7.72	81.86	26.42	73.72	71.58	**82.48**	79.98	62.76	68.40
	OT-DUS-11.5B (Ours)	7.73	82.07	27.34	74.74	71.91	82.26	**80.79**	62.96	**68.87**

Table 11: Ablation study on 1.72B and 11.5B *expanded* models.