Factorizing Knowledge in Neural Networks

Xingyi Yang, Jingwen Ye, and Xinchao Wang(✉)

National University of Singapore, Singapore, Singapore
xyang@u.nus.edu, {jingweny,xinchao}@nus.edu.sg

Abstract. In this paper, we explore a novel and ambitious knowledge-transfer task, termed Knowledge Factorization (KF). The core idea of KF lies in the modularization and assemblability of knowledge: given a pretrained network model as input, KF aims to decompose it into several factor networks, each of which handles only a dedicated task and maintains task-specific knowledge factorized from the source network. Such factor networks are task-wise disentangled and can be directly assembled, without any fine-tuning, to produce the more competent combined-task networks. In other words, the factor networks serve as Lego-brick-like building blocks, allowing us to construct customized networks in a plug-and-play manner. Specifically, each factor network comprises two modules, a common-knowledge module that is task-agnostic and shared by all factor networks, alongside with a task-specific module dedicated to the factor network itself. We introduce an information-theoretic objective, InfoMax-Bottleneck (IMB), to carry out KF by optimizing the mutual information between the learned representations and input. Experiments across various benchmarks demonstrate that, the derived factor networks yield gratifying performances on not only the dedicated tasks but also disentanglement, while enjoying much better interpretability and modularity. Moreover, the learned common-knowledge representations give rise to impressive results on transfer learning. Our code is available at https://github.com/Adamdad/KnowledgeFactor.

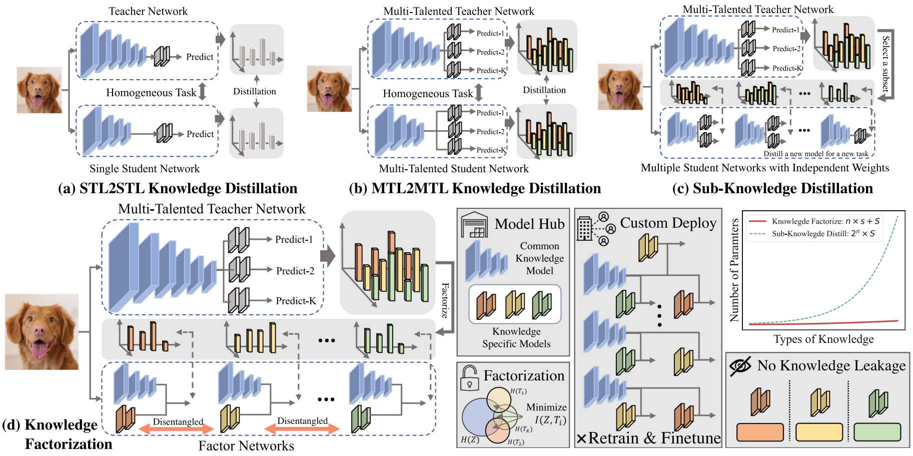
Keywords: Transfer learning · Knowledge factorization

1 Introduction

Over the past decade, deep neural networks (DNNs) have evolved to the *de facto* a standard approach for most if not all computer vision tasks, yielding unprecedentedly promising results. Due to the time- and resource-consuming DNN training process, many developers have generously released their pretrained models online, so that users may adopt these models in a plug-and-play manner

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-19830-4_5.

S. Avidan et al. (Eds.): ECCV 2022, LNCS 13694, pp. 73–91, 2022.
https://doi.org/10.1007/978-3-031-19830-4_5



(a) STL2STL Knowledge Distillation

(b) MTL2MTL Knowledge Distillation

(c) Sub-Knowledge Distillation

(d) Knowledge Factorization

Fig. 1. Illustration of (top) 3 types of Knowledge Distillation and (bottom) our proposed Knowledge Factorization. (a) Single-Task Learning to Single-Task Learning (STL2STL) KD refers to distill a single-tasked student from a single-tasked teacher, (b) Multi-Task Learning to Multi-Task Learning (MTL2MTL) KD stands for distilling a multi-tasked student from a multi-tasked teacher and (c) Sub-Knowledge Distillation distill a subset of the teacher's knowledge to its student model.

without training from scratch. Nevertheless, pretrained DNNs often come with heavy architectures, making them extremely cumbersome to be deployed in real-world scenarios, especially resource-critical applications such as edge computing. Numerous endeavors have thus been made towards reducing the sizes of DNNs, among which one mainstream scheme is known as Knowledge Distillation (KD). The goal of KD is to "distill" knowledge from a large pre-trained model known as a teacher, to a compact model known as a student. The derived student is expected to master the expertise of the teacher yet come with a much smaller size, making it applicable to edge devices. Since the seminal work of [20], a series of KD approaches have been proposed to strengthen the performances of student models [47,51,66].

Albeit encouraging results achieved, KD has largely been treated as a black-box procedure, in which the intrinsic knowledge flow process remains opaque. Consequently, the derived student model may inherit the teacher's task-wise competence but unfortunately lacks interpretability, since it is unclear how and what knowledge has been transferred to the student. In addition, as demonstrated in Fig. 1(a) and (b), conventional KD assumes that teacher and student models master homogeneous tasks or knowledge, which greatly limits its wide applications. Even if it is allowed to distill a subset of knowledge from the teacher, shown in Fig. 1(c), the problem setup of KD, by nature, overlooks the scalability of the student. For example, given a versatile classification teacher pretrained on ImageNet, if we are to learn two students, one handling cat-dog classification and one handling cat-fish, we will have to carry out the KD twice; if, however, we are to learn all k-class classification students from a pool of $1,000$ classes, we will have to conduct KD for $\sum_{k=1}^{1000}\binom{1,000}{k}=2^{1000}$ times, which is computational intractable.

In this paper, we introduce a novel task, termed Knowledge Factorization (KF), that alleviates the aforementioned flaws of KD at a problem-setup level. The core idea of KF regards the modularization and assemblability of knowledge: given a pretrained teacher, KF decomposes it into several *factor networks*, each of which masters one specific knowledge factorized from the teacher, while remaining disentangled with respect to others. Moreover, these factor networks are expected to be readily integratable, meaning that we may directly assemble multiple factor networks, without any fine-tuning, to produce a more competent multi-talented network. As shown in Fig. 1(d), those factor networks can be organized into a open-sourced model hub. At the same time, users could treat them as Lego-brick-like units of knowledge to build customized networks in a plug-and-play fashion, thereby lending itself to great scalability. Furthermore, the disentanglement property effectively enables the IP protection of network knowledge: since the factor networks are learned in a disentangled manner, they possess only task-specific knowledge, allowing the network owners to selectively conduct knowledge transfer without leaking knowledge of other tasks.

Admittedly, the aims of KF are unarguably ambitious, since the factor networks are, again, expected to be modularized and readily integratable, and meanwhile knowledge-wise disentangled and hence more interpretable. Notably, despite orthogonal in expertise, these factor networks will inherit the common knowledge shared by all tasks. As such, each factor network should be designed to account for both the task-agnostic commonality and its task-relevant specialization, which in turn reduces the overall parameter overhead for KF. As demonstrated in Fig. 1, given n types of knowledge, sub-KD requires an exponential number of 2^n models, each with S parameters, while KF reduces the model number to a linear scale, with one full-sized common knowledge model and n mini models, each with s parameter, where $s \ll S$.

To this end, we propose a dedicated scheme for conducting KF, that comprises two mechanisms, namely *structural factorization* and *representation factorization.*

- **Structural Factorization.** Structural factorization decomposes the teacher network into a set of factor networks with different functionalities. Each factor network comprises a shared *common-knowledge network* (CKN) and a *task-specific network* (TSN). CKN extracts task-agnostic representations to capture the commonality among tasks, whereas the TSN accounts for task-specific information. Factor networks are trained to specialize in an individual task via fusing task-agnostic and task-specific knowledge.
- **Representation Factorization.** Representation factorization disentangles the shared knowledge and task-level representations into statistically independent components. For this purpose, we introduce a novel information-theoretical objective, termed *InfoMax Bottleneck* (IMB). It maximizes the mutual information between input and the common features to encourage

the lossless information transmission in CKN. Meanwhile, IMB minimizes data-task mutual information to ensure that, the task features are only predictive for a specific task. Specifically, we derive a variational lower bound for IMB to practically optimize this loss.

By integrating both mechanisms, we demonstrate in the experiments that KF indeed achieves architecture-level and representation-level disentanglement. Different from KD that transmits holistic knowledge in a black-box manner, KF offers unique interpretability for the factor networks through the knowledge transfer. Moreover, the learned common-knowledge representations facilitate the transfer learning to unseen downstream tasks, as will be verified empirically in our experiments.

Our contribution are therefore summarized as follows

- We introduce a novel knowledge-transfer task, termed *Knowledge Factorization* (KF), which accounts for learning factor networks that are modularized and interpretable. Factor networks are expected to be readily integratable, without any retraining, to assemble multi-task networks.
- We propose an effective solution towards KF. Our approach decomposes a pretrained teacher into factor networks that are task-wise disentangled.
- We design an *InfoMax Bottleneck* objective to disentangle the representation between common knowledge and the task-specific representations, by exerting control over the mutual information between input and representations. We derive its variational bound for its numerical optimization.
- Our method achieves strong performance and disentanglement capability across various benchmarks, with better modularity and transferability.

2 Related Work

Knowledge Distillation. Knowledge distillation (KD) [20] refers to the process to transfer the knowledge from one model or an ensemble of models to a student model. KD is originally designed for model compression [5,31,36,50,55,63], but it has been found to be beneficial in other tasks like adversarial defense [46], domain adaptation [15,43], continual learning [32,67] and amalgamate the knowledge from multiple teachers [23,38,64]. Different from the common KD methods that disseminates knowledge as a whole, we factorize the knowledge of a multi-talented teacher to factor networks with disentangled representations.

Disentangled Representation Learning. It is often assumed that real-world observations should be controlled by factors. Therefore, a recent line of research argues the importance of finding disentangled variables in representation learning [4,13,35,44,48,62] while providing invariance in learning [1,14,22]. The disentanglement are usually done through adversarial learning [10,34,40,58] or variational auto-encoder [7,19,26]. In this work, we aim to disentangle the task-agnostic and task-related representation by optimizing the mutual information.

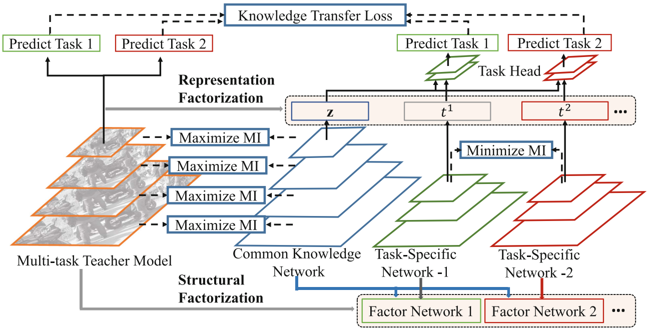


Fig. 2. The overall framework of the proposed knowledge factorization. The factor networks are trained to mimic the prediction of the teacher. The CKN learns to maximize the mutual information between input and its features, whereas the TSNs are dedicated to minimizing the task-wise mutual information.

InfoMax Principle and Information Bottleneck. As one of the foundations of machine learning, information theory has promoted a series of learning algorithms. *InfoMax* [33] is a core principle of representation learning that encourages the mutual information should be maximized between multi-views or between representation and input. This principle gave birth to the recent trend on self-supervised learning [2,21,59] and contrastive learning [9,16,17,25,45,56]. On the contrary, *Information Bottleneck* (IB) [57] aims to compress the representation while achieving realistic reconstruction results. In this study, we take a unified view of the two principles in multi-task learning. Infomax guarantees the learning of common knowledge across tasks, while IB promotes task-specific knowledge for an individual task.

Multi-task Learning. Multi-task learning (MTL) is designed to train models that handle multiple tasks by taking advantage of the common information among tasks. Some recent solutions explore on the decomposition between shared and task-specific processing [24,39,68]. Unlike conventional methods, we decompose a pre-training model into knowledge modules according to tasks.

3 Method

The essence of this work is to factorize a multi-task teacher into independent students by posing fine-grained control of the information among teacher and students. Figure 2 provides an overall sketch of our proposed KF. In what follows, we first give a definition of knowledge factorization, and then introduce the general procedure to decompose a teacher into factorized students.

3.1 Knowledge Factorization in Neural Network

We define Knowledge Factorization (KF) to be the process of subdividing a teacher network into multiple *factor networks*, each of which possesses distinctive knowledge to handle one task. Formally, assume we have a multi-task dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i^1, \ldots, y_i^K)\}$, where each input sample $\mathbf{x}$ may take one of K different labels $\{y^j\}_{j=1}^K$ sampled from the joint probability $P(X, Y_1, \ldots, Y_K)$. With a loose definition, we also deem the multi-classing as a special case for multi-tasking, by considering each or a group of categories as a task. Given a multi-task teacher model $\mathcal{T}$ that is able to predict K tasks simultaneously, KF aims to construct K factor networks $\{\mathcal{S}_j\}_{j=1}^K$, each of which, again, tackles one task independently.

Specifically, we focus on decomposing the teacher knowledge into task-specific and common representations, meaning that each factor network not only masters task-specific knowledge, but also benefits from a shared common feature to make final predictions. To this end, we design two mechanisms to factorize knowledge: *structural factorization* to decompose the teacher network into a set of factor networks, as well as *representation factorization* to disentangle the common features from task-specific features by optimizing mutual information.

3.2 Structural Factorization

The goal of structural factorization is to endow different sub-networks with functional distinctions. Each factor networks is expected to inherit only a portion of the knowledge from the teacher, and specializes in an individual task. Specifically, a factor network $\mathcal{S}_j$ for the j-th task comprises two modular networks: a Common Knowledge Network (CKN) $\mathcal{S}_C(\cdot; \Theta_{\mathcal{S}_C})$ which is shared across all tasks, and a Task-specific Network (TSN) $\mathcal{S}_{T_j}(\cdot; \Theta_{\mathcal{S}_{T_j}})$ which is task-exclusive. $\Theta_{\mathcal{S}_C}$ and $\Theta_{\mathcal{S}_{T_j}}$ are the model parameters for CKN and TSN respectively. For each input sample, $\mathcal{S}_C$ is adopted to extract the task-agnostic feature $\mathbf{z}$:

$$\mathbf{z} = \mathcal{S}_C(\mathbf{x}; \Theta_{\mathcal{S}_C}). \tag{1}$$

On the contrary, $\mathcal{S}_{T_j}$ learns the task-related knowledge $\mathbf{t}^j$ from the input $\mathbf{x}$, which together with $\mathbf{z}$ is processed by a task head $\mathcal{H}_j$ to make the final prediction:

$$\mathbf{t}^j = \mathcal{S}_{T_j}(\mathbf{x}; \Theta_{\mathcal{S}_{T_j}}); \hat{y}_S^j = \mathcal{H}_j(\mathbf{z}, \mathbf{t}^j; \Theta_{\mathcal{H}_j}), \tag{2}$$

which constrains each factor network $\mathcal{S}_j$ to share the same common knowledge network but maintain the task-specific one to handle different tasks.

Intuitively, we expect that $\mathcal{S}_j$ only masters the knowledge about task j by using the common knowledge $\mathbf{z}$ and $\mathbf{t}^j$. We accordingly define a structure factorization objective $\mathcal{L}_{sf}^{(j)}$ to enforce each single-task factor network to imitate the teacher's prediction while minimizing the supervised loss:

$$\mathcal{L}_{sf}^{(j)} = \mathcal{L}_{\text{sup}}^{(j)} + \lambda_{\text{kt}} \mathcal{L}_{\text{kt}}^{(j)}, \tag{3}$$

where $\mathcal{L}_{\text{sup}}^{(j)}$ and $\mathcal{L}_{\text{kt}}^{(j)}$ denote the supervised loss and the knowledge transfer loss for the j-th task, respectively, and λ_{kt} is the weight coefficient. Notably, we may readily adopt various implementations for each of the loss terms here. For example, $\mathcal{L}_{\text{sup}}^{(j)}$ may take the form of L2 norm for regression and cross-entropy for classification, while $\mathcal{L}_{\text{kt}}^{(j)}$ may take the form of soft-target [20], hint-loss [51], or attention transfer [66]. More details can be found in the supplement.

Structure factorization therefore enables us to construct new combined-task models by assembling multiple networks without retraining. If, for example, a 3-category classifier is needed, we can readily integrate CKN and the corresponding 3 TSNs from the pre-defined network pool. This property, in turn, greatly improves the scalability of the model.

3.3 Representation Factorization

Apart from the functionality disentanglement, we hope that learned representations of the factor networks are statistically independent as well, so that each sub-network masters task-wise disentangled knowledge. This means task-specific features should only contain minimal information only related to a certain task, while the common representation contains as much information as possible.

To this end, we introduce the *Infomax Bottleneck* (IMB) objective to optimize the mutual information (MI) between features and input. For two random variables X, Y, MI $\mathcal{I}(X, Y)$ quantifies the "number information" that variable X tells about Y, denoted by Kullback Leibler (KL) divergence between the joint probability $p(\boldsymbol{x}, \boldsymbol{y})$ and the product of marginal distribution $p(\boldsymbol{x})p(\boldsymbol{y})$:

$$\mathcal{I}(X, Y) = D_{KL}\Big[p(\boldsymbol{x}, \boldsymbol{y})||p(\boldsymbol{x})p(\boldsymbol{y})\Big]. \tag{4}$$

In our problem, for each input sample $\mathbf{x} \sim P(X)$, we compute its common knowledge feature $\mathbf{z} \sim P(Z)$ and the task-predictive representation $\mathbf{t}^j \sim P(T_j)$. Ultimately, IMB attempts to maximize $\mathcal{I}(X, Z)$ so that common knowledge keeps as much information of the input as possible, while minimize $\mathcal{I}(X, T_j)$ so that task representations only preserve information related to the task. The representation disentanglement can then be formulated as an optimization problem:

$$\max \mathcal{I}(T_j, Y_j); \quad \text{s.t. } \mathcal{I}(X, T_j) \leq \epsilon_1, -\mathcal{I}(X, Z) \leq \epsilon_2, \tag{5}$$

where ϵ_1 and ϵ_2 are the information constraints we define. In order to solve Eq. 5, we introduce two Lagrange multiplier $\alpha > 0, \beta > 0$ to construct the function:

$$\mathcal{L}_I^{(j)} = \mathcal{I}(T_j, Y_j) + \alpha\mathcal{I}(X, Z) - \beta\mathcal{I}(X, T_j). \tag{6}$$

By maximizing the first term $\mathcal{I}(T_j, Y_j)$, we ensure that the task representation $\mathbf{t}^j$ is capable to accomplish individual task j. $\mathcal{I}(X, Z)$ term encourages the lossless transmission of information and high fidelity feature extraction for the CKN, while minimizing $\mathcal{I}(X, T_j)$ enforces the only the task-informative representation is extracted by TSN, thus de-correlate the task knowledge $\mathbf{t}^j$ with the common knowledge $\mathbf{z}$. Unlike the convectional information bottleneck (IB) principle [57], our proposed IMB attempts to maximize $\mathcal{I}(X, Z)$ [21,37,45], so that the CKN learns a general representation $\mathbf{z}$ with high fidelity.

3.4 Variational Bound for Mutual Information

Due to the difficulty of estimating mutual information for continuous variables, we derive a variational lower bound to approximate the exact IMB objective[1]:

$$\hat{\mathcal{L}_I} = \mathbb{E}_{p(\mathbf{y}_j, \mathbf{t}_j)}[\log q(\mathbf{y}_j|\mathbf{t}_j)] + \alpha\big(\mathbb{E}_{p(\mathbf{z}, \mathbf{x})}[\log q(\mathbf{z}|\mathbf{x})] + H(Z)\big) - \beta\mathbb{E}_{p(\mathbf{t}_j)}\Big[D_{KL}[p(\mathbf{t}_j|\mathbf{x})||q(\mathbf{t}_j)]\Big], \tag{7}$$

[1] Due to space limitations, we only show the final formulations in the main body of this paper. The derivations can be found in the supplementary material.

where D_{KL} denotes the KL divergence between two distributions and $q(\cdot)$ denotes the variational distributions. We claim that $\mathcal{L}_I \geq \hat{\mathcal{L}_I}$, with the equality acheived if and only if $q(\mathbf{y}_j|\mathbf{t}_j) = p(\mathbf{y}_j|\mathbf{t}_j)$, $q(\mathbf{z}|\mathbf{x}) = p(\mathbf{z}|\mathbf{x})$ and $q(\mathbf{t}_j) = p(\mathbf{t}_j)$.

For better understanding, we explain the meaning of each term, specify the parametric forms of variational distribution and implementation details of Eq. 7.

Term 1. We maximize $\mathcal{I}(T_j, Y_j)$ by maximizing its lower bound $\mathbb{E}_{p(\mathbf{y}_j,\mathbf{t}_j)}$ $[\log q(\mathbf{y}_j|\mathbf{t}_j)]$. We set $q(\mathbf{y}_j|\mathbf{t}_j)$ to Gaussian for regression tasks and the multinomial distribution for classification tasks. Under this assumption, maximizing $\mathbb{E}_{p(\mathbf{y}_j,\mathbf{t}_j)}[\log q_{(}\mathbf{y}_j|\mathbf{t}_j)]$ is nothing more than minimizing the L2 norm or cross-entropy loss for the prediction. $q(\mathbf{y}_j|\mathbf{t}_j)$ is parameterized with another task head $\mathcal{H}_{j'}$ that takes $\mathbf{t}^j$ as input and makes the task prediction. Notably, $\mathcal{H}_{j'}$ is different from $\mathcal{H}_j$ since $\mathcal{H}_j$ takes both $\mathbf{z}$ and $\mathbf{t}^j$ as input.

Term 2. We maximize $\mathcal{I}(X, Z)$ by maximizing its lower bound $\mathbb{E}_{p(\mathbf{z},\mathbf{x})}[\log q(\mathbf{z}|\mathbf{x})] + H(Z)$. We choose $q(\mathbf{z}|\mathbf{x})$ to be an energy-based function that is parameterized by a critic function $f(\mathbf{x}, \mathbf{z}) : \mathcal{X} \times \mathcal{Z} \rightarrow \mathbb{R}$

$$q(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{z})}{C} e^{f(\mathbf{x},\mathbf{z})}, \text{where } C = \mathbb{E}_{p(\mathbf{z})}\left[e^{f(\mathbf{x},\mathbf{z})}\right]. \tag{8}$$

Substituting $q(\mathbf{z}|\mathbf{x})$ into the second term gives us an unnormalized lower bound:

$$\mathcal{I}(X, Z) \geq \mathbb{E}_{p(\mathbf{z},\mathbf{x})}[f(\mathbf{x}, \mathbf{z})] - \log \mathbb{E}_{p(\mathbf{x})}[C], \tag{9}$$

The same bound is also mentioned in Mutual Information Neural Estimation (MINE) [3]. Different from original MINE, in our implementation, we estimate the $\mathcal{I}(X, Z)$ through a feature-wise loss between teacher and students. With a slight abuse of notation, we refer $\mathbf{z}_{\mathcal{T}} = \mathcal{T}(\mathbf{x})_l \in \mathbb{R}^{d_{\mathcal{T}}}$ and $\mathbf{z}_{\mathcal{C}} = \mathcal{S}_{\mathcal{C}}(\mathbf{x})_l \in \mathbb{R}^{d_{\mathcal{C}}}$ as the intermediate feature vectors from teacher and CKN at the l-th layer. Given a pair of $(\mathbf{z}_{\mathcal{T}}, \mathbf{z}_{\mathcal{C}})$, f is defined as inner product of two vectors $f(\mathbf{x}, \mathbf{z}_{\mathcal{C}}) = \langle \mathbf{z}_{\mathcal{C}}, FFN(\mathbf{z}_{\mathcal{T}}) \rangle$, where $FFN(\cdot) : \mathbb{R}^{d_{\mathcal{T}}} \rightarrow \mathbb{R}^{d_{\mathcal{C}}}$ is a feed-forward network to align the dimensions between $\mathbf{z}_{\mathcal{T}}$ and $\mathbf{z}_{\mathcal{C}}$.

Term 3. $\mathbb{E}_{p(\mathbf{t}_j)}\left[D_{KL}[p(\mathbf{t}_j|\mathbf{x})||q(\mathbf{t}_j)]\right]$ is the expected KL divergence between the posterior $p(\mathbf{t}_j|\mathbf{x})$ and the prior $q(\mathbf{t}_j)$, which is a upper bound for $\mathcal{I}(X, T_j)$. We minimize $\mathcal{I}(X, T_j)$ by minimizing $\mathbb{E}_{p(\mathbf{t}_j)}\left[D_{KL}[p(\mathbf{t}_j|\mathbf{x})||q(\mathbf{t}_j)]\right]$.

Following the common practice in variational inference [19,27], we set the prior $q(\mathbf{t}_j)$ as zero-mean unit-variance Gaussian. Besides, we assume the $p(\mathbf{t}_j|\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_{t_j}, \text{diag}(\boldsymbol{\sigma}_{t_j}))$ is a Gaussian distribution. Accordingly, we compute the mean and variance for the task feature $\mathbf{t}_j$ in each forward pass:

$$\mathbf{t}_j = \mathcal{S}_{T_j}(\mathbf{x}; \Theta_{\mathcal{S}_{T_j}}); \boldsymbol{\mu}_{t_j} = \mathbb{E}[\mathbf{t}_j], \boldsymbol{\sigma}_{t_j}^2 = \text{Var}[\mathbf{t}_j], \tag{10}$$

Then, the KL divergence between $p(\mathbf{t}_j|\mathbf{x})$ and $q(\mathbf{t}_j)$ can be computed as:

$$D_{KL}[p(\mathbf{t}_j|\mathbf{x})||q(\mathbf{t}_j)] = \frac{1}{2}\sum_{l=1}^{L}(1 + \log \sigma_{t_j}^{(l)} - (\mu_{t_j}^{(l)})^2 - \sigma_{t_j}^{(l)}). \tag{11}$$

The superscript denotes the l-th element of $\boldsymbol{\mu}_{t_j}$ and $\boldsymbol{\sigma}_{t_j}$.

Training. We minimize the following overall loss to achieve both structural and representation factorization between students:

$$\min_{\Theta_{\mathcal{S}_C},\Theta_{\mathcal{S}_{T_j}},\Theta_{\mathcal{H}_j}} \sum_{j=1}^{K} \mathcal{L}_{sf}^{(j)} - \lambda_{\mathrm{I}}\mathcal{L}_{\mathrm{I}}^{(j)}, \tag{12}$$

where λ_{I} is weighting coefficient of the IMB objective.

4 Experiments

In this section, we investigate how factorization works to promote the performance, modularity and transferability of the model. Defaultly, we set $\alpha = 1.0$ and $\beta = 1\mathrm{e}{-3}$, $\lambda_{\mathrm{I}} = 1$ and $\lambda_{\mathrm{kt}} = 0.1$. Due to the space limit, more hyperparameter settings, distillation loss, implementation details, data descriptions, and definitions of the metrics are listed in supplementary material.

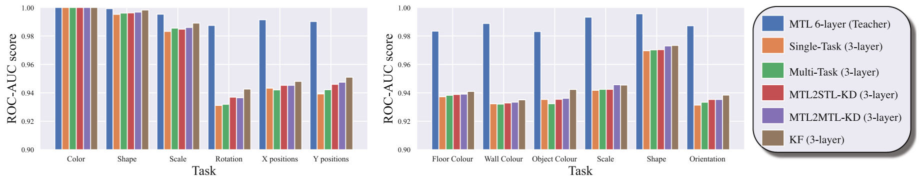
4.1 Factor Networks Make Strong Task Prediction

We conduct comprehensive experiments on synthetic and real-world classification and multi-task benchmarks to investigate whether the factorized networks still maintain competitive predictive performance, especially on each subtask.

Synthetic Evaluation. We first evaluate our KF on two synthetic imagery benchmarks dSprites [41] and Shape3D [6]. Two datasets are both generated by 6 ground truth independent latent factors. We define each latent factor as a prediction target and treat both datasets as multi-label classification benchmarks. We compare our KF with 4 other baseline methods: single-task baseline, multi-task baselines, MTL2MTL KD and MTL2STL KD. Single-task baseline denotes training 6 single-task networks, while multi-task denotes that one model trained to predict all 6 tasks. MTL2MTL KD distill a multi-tasked student, whereas MTL2STL KD refers to distilling 6 single-tasked students. KF represents our results with factor networks. We train a teacher network as 6-layer CNN model. Besides, all students network encoders, including both the CKN and TSNs, are parametrized by the 3-layer CNN. We take a random train-test split of 7:3 on each dataset and report the ROC-AUC score on the test split.

Results. Figure 3 visualizes the bar plots for the ROC-AUC scores for our KF and its KD opponents on two datasets. Though all method achieves a high AUC score larger than 0.92 on both datasets, it is evident that our KF not only surpasses the multi-tasked baseline but also exceeds two distillation paradigms. In addition, it is noted that multi-tasked models generally achieves better performance than their single-task counterpart, revealing that the prediction performance benefits from learning from multiple labels on two datasets.

Real Image Classification. We further evaluate our KF on two real image classification CIFAR-10 [29] and ImageNet1K [52]. To apply factorization, we construct two *Pseudo-Multi-task Datasets* by considering the category hierarchy.



(a) dSprites (b) Shape3D

Fig. 3. Test ROC-AUC comparison on dSprites and Shape3D datasets.

Table 1. Test accuracy (%) comparison on CIFAR-10 between KD and KF. We report mean ± std over 3 runs.

Teacher:Acc	Student/CKN:Acc	1-Task KD	2-Task KD	1-Task KF	2-Task KF
ResNet-18:94.54	MBNv2:93.58	93.79 ± 0.17	92.59 ± 0.08	94.03 ± 0.23	**94.41** ± 0.05
	ResNet-18:94.54	94.72 ± 0.24	93.69 ± 0.11	95.04 ± 0.12	**95.20** ± 0.04
	WRN28-2:93.98	94.57 ± 0.13	93.71 ± 0.22	**94.86** ± 0.17	94.77 ± 0.06
WRN28-2:93.98	MBNv2:93.58	94.14 ± 0.08	94.10 ± 0.03	94.34 ± 0.14	**94.56** ± 0.10
	ResNet-18:94.54	94.75 ± 0.22	94.22 ± 0.07	95.03 ± 0.12	**95.12** ± 0.12
	WRN28-2:93.98	94.02 ± 0.07	93.31 ± 0.12	94.59 ± 0.11	**94.62** ± 0.13
WRN28-10:95.32	MBNv2:93.58	94.47 ± 0.31	94.10 ± 0.22	94.80 ± 0.15	**94.97** ± 0.15
	ResNet-18:94.54	95.28 ± 0.14	94.62 ± 0.09	**95.40** ± 0.08	95.32 ± 0.05
	WRN28-2:93.98	94.68 ± 0.14	94.11 ± 0.26	94.80 ± 0.07	**95.03** ± 0.12

Table 2. Top-1 Accuracy (%) comparison on ImageNet.

Teacher:Acc	Student/CKN:Acc	1-Task KD	1-Task KF	11-Task KF
ResNet-18:69.90	MBNv2:71.86	72.15	72.20 (+0.05)	**72.52**(+0.37)
	ResNet-18:69.90	70.53	70.26(−0.27)	**70.93**(+0.40)
ResNet-34:73.62	MBNv2:71.86	72.58	72.95(+0.37)	**73.12**(+0.54)
	ResNet-18:69.90	70.82	70.98(+0.16)	**72.13**(+1.31)
ResNet-50:76.55	MBNv2: 71.86	72.73	72.92(+0.19)	**73.15**(+0.42)
	ResNet-18:69.90	71.12	71.14(+0.02)	**72.20** (+1.08)

The 10 classes in CIFAR-10 can be divided into 6 *animal* and 4 *vehicle* categories. Similarly, ImageNet1K classes are organized using WordNet [42] synset tree, with 11 super-classes. We accordingly construct the CIFAR-10 2-task and ImageNet1K 11-task datasets, with each task considering one super-class.

On the single-task and pseudo-multi-task evaluations, we take a pretrained classifier and distill or factorize its knowledge to single-task or pseudo-multi-task students. Each pseudo-multi-task factor/distilled network only manages to predict the categories within one super-class, with the concatenated output serving as the final prediction. We include ResNet-18 [18], WideResNet28-2 (WRN28-2) [65] and WideResNet28-10 (WRN28-10) [65] as our teacher networks on CIFAR-10; MobileNetv2 (MBNv2) [53], along with ResNet-18, WRN28-2 as student or CKN backbone. On ImageNet1K evaluation, the teacher networks are selected to be ResNet-18, ResNet-34 [18] and ResNet-50 [18], with MBNv2 and ResNet-18 as student or CKN backbone. We select a lightweight backbone MBNv2x0.5 to be TSNs. MBNv2x0.5 represents the width multiplier is 0.5.

Results. Table 1 and Table 2 provide the classification accuracy comparison between single-task or pseudo-multi-tasked KD and our proposed KF over 3 runs. Though both approaches improve the baselines under the single-task setting, we note that KD fails to improve the results on the pseudo-multi-tasked evaluation. We also do not report the 11-task KD results on ImageNet because the accuracy is generally lower than 20%. Notably, we observed that the imbalanced labeling causes the deterioration in training: when one network only masters one superclass and the rest of the classes are treated as negative samples, the distilled networks are prone to make low-confident predictions in the end. In comparison, KF has a CKN shared across all tasks, which considerably alleviates the imbalance problem in conventional KD. For example, factor networks obtained by 11-Task KF improve the performance of ResNet18-KD on ImageNet over 1.08% and 1.31% when learning from ResNet-50 and ResNet-34. On other evaluations, KF consistently makes progress overall the normal KD, which suggests that the factorization of task-specific and task-agnostic benefit the performance.

Multi-task Dense Prediction. Two multi-task dense prediction datasets are also used to verify the effectiveness of KF, including NYU Depth Dataset V2 (NYUDv2) [54] and PASCAL Context [11]. NYUDv2 dataset contains indoor scene images annotated for segmentation and monocular depth estimation. We include 4 tasks in PASCAL Context, including semantic/human part segmentation, normal prediction, and saliency detection. We use the mean intersection over union (mIoU), the angle mean error (mErr) and root mean square error (rmse) are used to measure the prediction quality.

We include both the single-task and multi-task together with their STL2STL/MTL2STL/MTL2MTL distilled models as our baselines. We adopt the HRNet48 [61] and ResNet-50 DeepLabv3 as teacher and HRNet18 and ResNet-18 DeepLabv3 as student or CNK. The TSN are set to MBNv2x0.5. We use a smaller $\beta = 1\text{e}{-5}$. The networks are initialized with the ImageNet pretrained weights.

Results. We show the evaluation results on NYUDv2 and PASCAL datasets in Table 3 and Table 4. On NYUDv2, the multi-task baselines are generally better-performed than its single-task competitors. On the contrary, in the PASCAL experiments of HRNet48, ResNet18 and ResNet50, the performance of multitask baseline has largely degraded. It reveals the *negative transfer* problem in MTL that the joint optimization of multiple objective might cause the contradiction between tasks, thus leading to undesirable performance reduction.

The same problem remains when comparing MTL2MTL-KD to STL2-STL-KD in Table 4, where the MTL teacher is inferior to STL ones. Our factor networks automatically resolve this problem, because different TSNs are structurally and representationally independent. As a result, KF achieved strong student performance compared to other baselines.

Table 3. Performance comparison on the NYUDv2 dataset.

Method	Teacher	Student/CKN	Seg. (mIoU)↑	Depth (rmse)↓
Single-task	–	HRNet18	27.37	0.612
Multi-task	–	HRNet18	37.59	0.641
Single-task	–	HRNet48	48.19	0.556
Multi-task	–	HRNet48	48.92	0.578
STL2STL-KD	HRNet48	HRNet18	39.27	0.603
MTL2MTL-KD	HRNet48	HRNet18	38.02	0.604
MTL2STL-KD	HRNet48	HRNet18	39.04	0.601
Ours	HRNet48	HRNet18	**40.78**	**0.592**
Single-task	–	ResNet-18	38.07	0.652
Multi-task	–	ResNet-18	39.18	0.623
Single-task	–	ResNet-50	44.30	0.625
Multi-task	–	ResNet-50	44.78	0.602
STL2STL-KD	ResNet-50	ResNet-18	39.76	0.633
MTL2MTL-KD	ResNet-50	ResNet-18	39.98	0.623
MTL2STL-KD	ResNet-50	ResNet-18	40.60	0.621
Ours	ResNet-50	ResNet-18	**41.33**	**0.615**

Table 4. Performance comparison on the PASCAL dataset.

Method	Teacher	Student/CKN	Seg.(mIoU)↑	H.Part(mIOU)↑	Norm.(mErr)↓	Sal.(mIOU)↑
Single-task	–	HRNet18	51.18	64.10	14.54	56.08
Multi-task	–	HRNet18	54.61	62.40	14.77	66.07
Single-task	–	HRNet48	60.92	67.15	14.53	68.12
Multi-task	–	HRNet48	55.93	67.06	14.31	67.08
STL2STL-KD	HRNet48	HRNet18	52.63	64.98	14.49	60.72
MTL2MTL-KD	HRNet48	HRNet18	52.02	60.33	14.63	65.45
MTL2STL-KD	HRNet48	HRNet18	54.77	65.18	14.53	64.31
Ours	HRNet48	HRNet18	**56.65**	**66.83**	**14.44**	**67.05**
Single-task	–	ResNet-18	64.75	58.68	13.95	65.59
Multi-task	–	ResNet-18	63.48	58.17	15.12	64.50
Single-task	–	ResNet-50	70.29	61.47	14.65	66.22
Multi-task	–	ResNet-50	68.04	63.05	14.88	65.65
STL2STL-KD	ResNet-50	ResNet-18	66.10	59.43	**14.19**	66.33
MTL2MTL-KD	ResNet-50	ResNet-18	61.31	60.14	14.73	62.45
MTL2STL-KD	ResNet-50	ResNet-18	66.60	**62.33**	14.29	66.14
Ours	ResNet-50	ResNet-18	**67.18**	61.09	14.31	**66.83**

4.2 Factorization Brings Disentanglement

Given the distilled and factorized models in the previous section we measure a set of disentanglement metrics and representation similarity to confirm that the knowledge factorization captures the independent variables across tasks.

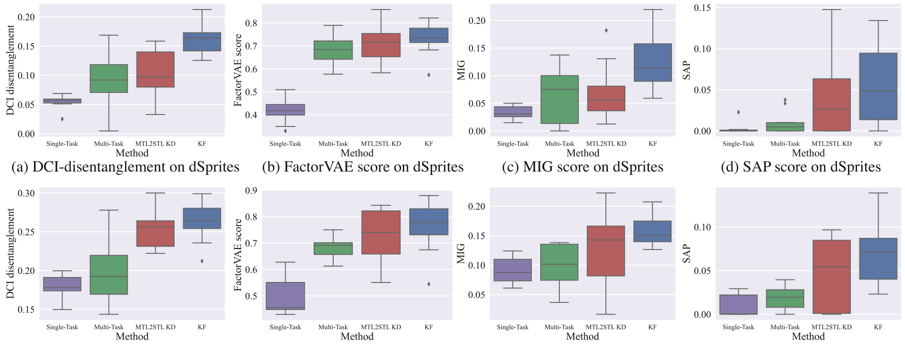
Disentanglement Evaluation Setup. We first validate the disentanglement between factor models on dSprites [41] and Shape3D [6]. We measure 4 disentanglement metrics to quantify how well the learned representations summarize the factor variables. Those metrics are disentanglement-completness-informativeness (DCI) [12], Mutual information gap (MIG) [8], FactorVAE metric [26], and Separated Attribute Predictability (SAP) score [30]. Higher means better.

We compare our KF with 3 other baseline methods: single-task baseline, multi-task baselines, and MTL2STL KD students, which has been introduced in previous section. Following the evaluation protocol in [35], we adopt the concatenation of all average-pooled task-specific representations as our final feature vector for evaluation and compute all scores on test set.

Results. Figure 4 illustrates the quantitative results of different disentanglement metrics using box plots. First, we see that multi-task learning naturally comes with disentangled representations, where MTL achieves a slightly higher score than the STL. Another observation is that knowledge transfer methods like KD and KF also help the model to find factors that are unappreciable for the teachers. The features extracted by our factor networks generally score the best, especially on the dSprites dataset, with an improvement over median of 0.47 and 0.09 on DCI and MIG scores. It is in line with our expectation that decomposing the knowledge into parts leads to disentangled representations.

Representation Similarity. We further conduct representation similarity analysis using centered kernel alignment (CKA) [28] between teacher models, distilled models and our factorized models across 4 datasets, including dSprites, Shape3D, CIFAR10 and NYUDv2. On each dataset, CKA is adopted to quantifying feature similarity among (1) MTL teacher (2) MTL2MTL-KD student (3) MTL2STL students and (4) Our CKN and TSNs. We compute linear kernel CKA between all pairs of models at the last feature layer on test set. The model architectures are described in the Appendix. The higher CKA index suggests higher correlation between two networks.

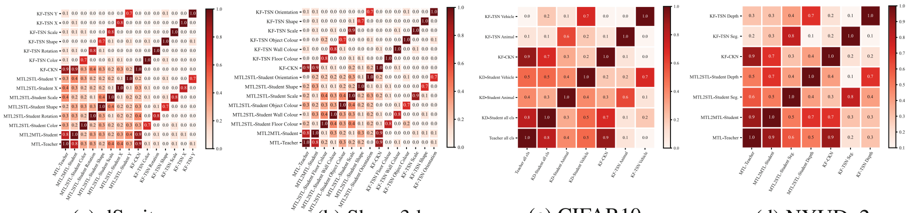
Results. Figure 5 visualizes the CKA confusion matrix between all model pairs on 4 tasks. We made the following observations. First, models mastering the same subtask has high feature similarity. Second, our factorized TSN captures more "pure" knowledge compared with MTL2STL students. On each heatmap, the bottom left region has high similarity (in darker red), suggesting that the conventional distilled models still maintains high similarity with its peers even though they are trained on dedicated tasks. In comparison, factorized TSNs achieve smaller similarities (in upper right region), again supporting our argument that factor networks capture the disentangled factors across tasks.



(a) DCI-disentanglement on dSprites (b) FactorVAE score on dSprites (c) MIG score on dSprites (d) SAP score on dSprites

(e) DCI-disentanglement on Shape3D (f) FactorVAE score on Shape3D (g) MIG score on Shape3D (h) SAP score on Shape3D

Fig. 4. Disentanglement Metrics comparison between (1) Single-Task Baseline, (2) Multi-Task Baseline, (3) KD, and (4) our proposed KF on dSprite (top) and Shape3D (bottom) datasets. Each experiment is repeated over 10 runs.



(a) dSprites (b) Shape3d (c) CIFAR10 (d) NYUDv2

Fig. 5. CKA representation similarity between distilled and factorized models.

4.3 Common Knowledge Benefits Transferring

We then finetune the factorized CKN on two downstream tasks to see if the common knowledge facilities the transfer learning to unseen domains. We train ResNet-18 networks with different initializations on Caltech-UCSD Birds (CUB-200) [60] and MIT indoor scene (Scene) [49]. The trained models are then reestablished as teachers to educate student networks like MBNv2 and ShuffleNetv2.

Results. Table 5 shows the transfer learning performance and distillation accuracy using different pretrained weights. R18 w/ImageNet-CKN refers to the ResNet-18 CKN factorized from ImageNet pretrained ResNet-18. Compared with the original pretrained weights, ImageNet-CKN achieves substantial improvement on both datasets. By reusing the finetuned ResNet-18 as teacher network, we show in Fig. 5 that CKN serves as a better role model to educate the student networks. It provides compelling evidence that common knowledge factorized from the teacher network benefits the transfer learning to other tasks.

Table 5. Finetuning performance and distillation accuracy with different pretrained weights. R18 is the short for ResNet-18.

Teacher	Student	CUB-200	Scene
–	R18 w/Rand init.	46.14	65.17
–	R18 w/ImageNet	65.28	65.19
–	R18 w/ImageNet-CKN	**69.17**	**72.37**
–	MobileNetV2 w/Rand init.	48.80	64.59
R18 w/Rand init.	MobileNetV2 w/Rand init.	54.18	66.78
R18 w/ImageNet	MobileNetV2 w/Rand init.	61.30	66.40
R18 w/ImageNet-CKN	MobileNetV2 w/Rand init.	**64.25**	**70.94**
–	ShuffleNetv2 w/Rand init.	52.51	64.39
R18 w/Rand init.	ShuffleNetv2 w/Rand init.	48.19	65.70
R18 w/ImageNet	ShuffleNetv2 w/Rand init.	59.15	66.00
R18 w/ImageNet-CKN	ShuffleNetv2 w/Rand init.	**60.69**	**68.95**

5 Conclusion

In this paper, we introduce a novel knowledge-transfer task termed *Knowledge Factorization*. Given a pretrained teacher, KF decomposes it into task-disentangled factor networks, each of which masters the task-specific and the common knowledge factorized from the teacher. Factor networks may operate independently, or be integrated to assemble multi-task networks, allowing for great scalability. We design an InfoMax Bottleneck objective to disentangle the common and task-specific representations by optimizing the mutual information between input and representations. Our method achieves strong and robust performance, and meanwhile demonstrates great disentanglement capability across various benchmarks, with better modularity and transferability.

Acknowledgement. This work is supported by NUS Advanced Research and Technology Innovation Centre (ARTIC) Project Reference ECT-RP2, and Faculty Research Committee Grant (WBS: A-0009440-00-00). Xinchao Wang is the corresponding author.

References

1. Achille, A., Soatto, S.: Emergence of invariance and disentanglement in deep representations. J. Mach. Learn. Res. **19**(1), 1947–1980 (2018)
2. Bachman, P., Hjelm, R.D., Buchwalter, W.: Learning representations by maximizing mutual information across views. arXiv preprint arXiv:1906.00910 (2019)
3. Belghazi, M.I., et al.: MINE: mutual information neural estimation. arXiv preprint arXiv:1801.04062 (2018)
4. Bengio, Y., Courville, A., Vincent, P.: Representation learning: a review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell. **35**(8), 1798–1828 (2013)

5. Buciluǎ, C., Caruana, R., Niculescu-Mizil, A.: Model compression. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 535–541 (2006)
6. Burgess, C., Kim, H.: 3d shapes dataset (2018). https://github.com/deepmind/3dshapes-dataset/
7. Burgess, C.P., et al.: Understanding disentangling in β-VAE. arXiv preprint arXiv:1804.03599 (2018)
8. Chen, R.T., Li, X., Grosse, R., Duvenaud, D.: Isolating sources of disentanglement in variational autoencoders. arXiv preprint arXiv:1802.04942 (2018)
9. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning, pp. 1597–1607. PMLR (2020)
10. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: InfoGAN: interpretable representation learning by information maximizing generative adversarial nets. In: NIPS (2016)
11. Chen, X., Mottaghi, R., Liu, X., Fidler, S., Urtasun, R., Yuille, A.: Detect what you can: detecting and representing objects using holistic models and body parts. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1971–1978 (2014)
12. Eastwood, C., Williams, C.K.: A framework for the quantitative evaluation of disentangled representations. In: International Conference on Learning Representations (2018)
13. Feng, Z., Wang, X., Ke, C., Zeng, A., Tao, D., Song, M.: Dual swap disentangling. In: Conference on Neural Information Processing Systems (2018)
14. Goodfellow, I., Lee, H., Le, Q., Saxe, A., Ng, A.: Measuring invariances in deep networks. Adv. Neural. Inf. Process. Syst. **22**, 646–654 (2009)
15. Granger, E., Kiran, M., Dolz, J., Blais-Morin, L.A., et al.: Joint progressive knowledge distillation and unsupervised domain adaptation. In: 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2020)
16. Grill, J.B., et al.: Bootstrap your own latent: a new approach to self-supervised learning. arXiv preprint arXiv:2006.07733 (2020)
17. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9729–9738 (2020)
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
19. Higgins, I., et al.: beta-VAE: learning basic visual concepts with a constrained variational framework (2016)
20. Hinton, G.E., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. ArXiv abs/1503.02531 (2015)
21. Hjelm, R.D., et al.: Learning deep representations by mutual information estimation and maximization. arXiv preprint arXiv:1808.06670 (2018)
22. Jaiswal, A., Wu, Y., AbdAlmageed, W., Natarajan, P.: Unsupervised adversarial invariance. arXiv preprint arXiv:1809.10083 (2018)
23. Jing, Y., Yang, Y., Wang, X., Song, M., Tao, D.: Amalgamating knowledge from heterogeneous graph neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15709–15718 (2021)

24. Kanakis, M., Bruggemann, D., Saha, S., Georgoulis, S., Obukhov, A., Van Gool, L.: Reparameterizing convolutions for incremental multi-task learning without task interference. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12365, pp. 689–707. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58565-5_41
25. Khosla, P., et al.: Supervised contrastive learning. arXiv preprint arXiv:2004.11362 (2020)
26. Kim, H., Mnih, A.: Disentangling by factorising. ArXiv abs/1802.05983 (2018)
27. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
28. Kornblith, S., Norouzi, M., Lee, H., Hinton, G.: Similarity of neural network representations revisited. In: International Conference on Machine Learning, pp. 3519–3529. PMLR (2019)
29. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
30. Kumar, A., Sattigeri, P., Balakrishnan, A.: Variational inference of disentangled latent concepts from unlabeled observations. arXiv preprint arXiv:1711.00848 (2017)
31. Li, T., Li, J., Liu, Z., Zhang, C.: Few sample knowledge distillation for efficient network compression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14639–14647 (2020)
32. Li, Z., Hoiem, D.: Learning without forgetting. IEEE Trans. Pattern Anal. Mach. Intell. **40**(12), 2935–2947 (2017)
33. Linsker, R.: Self-organization in a perceptual network. Computer **21**(3), 105–117 (1988)
34. Liu, Y., Wang, Z., Jin, H., Wassell, I.: Multi-task adversarial network for disentangled feature learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3743–3751 (2018)
35. Locatello, F., et al.: Challenging common assumptions in the unsupervised learning of disentangled representations. In: International Conference on Machine Learning, pp. 4114–4124. PMLR (2019)
36. Lopes, R.G., Fenu, S., Starner, T.: Data-free knowledge distillation for deep neural networks. arXiv preprint arXiv:1710.07535 (2017)
37. Löwe, S., O'Connor, P., Veeling, B.S.: Greedy infomax for self-supervised representation learning (2019)
38. Luo, S., Wang, X., Fang, G., Hu, Y., Tao, D., Song, M.: Knowledge amalgamation from heterogeneous networks by common feature learning. In: Kraus, S. (ed.) Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, 10–16 August 2019, pp. 3087–3093. ijcai.org (2019). https://doi.org/10.24963/ijcai.2019/428
39. Maninis, K.K., Radosavovic, I., Kokkinos, I.: Attentive single-tasking of multiple tasks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1851–1860 (2019)
40. Mathieu, M.F., Zhao, J.J., Zhao, J., Ramesh, A., Sprechmann, P., LeCun, Y.: Disentangling factors of variation in deep representation using adversarial training. In: Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 29. Curran Associates, Inc. (2016). https://proceedings.neurips.cc/paper/2016/file/ef0917ea498b1665ad6c701057155abe-Paper.pdf
41. Matthey, L., Higgins, I., Hassabis, D., Lerchner, A.: dSprites: disentanglement testing sprites dataset (2017). https://github.com/deepmind/dsprites-dataset/

42. Miller, G.A.: WordNet: a lexical database for English. Commun. ACM **38**(11), 39–41 (1995)
43. Nguyen-Meidine, L.T., Belal, A., Kiran, M., Dolz, J., Blais-Morin, L.A., Granger, E.: Unsupervised multi-target domain adaptation through knowledge distillation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 1339–1347 (2021)
44. Niemeyer, M., Geiger, A.: GIRAFFE: representing scenes as compositional generative neural feature fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11453–11464 (2021)
45. van den Oord, A., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
46. Papernot, N., McDaniel, P., Wu, X., Jha, S., Swami, A.: Distillation as a defense to adversarial perturbations against deep neural networks. In: 2016 IEEE Symposium on Security and Privacy (SP), pp. 582–597. IEEE (2016)
47. Passalis, N., Tefas, A.: Learning deep representations with probabilistic knowledge transfer. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11215, pp. 283–299. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01252-6_17
48. Peters, J., Janzing, D., Schölkopf, B.: Elements of Causal Inference: Foundations and Learning Algorithms. The MIT Press, Cambridge (2017)
49. Quattoni, A., Torralba, A.: Recognizing indoor scenes. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 413–420. IEEE (2009)
50. Ren, S., Zhou, D., He, S., Feng, J., Wang, X.: Shunted self-attention via multi-scale token aggregation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022)
51. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: FitNets: hints for thin deep nets. arXiv preprint arXiv:1412.6550 (2014)
52. Russakovsky, O., et al.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vision **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y
53. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: MobileNetV 2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510–4520 (2018)
54. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from RGBD images. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7576, pp. 746–760. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33715-4_54
55. Sun, S., Cheng, Y., Gan, Z., Liu, J.: Patient knowledge distillation for BERT model compression. In: EMNLP (2019)
56. Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., Isola, P.: What makes for good views for contrastive learning? arXiv preprint arXiv:2005.10243 (2020)
57. Tishby, N., Pereira, F.C., Bialek, W.: The information bottleneck method. arXiv preprint physics/0004057 (2000)
58. Tran, L., Yin, X., Liu, X.: Disentangled representation learning GAN for pose-invariant face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1415–1424 (2017)
59. Tschannen, M., Djolonga, J., Rubenstein, P.K., Gelly, S., Lucic, M.: On mutual information maximization for representation learning. arXiv preprint arXiv:1907.13625 (2019)

60. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD birds-200-2011 dataset. Technical report. CNS-TR-2011-001, California Institute of Technology (2011)
61. Wang, J., et al.: Deep high-resolution representation learning for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. **43**, 3349–3364 (2020)
62. Yang, Y., Feng, Z., Song, M., Wang, X.: Factorizable graph convolutional networks. In: Conference on Neural Information Processing Systems (2020)
63. Yang, Y., Qiu, J., Song, M., Tao, D., Wang, X.: Distilling knowledge from graph convolutional networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020)
64. Ye, J., Ji, Y., Wang, X., Ou, K., Tao, D., Song, M.: Student becoming the master: Knowledge amalgamation for joint scene parsing, depth estimation, and more. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2829–2838 (2019)
65. Zagoruyko, S., Komodakis, N.: Wide residual networks. arXiv preprint arXiv:1605.07146 (2016)
66. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer. In: ICLR (2017). https://arxiv.org/abs/1612.03928
67. Zenke, F., Poole, B., Ganguli, S.: Continual learning through synaptic intelligence. In: International Conference on Machine Learning, pp. 3987–3995. PMLR (2017)
68. Zhang, J.O., Sax, A., Zamir, A., Guibas, L., Malik, J.: Side-tuning: a baseline for network adaptation via additive side networks. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12348, pp. 698–714. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58580-8_41