Optimize Incompatible Parameters Through Compatibility-aware Knowledge Integration

Zheqi Lv[1,3], Keming Ye[1], Zishu Wei[1], Qi Tian[2], Shengyu Zhang[1*], Wenqiao Zhang[1], Wenjie Wang[3*], Kun Kuang[1*], Tat-Seng Chua[3], Fei Wu[1]

[1]Zhejiang University, Hangzhou, China
[2]Tencent TEG., Shenzhen, China
[3] National University of Singapore, Singapore
{zheqilv, kemingye, weizishu, sy_zhang, wenqiaozhang, kunkuang, wufei}@zju.edu.cn,
noaltian@tencent.com, wenjiewang96@gmail.com, dcscts@nus.edu.sg

Abstract

Deep neural networks have become foundational to advancements in multiple domains, including recommendation systems, natural language processing, and so on. Despite their successes, these models often contain incompatible parameters that can be underutilized or detrimental to model performance, particularly when faced with specific, varying data distributions. Existing research excels in removing such parameters or merging the outputs of multiple different pre-trained models. However, the former focuses on efficiency rather than performance, while the latter requires several times more computing and storage resources to support inference. In this paper, we set the goal to explicitly improve these incompatible parameters by leveraging the complementary strengths of different models, thereby directly enhancing the models without any additional parameters. Specifically, we propose **Compatibility-aware Knowledge Integration** (**CKI**), which consists of Parameter Compatibility Assessment and Parameter Splicing, which are used to evaluate the knowledge content of multiple models and integrate the knowledge into one model, respectively. The integrated model can be used directly for inference or for further fine-tuning. Extensive experiments on various recommendation and language datasets show that CKI can effectively optimize incompatible parameters under multiple tasks and settings to break through the training limit of the original model without increasing the inference cost.

Introduction

Deep neural networks have excelled across various domains including natural language processing (Liu et al. 2019; Devlin et al. 2018), recommender systems (Hidasi et al. 2016; Kang and McAuley 2018), etc. The work on the lottery ticket hypothesis (Frankle et al. 2020; Frankle 2023) shows that there exist subnetworks that can be trained in isolation to achieve full accuracy. However, the parameters except these subnetworks, when applied to specific data distributions, often contain parameters that contribute minimally or even negatively to task performance (Wang et al. 2021; Raedt 2022). These parameters can be referred to as "incompatible parameters".

Existing methods to tackle the negative impact of these parameters primarily include: 1) Removing these incompatible parameters (such as pruning and so on) (Raedt 2022; Ma, Fang, and Wang 2023; Guo, Ouyang, and Xu 2020; Zhou et al. 2023; Han et al. 2015; Wang et al. 2020; Dong, Chen, and Pan 2017; Sanh, Wolf, and Rush 2020) to enhance computational efficiency is well-established. However, in dynamic environments, characterized by shifting data distributions (*e.g.*, context and scenarios on language tasks, user preferences on recommendation tasks.), pruning may lose its efficiency. 2) An alternative strategy is training multiple models under different conditions (e.g., with different random seeds and so on) and merging their outputs during inference (such as outputs ensemble (Dong et al. 2020; Zhou et al. 2021)), which increase accuracy by aggregating the outputs of several models trained under different conditions. This approach can mitigate the negative effects of incompatible parameters by leveraging complementary strengths from diverse models. However, this benefit comes at the cost of significantly higher computational demands during inference, and it does not address the core issue of improving the incompatible parameters themselves.

Inspired by the complementary model capabilities in output ensemble, we propose a novel perspective in optimizing incompatible parameters, that is identifying these parameters and integrating complementary counterparts from models trained under diverse conditions for enhancement. Our goal is to establish a unified framework for knowledge integration, applicable to multiple domains including recommendation and language tasks. This integrative approach acknowledges the ubiquitous presence of potentially complementary models in real-world applications. For instance, within large online platforms such as Amazon and Taobao, models dedicated to understanding multi-modal content and user behaviors at different stages, such as initial engagement, search, and post-purchase, can provide mutually reinforcing insights. A straightforward strategy might involve averaging the parameters across models to synthesize a unified model for inference. However, this approach can fall short of expectations, as illustrated in Figure 1.

*Corresponding authors.
Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

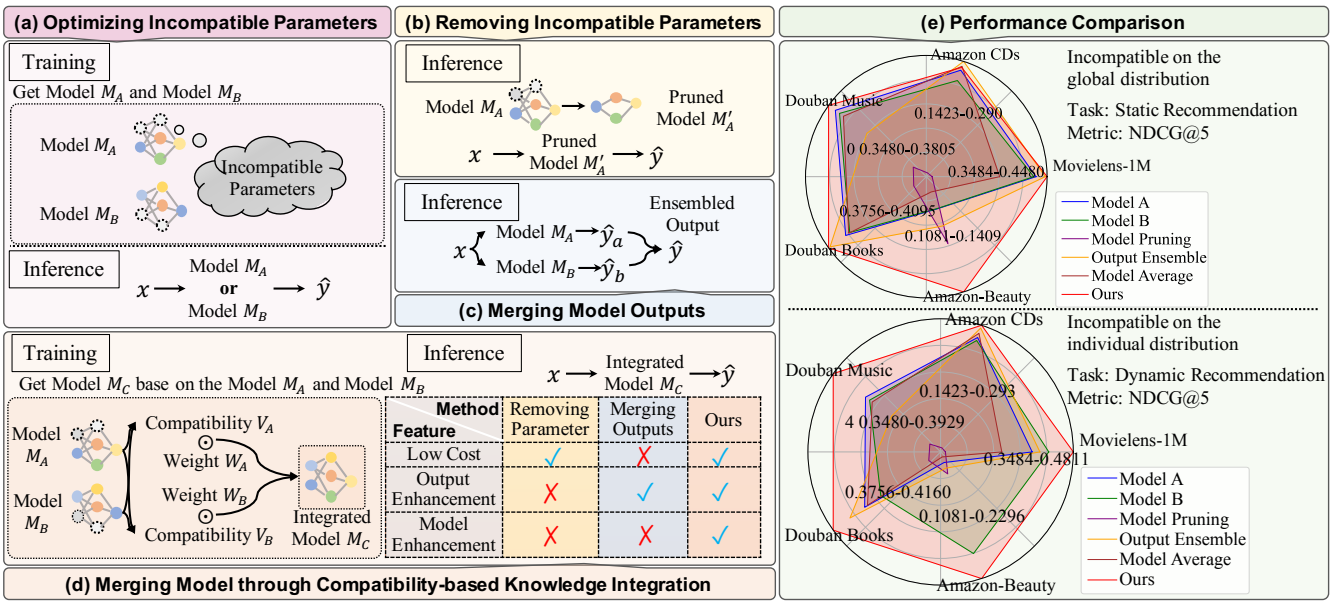

Figure 1: (a) shows the Incompatible Parameter issue. (b) describes Model Pruning, which removes incompatible parameters from M_A. (c) presents output ensemble, which combines the inference results of M_A and M_B for a final result. (d) introduces CKI, which evaluates each parameter's compatibility in global and local views, then integrates the knowledge of M_A and M_B to get model M_C. (e) shows that CKI outperforms baselines in different scenarios.

Specifically, we design a novel Unified Compatibility-aware Knowledge Integration (CKI) algorithm, to address the aforementioned limitations. The core idea of our approach, is to assess the compatibility at each parameter position of each model, and then fuse the parameter across multiple models based on these uncertainties. As shown in Figure 1, CKI comprises the following parts: (1) **Parameter Compatibility Assessment** includes two aspects: local-level parameter uncertainty to assess the uncertainty of each parameter position, and global-level model information content to assess the significance of the entire model. Subsequently, Parameter Compatibility Assessment fuses the local-level model uncertainty and global-level model information content into a comprehensive parameter compatibility. This dual-perspective framework is predicated on the understanding that the fidelity of model parameters is not uniformly distributed across the model's architecture; discrepancies can arise both at the granularity of individual parameters and at the holistic level of the model's overall parameter configuration. This distinction underpins our methodology, which synergistically integrates these two evaluative lenses into a unified parameter compatibility metric. *Local-level Uncertainty Assessment* examines the variance in parameter values across analogous positions within different model instances. This analysis hinges on the premise that the consistency—or lack thereof—in parameter values at a given position serves as an indicator of the reliability and uncertainty of those parameters. For example, minimal variance among models in a specific parameter suggests a convergence towards a stable, and presumably optimal, value, whereas an outlier may signal a deviation from this collective wisdom. *Global-level Information Content Assessment* converts model parameters into histogram distribution, and then further obtains information entropy, which is also the basis for measuring the global-level information content of the model or the global-level information content measurement. We posit that a model's information entropy—a measure of the diversity and richness of information encapsulated within—correlates with the significance of its parameters. A model distinguished by high information entropy is posited to possess a more robust and nuanced parameter set, capable of capturing a broader spectrum of features and nuances within the training data. *Dual-Perspective Compatibility Blend* adopts a holistic evaluation of parameter compatibility that encapsulates both individual parameter efficacy and the collective synergy of the model's parameters. This comprehensive approach enables a more nuanced understanding of model compatibility, facilitating the identification of models that not only perform well in aggregate but are also constructed from high-compatibility components at every level of their architecture. (2) **Parameter Splicing** involves splicing the parameters of multiple models based on the parameter compatibility. The information entropy of multiple models at each position is extracted nonlinear features by neural networks to be used as splicing weights. To splice the parameters, we design hard splicing and soft splicing methods. *Hard Splicing* adopts a decisive approach, selecting for each parameter position the most optimal parameter from the pool of pre-trained models based on compatibility assessment. This method ensures that the spliced model is made up of the best parts available, albeit with the caveat of potential information loss due to the outright ex-

clusion of lower-compatibility parameters. To mitigate this, we introduce *Soft Splicing*, which calculates the weights of the spliced model under the guidance of the weighted sum of parameter compatibility and parameter weights, maximizing the preservation of parameter information and optimizing model parameters.

It should be noted that, the integrated model has the same structure and the same number of parameters as the original pre-trained models, so it does not add even a little bit of additional inference cost. Our CKI algorithm has a very wide range of applications. It can optimize parameters that are incompatible with the global data distribution as well as those that are incompatible with the individual data distribution. The integrated model can be used for inference without any retraining. Furthermore, the integrated model can serve as a better initialization model, requiring only one epoch of retraining to achieve better performance.

In summary, our contributions can be summarized as:

- We focus on optimizing incompatible parameters, which is under-explored but important, with the goal of directly enhancing the model without any additional parameters.
- We instantiated the incompatible parameter optimization method and designed a universal CKI framework consists of Parameter Compatibility Assessment and Parameter Splicing.
- We conducted comprehensive and extensive experiments on recommendation and language tasks. The rich experimental results prove the effectiveness and universality of our method.

Related Work

Incompatible Parameters. Deep learning models often suffer from excessive parameters and redundancies. Network pruning (Guo, Ouyang, and Xu 2020; Wang et al. 2020; Sanh, Wolf, and Rush 2020; Han et al. 2015; Dong, Chen, and Pan 2017; Ma, Fang, and Wang 2023; Zhou et al. 2023; Fang, Ma, and Wang 2024) aims to eliminate non-essential parameters to maintain effectiveness. Evaluation methods include second-order derivatives of the loss function, parameter magnitudes, and other metrics. Output Ensemble (Lee et al. 2021; Zhou et al. 2021; Kumar et al. 2023; Mazari, Boudoukhani, and Djeffal 2024; Yuan, Li, and Hao 2023; Zhu et al. 2024; Wang et al. 2024) aggregates results from multiple models trained under different conditions, optimizing final output without altering model parameters.

Model Aggregation. Model Aggregation is widely used in federated learning (McMahan et al. 2017; Mills, Hu, and Min 2021; Marfoq et al. 2021; Ye et al. 2023; Wu et al. 2023; Li et al. 2023; Ezzeldin et al. 2023; Huang et al. 2023). Typically, models are trained on devices, with parameters or gradients sent to the cloud, then models are aggregated and send back. Federated learning aims to match the performance of a globally trained model under privacy constraints, which is different from our target, its performance upper bound is a global model's performance. Model Grafting (Panigrahi et al. 2023; Matena and Raffel 2022) also supports model aggregation but requires pre-trained models on multiple datasets and has different objectives.

Deep Learning Models. Deep learning models have achieved excellent performance in various tasks. In vision tasks, models such as ResNet (He et al. 2016), SqueezeNet (Iandola et al. 2016), MobileNet (Howard et al. 2017; Sandler et al. 2018; Howard et al. 2019) have significantly improved efficiency. In language tasks, GPT (Radford et al. 2018, 2019; Brown et al. 2020), BERT (Devlin et al. 2018), and Roberta (Liu et al. 2019; Devlin et al. 2018) excel through pre-training and attention mechanisms. In recommendation tasks, models like DeepFM (Guo et al. 2017), LightGCN (He et al. 2020), DIN (Zhou et al. 2018), GRU4Rec (Hidasi et al. 2016), SASRec (Kang and McAuley 2018), Bert4Rec (Sun et al. 2019), and ICLRec (Chen et al. 2022) are often used. Dynamic neural networks are now widely used in computer vision (Alaluf et al. 2022; Dinh et al. 2022), recommendation systems (Lv et al. 2024a; Yan et al. 2022), large language models (Zhang et al. 2024), device-cloud collaboration (Lv et al. 2023, 2024b), and other fields. They can adjust model parameters based on samples, enabling model personalization.

Methodology

Problem Formulation and Notations

First, we need to introduce the notation in deep learning. We use $\mathcal{X} = \{x\}$ to represent a piece of data, and $\mathcal{Y} = \{y\}$ to represent the corresponding label. We represent the dataset as $\mathcal{D}$, where $\mathcal{D} = \{X, Y\}$. More specifically, we use $\mathcal{D}_{\text{Train}}$ to represent the training set and $\mathcal{D}_{\text{Test}}$ to represent the test set. Roughly speaking, let $\mathcal{L}$ be the loss obtained from training on dataset $\mathcal{D}_{\text{Train}}$. Then, the model parameters W can be obtained through the optimization function $\arg\min \mathcal{L}$. The symbol ":=" denotes assigning the value on the right side to the left side. Here, we formalize Model Pruning, Output Ensemble, and CKI to more clearly demonstrate the differences among these three approaches.

Model Pruning. involves trimming the less important model parameters after training and obtaining the model parameters W. There are various ways of pruning, and if we assume M is a mask matrix, these pruning algorithms can be roughly formalized as:

$$W := W \odot M \tag{1}$$

Output Ensemble. assumes that based on the dataset $\mathcal{D}_{\text{Train}}$, n models are trained (where $n \geq 2$). For simplicity, let's initially set n to 2 and assume that the outputs of the two trained models for the same sample x are $\hat{y}_A$ and $\hat{y}_B$, respectively. Then, Output Ensemble can be formalized as:

$$\hat{y} = \alpha \cdot \hat{y}_A + \beta \cdot \hat{y}_B \tag{2}$$

CKI. CKI also assumes that based on the dataset $\mathcal{D}_{\text{Train}}$ like Output Ensemble, n models are trained (where $n \geq 2$). For simplicity, let's initially set n to 2 and assume that the parameters of the two trained models are W_A and W_B. Similar to Model Pruning, let's assume that the uncertainty matrices for W_A and W_B are $V^{(A)}$ and $V^{(B)}$, respectively. Then, CKI can be formalized as,

$$W = W_A \odot V^{(A)} + W_B \odot V^{(B)} \tag{3}$$

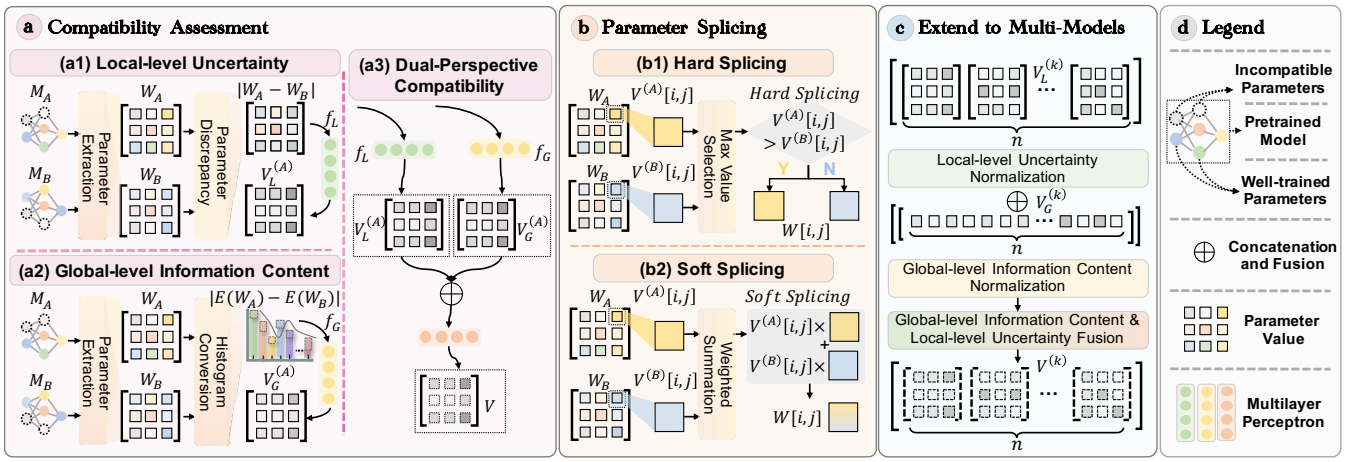

Figure 2: Overview of the proposed CKI. Our CKI includes two parts: Parameter Compatibility Assessment and Parameter Splicing. (a) describes the Parameter Compatibility Assessment. It consists of 3 parts: (a1) Local-level Parameter Uncertainty Assessment, (a2) Global-level Model Information Content Assessment, and (a3) Dual-Perspective Parameter Compatibility Assessment. (b) describes the Parameter Splicing, which includes (b1) Hard Splicing and (b2) Soft Splicing. (c) describes the extension of CKI from 2 models to multiple models.

CKI for Dual Models

In this section, we will introduce our CKI. As shown in Figure 2, our CKI includes two parts: Parameter Compatibility Assessment and Parameter Splicing. (a) Describes the Parameter Compatibility Assessment. It consists of three parts: (a1) describes the Local-level Parameter Uncertainty Assessment method, which is used to measure the uncertainty of each parameter in each model. (a2) describes the Global-level Model Information Content Assessment, which is used to measure the model information content from a global view. (a3) describes the Dual-Perspective Parameter Compatibility Assessment, which assesses model uncertainty from both local-level and global-level perspectives. (b) Describes the Parameter Splicing, which includes two parts: (b1) describes the hard splicing, where low-compatibility parameters at each position directly replace high-compatibility parameters from two models. (b2) describes the soft splicing, where each parameter is weighted and fused based on the compatibility of the parameter at that position in the two models. (c) describes the extension of CKI from 2 models to multiple models. The details are elaborated in the following subsections.

Parameter Compatibility Assessment Here we introduce the methods for assessing parameter compatibility, including Local-level Parameter Uncertainty Assessment, and Global-level Model Information Content Assessment.

Local-level Parameter Uncertainty Assessment. In the Local-level Parameter Uncertainty Assessment, we focus on the uncertainty calculation of each parameter. The basic paradigm is to train a Parameter Uncertainty Assessment network f, inputting the network parameters to directly obtain the uncertainty of the parameters. Assuming the dimension of the parameter matrix $W \in \mathbb{R}^{N_{\text{row}} \times N_{\text{col}}}$ is $N_{\text{row}} \times N_{\text{col}}$, then the dimension of the resulting parameter uncertainty matrix $V \in \mathbb{R}^{N_{\text{row}} \times N_{\text{col}}}$ will also be $N_{\text{row}} \times N_{\text{col}}$. We use ϕ to represent a series of transformations, then the basic paradigm can be formulated as $V = f(\phi(W))$. The difference in parameters of multiple models obtained from the same data is important for judging the uncertainty of the parameters at a position. An intuitive example is, if the parameter values at the same position in two models are equal, then the uncertainty of these two parameters may be considered little. If the parameter values at the same position in two models are not equal, then the uncertainty of these two parameters may be considered large. The uncertainty formula is as follows,

$$V_L^{(A)}[i,j] = f_L(|W_A[i,j] - W_B[i,j]|). \tag{4}$$

$V_L^{(A)}$ represents the local-level parameter uncertainty matrix of the model M_A.

Global-level Model Information Content Assessment. Although Local-level Parameter Uncertainty Assessment can measure the uncertainty of each parameter at its position, it lacks a global perspective. We need to focus on the global perspective to determine which model should be given more credence during model Parameter Splicing, and what the weights should be. Therefore, we further designed Global-level Model Information Content Assessment. This concept primarily draws inspiration from the design philosophies of CNNs and transformers. CNNs use a certain receptive field to continuously extract features, enabling smaller feature maps to have a global perspective. Similarly, transformers require that at each stage, not only local-level features but also global features need to be extracted. However, the global-level information content cannot be directly calculated because the values are too discrete. Therefore, we first obtain the parameter distribution histogram based on the parameter values and then derive the information entropy from the parameter distribution histogram to quantify the information content of a model from a global perspective. We

divide the range between the minimum and maximum values of the model parameters into u equal parts. Then, we can determine the interval for each segment and define the lower bound and upper bound of the t-th according to the following formula.

$$\begin{cases} L_t = W_{\min} + \dfrac{(W_{\max} - W_{\min})(t-1)}{u}, \\ U_t = W_{\min} + \dfrac{(W_{\max} - W_{\min})t}{u}. \end{cases} \tag{5}$$

In the above expression, $W_{\max}$ and $W_{\min}$ represent the maximum and minimum values of all the parameters in the parameter matrix, respectively. Subsequently, we count the number c_t of parameters falling into each segment to get the probability p of the parameters falling within a distribution.

$$c_t = \#\{w \in W \mid w \in [L_t, U_t]\}. \tag{6}$$

From this, we can calculate the information entropy $E(W)$ of the parameters W based on c_t.

$$E(W) = -\sum_{i=1}^{k} \left(\frac{c_t}{N_{\text{row}} \times N_{\text{col}}} \right) \log \left(\frac{c_t}{N_{\text{row}} \times N_{\text{col}}} \right). \tag{7}$$

The global-level information content of the parameters is calculated based on the difference in information entropy.

$$V_G^{(A)} = f_G(|E(W_A) - E(W_B)|). \tag{8}$$

$V_G^{(A)}$ represents the global-level model information content of the model M_A.

Dual-Perspective Parameter Compatibility Assessment. The method of fusing global-level information content and local-level parameter uncertainty is as follows.

$$V^{(A)} = V_G^{(A)} * (1 - \exp(-V_G^{(A)} \times V_L^{(A)})) \tag{9}$$

$$V^{(A)} := \frac{\exp(V^{(A)})}{\exp(V^{(A)}) + \exp(V^{(B)})} \tag{10}$$

The purpose of the above equation is to ensure that $V^{(A)} + V^{(B)} = 1$, in order to avoid the parameters of the integrated model being too large or too small.

Parameter Splicing After the assessment of the parameters' uncertainty, it is necessary to proceed with the splicing of the parameters.

Hard splicing. Hard splicing of parameters involves directly replacing parameters with lower compatibility with those of higher compatibility. W represents the parameters of the spliced model.

$$\begin{aligned} &W = W_A \odot V^{(A)} + W_B \odot V^{(B)}, \\ &\forall v^{(A)} \in V^{(A)}, v^{(A)} \in \{0, 1\}, \\ &\forall v^{(B)} \in V^{(B)}, v^{(B)} \in \{0, 1\}. \end{aligned} \tag{11}$$

Soft splicing. While hard splicing, which selects low-compatibility parameters and discards high-compatibility ones, seems reasonable, it is important to note that low-compatibility parameters may not necessarily be suitable for the final fused model. Therefore, the direct transplantation of parameters could potentially lead to a decrease rather than an increase in the performance of the fused model. Therefore, a softer approach to splicing is worth exploring, namely soft splicing. W represents the parameters of the spliced model.

$$\begin{aligned} &W = W_A \odot V^{(A)} + W_B \odot V^{(B)}, \\ &\forall v^{(A)} \in V^{(A)}, 0 \le v^{(A)} \le 1, \\ &\forall v^{(B)} \in V^{(B)}, 0 \le v^{(B)} \le 1. \end{aligned} \tag{12}$$

Generalization for CKI. In order to better understand the generalization of CKI, we take hard splicing as an example and use classical generalization theory to analyze the generalization boundary of the spliced model. We show that under some assumptions, the generalization boundary of spliced model W can be bounded, see Appendix for more theoretical information.

CKI for Multiple Models

Parameter Compatibility Assessment Here we introduce how to do parameter compatibility assessment based on multiple models.

Local-level Parameter Uncertainty Assessment. When the trained model is no longer just the two models, but rather a collection of n models, the set of the models parameter is $\mathcal{M} = \{M_1, M_2, ..., M_n\}$, the set of the models parameter is $\mathcal{W} = \{W_1, W_2, ..., W_n\}$, the local-level uncertainty of all pairs can be regarded as a $n \times n$ matrix.

$$\begin{bmatrix} V_L^{(1,1)} & V_L^{(1,2)} & \cdots & V_L^{(1,n)} \\ V_L^{(2,1)} & V_L^{(2,2)} & \cdots & V_L^{(2,n)} \\ \vdots & \vdots & \ddots & \vdots \\ V_L^{(n,1)} & V_L^{(n,2)} & \cdots & V_L^{(n,n)} \end{bmatrix} \tag{13}$$

Then, following the formula above, we sum up the uncertainty of each row, and then we can obtain an uncertainty set $\{V_L\}$ of length n. $V_L^{(k)}$ represents the local-level uncertainty of the k-th model, i and j represent the row index and column index of the parameter matrix, respectively.

$$\{V_L^{(k)}[i,j] = \sum_{\substack{W_l \in \mathcal{W} \\ W_l \neq W_k}} f_L(|W_k[i,j] - W_l[i,j]|) \mid W_k \in \mathcal{W}\} \tag{14}$$

Global-level Model Information Content Assessment. Similarly to Equation 13, we obtain the global-level information content matrix. Then, like Equation 14, we get a global-level information content set $\{V_G\}$ of length n. $V_G^{(k)}$ represents the global-level information content of the k-th model. Specifically, it can be formulated as follow,

$$\{V_G^{(k)} = \sum_{\substack{W_l \in \mathcal{W} \\ W_l \neq W_k}} f_G(|E(W_k) - E(W_l)|) \mid W_k \in \mathcal{W}\} \tag{15}$$

Dual-Perspective Parameter Compatibility Assessment. After obtaining V_L and V_G, we fuse the local-level uncertainty V_L and global-level information content V_G for each model M_k according to the following formula to get the model information content matrix $V^{(k)}$.

$$\{V^{(k)} = V_G^{(k)} * (1 - e^{-V_G^{(k)} \times V_L^{(k)}}) \mid M_k \in \mathcal{M}\} \tag{16}$$

Datasets	Methods	Metrics				Datasets	Methods	Metrics			
		NDCG@5	HR@5	NDCG@10	HR@10			NDCG@5	HR@5	NDCG@10	HR@10
Beauty	Model A	0.1137	0.1878	0.1561	0.3203	Music	Model A	0.3779	0.5036	0.4179	0.6273
	Model B	0.1136	0.1888	0.1538	0.3143		Model B	0.3763	0.5034	0.4173	0.6301
	Pruning	0.1250	0.2129	0.1833	0.3936		Pruning	0.3480	0.4727	0.3890	0.5998
	Ensemble	0.1193	0.2038	0.1679	0.3534		Ensemble	0.3658	0.4882	0.4078	0.6181
	Averaging	0.1081	0.1958	0.1682	0.3815		Averaging	0.3748	0.5030	0.4149	0.6269
	Ours	**0.1409**	**0.2420**	**0.1905**	**0.3956**		**Ours**	**0.3805**	**0.5064**	**0.4212**	**0.6326**
Book	Model A	0.4027	0.5260	0.4424	0.6490	Movielens	Model A	0.4384	0.5897	0.4818	0.7208
	Model B	0.4011	0.5255	0.4428	0.6544		Model B	0.4365	0.5901	0.4771	0.7158
	Pruning	0.3756	0.4990	0.4190	0.6337		Pruning	0.3484	0.5014	0.3975	0.6534
	Ensemble	0.4089	0.5322	0.4503	0.6600		Ensemble	0.4464	0.5982	**0.4877**	**0.7257**
	Averaging	0.4015	0.5264	0.4428	0.6538		Averaging	0.4069	0.5674	0.4497	0.6994
	Ours	**0.4095**	**0.5344**	**0.4505**	**0.6612**		**Ours**	**0.4480**	**0.6001**	0.4875	0.7219

Table 1: Performance comparison on recommendation tasks when the pre-trained model is a static models.

Datasets	Methods	Metric			
		Acc	AUC	Recall	F1
SST-2	Model A	0.8429	0.8432	0.8429	0.8429
	Model B	0.8440	0.8444	0.8440	0.8440
	Pruning	0.8177	0.8111	0.8131	0.8106
	Ensemble	0.8532	0.8530	0.8532	0.8532
	Averaging	0.8406	0.8406	0.8406	0.8406
	Ours	**0.8544**	**0.9161**	**0.8853**	**0.8853**
RTE	Model A	0.5884	0.5959	0.5884	0.5815
	Model B	0.5848	0.5854	0.5848	0.5851
	Pruning	0.5271	0.5495	0.5523	0.5513
	Ensemble	0.6065	**0.6091**	0.6065	0.6061
	Averaging	0.5740	0.5837	0.5740	0.5610
	Ours	**0.6245**	0.5628	**0.6679**	**0.6665**

Table 2: Performance comparison on language tasks.

Then we proceed by normalizing the model information content.

$$\{V^{(k)} := \frac{\exp(V^{(k)})}{\sum_{M_l \in \mathcal{M}} \exp(V^{(l)})} \mid M_k \in \mathcal{M}\} \tag{17}$$

The purpose of the above equation is to ensure that $V^{(A)} + V^{(B)} = 1$, in order to avoid the parameters of the integrated model being too large or too small.

Parameter Splicing The Parameter Splicing methods also include two types: hard splicing and soft splicing. In terms of Parameter Splicing, we extend the formulas based on Equations 11 and 12 to a broader range of applications.

$$W = \sum_{W_k \in \mathcal{W}} W_k \odot V^{(k)}, \forall v^{(k)} \in V^{(k)}, 0 \leq V^{(k)} \leq 1. \tag{18}$$

Experiments

Experimental Setup

Datasets In recommendation tasks, we evaluated our method on 4 widely used datasets `Amazon-Beauty (Beauty)`, `Douban-Book (Book)`, `Douban-Music (Music)`, `Movielens-1M (Movielens)`. In language tasks, we evaluated our method on 2 widely used datasets `RTE`, and `SST-2`. Since the usage of the aforementioned language datasets is relatively conventional and straightforward, we will not provide an extensive introduction to the preprocessing of these datasets here. Due to the more complex data distribution in recommendation tasks, the distribution shift of the data is more pronounced and rapid. Therefore, we prefer to conduct experiments on recommendation tasks.

Baselines To evaluate the effectiveness of our method, we selected baselines from the following categories: **Static Recommendation Models.** *DIN* (Zhou et al. 2018), *GRU4Rec* (Hidasi et al. 2016), and *SASRec* (Kang and McAuley 2018) are three widely used sequential recommendation methods. **Dynamic Recommendation Framework.** *DUET* (Lv et al. 2023) is a framework that can generate model parameters for the static model during inference based on the sample. **Language Models.** *Roberta* (Liu et al. 2019) is a widely used language method, which is improved based on *Bert* (Devlin et al. 2018). **Methods to address incompatible parameter issue.** *Model Pruning* (Guo, Ouyang, and Xu 2020; Han et al. 2015) and *Output Ensemble* (Zhou et al. 2021) address the incompatible parameter issue to some extent by cutting off unimportant connections in the neural network and integrating inference results, respectively. **Other model fusion methods.** *Parameter Average* averages parameters of multiple pretrained models.

Evaluation Metrics In the experiments, we use the widely adopted *Acc* (Accuracy), *AUC*, *Recall*, *F1* for language tasks, and use widely adopted *AUC*, *UAUC*, *NDCG*, and *HR* (HitRate) as the metrics to evaluate model performance.

Experimental Results

To facilitate performance comparison, the **best** value is highlighted in bold, while the second best value is underlined. Some tables only highlight the maximum value.

CKI for Global Incompatible Parameters Table 1 and 2 summarize the comparison of our method and other methods to address incompatible parameter issue based on static rec-

Datasets	Methods	Metrics				Datasets	Methods	Metrics			
		NDCG@5	HR@5	NDCG@10	HR@10			NDCG@5	HR@5	NDCG@10	HR@10
Beauty	Model A	0.1137	0.1878	0.1561	0.3203	Music	Model A	0.3779	0.5036	0.4179	0.6273
	Model B (G)	0.2046	0.2982	0.2354	0.3926		Model B (G)	0.3760	0.5082	0.4173	0.6356
	Pruning	0.1370	0.2229	0.1862	0.3775		Pruning	0.3150	0.4464	0.3614	0.5901
	Ensemble	0.1901	0.2751	0.2181	0.3604		Ensemble	0.3728	0.5029	0.4136	0.6292
	Averaging	0.1613	0.2620	0.2064	0.4016		Averaging	0.3747	0.5008	0.4159	0.6278
	Ours	**0.2296**	**0.3183**	**0.2573**	**0.4046**		Ours	**0.3929**	**0.5201**	**0.4332**	**0.6444**
Book	Model A	0.4027	0.5260	0.4424	0.6490	Movielens	Model A	0.4384	0.5897	0.4818	0.7228
	Model B (G)	0.3959	0.5201	0.4374	0.6484		Model B (G)	0.4556	0.6205	0.4964	0.7463
	Pruning	0.3463	0.4739	0.3904	0.6101		Pruning	0.3335	0.4963	0.3857	0.6580
	Ensemble	0.4078	0.5315	0.4472	0.6536		Ensemble	0.4665	0.6215	0.5038	0.7361
	Averaging	0.4080	0.5313	0.4487	0.6571		Averaging	0.4680	0.6273	0.5072	0.7480
	Ours	**0.4160**	**0.5436**	**0.4574**	**0.6716**		Ours	**0.4811**	**0.6315**	**0.5211**	**0.7535**

Table 3: Performance comparison on recommendation task when pre-trained models include both static and dynamic models.

ommendation models. The 1-st and 2-nd rows, Model M_A and Model M_B, are pre-trained models with different initial conditions. The 3-rd row, Pruning, is a model pruning based on Model M_A. The $4-$th row is to ensemble the outputs by the Model M_A and Model M_B. The $5-$th row is to average the parameters of the Model M_A and Model M_B. The $6-$th row is our method, which can do knowledge integration based on the compatibility. To further validate the effectiveness of our method, we compared it with other model fusion methods and output fusion methods. From this table, we have the following conclusions: (1) Pruning often performs worse than the original model because the purpose of pruning is not to enhance the capability of incompatible parameters but to lighten the model. In the process of removing incompatible parameters, it sometimes also cuts well-trained parameters. (2) In some cases, our method achieves significant improvements compared to all these baselines. It is worth noting that the inference cost of the output ensemble is n times that of our method. The reason why result ensemble has an Inference cost of $n\times$ is because it requires multiple models (in this case, $\#\text{models} = 2$) to perform inference separately and then integrate the results. Nevertheless, our method achieves better performance than all baselines in almost all cases. Experimental results show that our method significantly improves model performance with the same resource consumption. This indicates that our approach is effective in optimizing parameters that are incompatible with the global data distribution.

CKI for Individual Incompatible Parameters Table 3 compares the performance of parameter fusion based on dynamic recommendation models (model M_B (G)). Unlike the parameter fusion methods for static models, the parameters of M_B (G) are dynamically generated by a hypernetwork based on the input samples. Both dynamic and static models use the same base model for parameter fusion. Across all datasets and metrics, parameter fusion significantly outperforms other baselines. Additionally, Comparing Table 1 and Table 3, we find that parameter fusion between dynamic and static models achieves a more significant performance improvement over baseline models than parameter fusion between static models. This is because the parameters of dynamic models are adaptively generated in an unsupervised manner based on input samples. Although the performance of dynamic models significantly surpasses that of static models, they also exhibit greater instability. This instability can be compensated by incorporating parameters from static models. Experimental results demonstrate that this method is highly effective in optimizing parameters that are incompatible with individual data distributions.

Models	Metrics			
	NDCG@5	HR@5	NDCG@10	HR@10
Base	0.3779	0.5036	0.4179	0.6273
Ours	0.3823	0.5071	0.4221	0.6302
Ours (F)	**0.3831**	**0.5091**	**0.4223**	**0.6304**

Table 4: Compatibility-aware Knowledge Integration for model initialization and re-training.

CKI for Initialization and Re-training After the model parameters are integrated, the resulting model can be directly used for inference, or it can serve as a better initial parameter set, allowing the model to be further trained on this basis for improved results. To verify the initial performance of the model after CKI, we trained the model again post-integrating. As shown in Table 4, We found that the integrated model only needs one epoch to converge and achieve better results than the parameter integrated model and other baselines. This also demonstrates the benefits of CKI for model initialization.

Conclusion

We introduced CKI, a novel method to optimize incompatible parameters in deep neural networks by leveraging the complementary strengths of different pretrained models, without adding extra parameters. Our experiments on recommendation and language tasks demonstrate that CKI effectively enhances model performance without increasing inference costs, offering a promising approach for deploying more robust and efficient models.

Acknowledgments

This work was supported by 2030 National Science and Technology Major Project (2022ZD0119100), Scientific Research Fund of Zhejiang Provincial Education Department (Y202353679), National Natural Science Foundation of China (No. 62402429, 62376243, 62441605, 62037001), the Key Research and Development Program of Zhejiang Province (No. 2024C03270), ZJU Kunpeng&Ascend Center of Excellence, the Key Research and Development Projects in Zhejiang Province (No.2024C01106), Zhejiang University Education Foundation Qizhen Scholar Foundation, the Starry Night Science Fund at Shanghai Institute for Advanced Study (Zhejiang University). This work was also supported by Ant group.

References

Alaluf, Y.; Tov, O.; Mokady, R.; Gal, R.; and Bermano, A. 2022. Hyperstyle: Stylegan inversion with hypernetworks for real image editing. In *Proceedings of the IEEE/CVF conference on computer Vision and pattern recognition*, 18511–18521.

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

Chen, Y.; Liu, Z.; Li, J.; McAuley, J.; and Xiong, C. 2022. Intent contrastive learning for sequential recommendation. In *Proceedings of the ACM Web Conference 2022*, 2172–2182.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dinh, T. M.; Tran, A. T.; Nguyen, R.; and Hua, B.-S. 2022. Hyperinverter: Improving stylegan inversion via hypernetwork. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11389–11398.

Dong, X.; Chen, S.; and Pan, S. 2017. Learning to prune deep neural networks via layer-wise optimal brain surgeon. *Advances in neural information processing systems*, 30.

Dong, X.; Yu, Z.; Cao, W.; Shi, Y.; and Ma, Q. 2020. A survey on ensemble learning. *Frontiers of Computer Science*, 14: 241–258.

Ezzeldin, Y. H.; Yan, S.; He, C.; Ferrara, E.; and Avestimehr, A. S. 2023. Fairfed: Enabling group fairness in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 7494–7502.

Fang, G.; Ma, X.; and Wang, X. 2024. Structural pruning for diffusion models. *Advances in neural information processing systems*, 36.

Frankle, J. 2023. *The Lottery Ticket Hypothesis: On Sparse, Trainable Neural Networks*. Ph.D. thesis, Massachusetts Institute of Technology.

Frankle, J.; Dziugaite, G. K.; Roy, D.; and Carbin, M. 2020. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, 3259–3269. PMLR.

Guo, H.; Tang, R.; Ye, Y.; Li, Z.; and He, X. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *International Joint Conference on Artificial Intelligence*.

Guo, J.; Ouyang, W.; and Xu, D. 2020. Multi-dimensional pruning: A unified framework for model compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1508–1517.

Han, S.; Pool, J.; Tran, J.; and Dally, W. 2015. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28.

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

He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; and Wang, M. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, 639–648.

Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; and Tikk, D. 2016. Session-based recommendations with recurrent neural networks. *International Conference on Learning Representations 2016*.

Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. 2019. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1314–1324.

Howard, A. G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; and Adam, H. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *CoRR*, abs/1704.04861.

Huang, W.; Ye, M.; Shi, Z.; Li, H.; and Du, B. 2023. Rethinking federated learning with domain shift: A prototype view. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 16312–16322. IEEE.

Iandola, F. N.; Han, S.; Moskewicz, M. W.; Ashraf, K.; Dally, W. J.; and Keutzer, K. 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and¡ 0.5 MB model size. *arXiv preprint arXiv:1602.07360*.

Kang, W.-C.; and McAuley, J. 2018. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, 197–206. IEEE.

Kumar, J.; Gupta, R.; Saxena, D.; and Singh, A. K. 2023. Power consumption forecast model using ensemble learning for smart grid. *The Journal of Supercomputing*, 79(10): 11007–11028.

Lee, K.; Laskin, M.; Srinivas, A.; and Abbeel, P. 2021. Sunrise: A simple unified framework for ensemble learning in deep reinforcement learning. In *International Conference on Machine Learning*, 6131–6141. PMLR.

Li, Z.; Lin, T.; Shang, X.; and Wu, C. 2023. Revisiting weighted aggregation in federated learning with neural networks. In *International Conference on Machine Learning*, 19767–19788. PMLR.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Lv, Z.; He, S.; Zhan, T.; Zhang, S.; Zhang, W.; Chen, J.; Zhao, Z.; and Wu, F. 2024a. Semantic Codebook Learning for Dynamic Recommendation Models. In *ACM Multimedia*, 9611–9620. ACM.

Lv, Z.; Zhang, W.; Chen, Z.; Zhang, S.; and Kuang, K. 2024b. Intelligent Model Update Strategy for Sequential Recommendation. In *WWW*, 3117–3128. ACM.

Lv, Z.; Zhang, W.; Zhang, S.; Kuang, K.; Wang, F.; Wang, Y.; Chen, Z.; Shen, T.; Yang, H.; Ooi, B. C.; and Wu, F. 2023. DUET: A Tuning-Free Device-Cloud Collaborative Parameters Generation Framework for Efficient Device Model Generalization. In *Proceedings of the ACM Web Conference 2023*.

Ma, X.; Fang, G.; and Wang, X. 2023. Llm-pruner: On the structural pruning of large language models. *Advances in neural information processing systems*, 36: 21702–21720.

Marfoq, O.; Neglia, G.; Bellet, A.; Kameni, L.; and Vidal, R. 2021. Federated multi-task learning under a mixture of distributions. *Advances in Neural Information Processing Systems*, 34: 15434–15447.

Matena, M.; and Raffel, C. 2022. Merging Models with Fisher-Weighted Averaging. In *NeurIPS*.

Mazari, A. C.; Boudoukhani, N.; and Djeffal, A. 2024. BERT-based ensemble learning for multi-aspect hate speech detection. *Cluster Computing*, 27(1): 325–339.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, 1273–1282. PMLR.

Mills, J.; Hu, J.; and Min, G. 2021. Multi-task federated learning for personalised deep neural networks in edge computing. *IEEE Transactions on Parallel and Distributed Systems*, 33(3): 630–641.

Panigrahi, A.; Saunshi, N.; Zhao, H.; and Arora, S. 2023. Task-specific skill localization in fine-tuned language models. In *International Conference on Machine Learning*, 27011–27033. PMLR.

Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I.; et al. 2018. Improving language understanding by generative pre-training.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Raedt, L. D., ed. 2022. *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*. ijcai.org.

Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; and Chen, L.-C. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4510–4520.

Sanh, V.; Wolf, T.; and Rush, A. 2020. Movement pruning: Adaptive sparsity by fine-tuning. *Advances in Neural Information Processing Systems*, 33: 20378–20389.

Sun, F.; Liu, J.; Wu, J.; Pei, C.; Lin, X.; Ou, W.; and Jiang, P. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, 1441–1450.

Wang, H.; Qin, C.; Bai, Y.; Zhang, Y.; and Fu, Y. 2021. Recent advances on neural network pruning at initialization. *arXiv preprint arXiv:2103.06460*.

Wang, Y.; Zhang, X.; Xie, L.; Zhou, J.; Su, H.; Zhang, B.; and Hu, X. 2020. Pruning from scratch. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 12273–12280.

Wang, Y.; Zhu, Y.; Zhang, W.; Zhuang, Y.; Liyunfei, L.; and Tang, S. 2024. Bridging Local Details and Global Context in Text-Attributed Graphs. In *Proc. of EMNLP*, 14830–14841.

Wu, Y.; Zhang, S.; Yu, W.; Liu, Y.; Gu, Q.; Zhou, D.; Chen, H.; and Cheng, W. 2023. Personalized federated learning under mixture of distributions. In *International Conference on Machine Learning*, 37860–37879. PMLR.

Yan, B.; Wang, P.; Zhang, K.; Li, F.; Deng, H.; Xu, J.; and Zheng, B. 2022. Apg: Adaptive parameter generation network for click-through rate prediction. *Advances in Neural Information Processing Systems*, 35: 24740–24752.

Ye, R.; Xu, M.; Wang, J.; Xu, C.; Chen, S.; and Wang, Y. 2023. Feddisco: Federated learning with discrepancy-aware collaboration. In *International Conference on Machine Learning*, 39879–39902. PMLR.

Yuan, J.; Li, J.; and Hao, J. 2023. A dynamic clustering ensemble learning approach for crude oil price forecasting. *Engineering Applications of Artificial Intelligence*, 123: 106408.

Zhang, W.; Lin, T.; Liu, J.; Shu, F.; Li, H.; Zhang, L.; Wanggui, H.; Zhou, H.; Lv, Z.; Jiang, H.; et al. 2024. HyperLLaVA: Dynamic Visual and Language Expert Tuning for Multimodal Large Language Models. *arXiv preprint arXiv:2403.13447*.

Zhou, G.; Zhu, X.; Song, C.; Fan, Y.; Zhu, H.; Ma, X.; Yan, Y.; Jin, J.; Li, H.; and Gai, K. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1059–1068.

Zhou, K.; Yang, Y.; Qiao, Y.; and Xiang, T. 2021. Domain adaptive ensemble learning. *IEEE Transactions on Image Processing*, 30: 8008–8018.

Zhou, Y.; Yang, Y.; Chang, A.; and Mahoney, M. W. 2023. A three-regime model of network pruning. In *International Conference on Machine Learning*, 42790–42809. PMLR.

Zhu, Y.; Wang, Y.; Shi, H.; and Tang, S. 2024. Efficient Tuning and Inference for Large Language Models on Textual Graphs. In Larson, K., ed., *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, 5734–5742. International Joint Conferences on Artificial Intelligence Organization. Main Track.