

PROBABILISTIC KNOWLEDGE TRANSFER FOR LIGHTWEIGHT VISION TRANSFORMERS

Ioannis Bountouris, Nikolaos Passalis, and Anastasios Tefas

Dept. of Informatics, Faculty of Sciences

Aristotle University of Thessaloniki

Thessaloniki, Greece

E-mails: mpountoui@csd.auth.gr, passalis@auth.gr, tefas@csd.auth.gr

Abstract—The advent of Vision Transformers has reshaped the landscape of computer vision, harnessing the ability of the Transformer architecture to capture intricate long-range dependencies within data. Nevertheless, the computational demands of Transformers remain persistent, prompting the need for methodologies focused on model compression. Knowledge distillation has emerged as a viable solution, facilitating the transfer of insights from larger deep learning models to more compact versions while maintaining accuracy. Among them, Probabilistic Knowledge Transfer (PKT) has been proven particularly effective since instead of relying solely on the final predictions of the teacher, it models the conditional probabilities of data representations extracted through various layers to capture the underlying geometry of the data representations. However, applying PKT in the context of Transformer models is not straightforward, leading to significant challenges (e.g., density estimation in high dimensional spaces). The main contribution of this work is a patch-based knowledge distillation approach, building upon the powerful modeling capabilities of PKT and focusing on Vision Transformers architectures. To this end, we employ a lightweight Vision Transformer architecture, targeting deployment on embedded systems. We demonstrate the effectiveness of the proposed approach through comprehensive experiments on CIFAR-10 and Tiny-Imagenet datasets.

I. INTRODUCTION

Vision Transformers [1] have revolutionized the computer vision landscape by leveraging the Transformer architecture’s ability to capture long-range dependencies in data. These models divide input images into patches and process them like the tokens in natural language processing tasks. Nevertheless, the computational demands of Transformers remain persistent, prompting the need for methodologies focused on model compression. To this end, researchers have explored lightweight alternatives like MobileViT [2], a specialized version of Vision Transformer models designed for efficiency on resource-constrained devices. However, despite the advantages these models provide, their performance is still below their larger counterparts.

At the same time, knowledge distillation has proven effective in this regard [3], [4], enabling the transfer of knowledge encoded in larger deep learning models to more compact ones, while retaining accuracy. The concept of transferring both explicit and implicit knowledge from the teacher to the student has driven numerous recent works in the field [5]. This transfer of knowledge may occur for example through

soft probabilities [3], allowing the student model to glean insights beyond mere one-hot labels, or might include intermediate layers as well. For example, FitNets [6] suggested the use of intermediate representations (hidden layers) from the teacher model to provide guidance to the student model, enabling a more effective transfer process. Flow of Solution Procedure (FSP) matrices were also similarly employed in [7], in order to model and transfer knowledge from intermediate layers. Attention Transfer [8] highlights the importance of attention maps in the transfer process, improving student model performance by allowing it to focus on salient input regions. As far as Transformers are concerned, techniques like DistillBERT [9] and TinyBERT [10] have paved the way for effective knowledge distillation, opening avenues for making these powerful architectures more accessible and efficient, but mainly focusing on applying regular distillation.

Most of these methods rely either on direct representation matching or require intermediate representations of the same size (which cannot always be guaranteed when the teacher and student models have different sizes). To this end, Probabilistic Knowledge Transfer (PKT) has been introduced [11]. Instead of relying solely on the final predictions of the teacher, PKT models the conditional probabilities of data representations extracted through various layers of a network to capture the underlying geometry of the data representations. Then, a probability distribution divergence metric is employed to transfer the knowledge from the teacher to the student model.

Despite the potential of such an approach, using it in the context of Vision Transformers is not straightforward, since it is not trivial to combine token representations to get a compact representation to transfer knowledge in an efficient manner. One potential approach would be to rely on classification (“cls”) tokens for distillation, but this would lead to loss of valuable information, encoded by the remaining tokens, during the distillation process. Another approach could be to rely on fused representations of tokens. Even though this approach retains all the information, it leads to high dimensional representations, which can negatively affect the distillation process [12].

The main contribution of this work is a patch-based knowledge distillation approach that is targeted specifically to Vision Transformers and builds upon the powerful modeling capabilities

ties of PKT, while also overcoming the difficulties of handling the large intermediate representations of Vision Transformers. To this end, we propose separately distilling the knowledge encoded by tokens that correspond to different patches, overcoming the aforementioned difficulties. Furthermore, we also employed a lightweight Vision Transformer architecture, targeting deployment on embedded systems. We demonstrate the effectiveness of the proposed approach through comprehensive experiments on CIFAR-10 and Tiny-Imagenet datasets, as well as experimental studies that demonstrate the impact of design various choices on the performance of the proposed approach.

The rest of the paper is structured as follows. Section II introduces the proposed method, while the experimental evaluation is provided in Section III. Finally, conclusions are drawn in Section IV.

II. PROPOSED METHOD

At the heart of our approach lies the Probabilistic Knowledge Transfer (PKT) method [11], which serves as a powerful mean to facilitate efficient knowledge transfer between the teacher and student models. Therefore, we first outline the PKT approach. Then, we proceed by presenting the proposed method for applying PKT in the context of Transformer models, discussing challenges and various design choices.

A. Probabilistic Knowledge Transfer

The way the PKT method works is briefly outlined below. First, we use a teacher network $f(\cdot) \in \mathbb{R}^{N_t}$ to extract a representation for each input sample. The student model is denoted by $g_{\mathbf{W}}(\cdot) \in \mathbb{R}^{N_s}$. The dimensionality of the embedding space of the student (N_s) and teacher model (N_t) can be different. During knowledge transfer the student's parameters \mathbf{W} are trained in order to make the student model behave similarly to the teacher model. Let $\mathbf{x}_i^{(T)} = f(\mathbf{t}_i)$ denote the representation extracted from a specific layer of the teacher model from which we want to transfer the knowledge and $\mathbf{x}_i^{(S)} = g_{\mathbf{W}}(\mathbf{t}_i)$ be the representation extracted from student model, where we want to transfer the knowledge. Note that \mathbf{t}_i is a transfer sample contained in the transfer set $\mathcal{T} = \{\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_N\}$ used for knowledge distillation, where N denotes the number of samples used for knowledge distillation.

PKT works by estimating the conditional probability distribution of the representation of two data samples using both the teacher and student models. Therefore, we define the teacher conditional probability distribution $p_{i|j}$ which expresses how similar the samples in the feature space formed by a layer of the teacher model. This probability can be estimated using kernel density estimation as:

$$p_{i|j} = \frac{K(\mathbf{x}_i^{(T)}, \mathbf{x}_j^{(T)}; 2\sigma_t^2)}{\sum_{k=1, k \neq j} K(\mathbf{x}_k^{(T)}, \mathbf{x}_j^{(T)}; 2\sigma_t^2)}, \quad (1)$$

where $\mathbf{x}_i^{(T)}$ denotes the representation of the i -th training as extracted by a layer of the teacher model. Also, $K(\mathbf{a}, \mathbf{b}; \sigma_t^2)$ denotes is a symmetric kernel with width σ_t . Similarly, the

corresponding density can be estimated for the student model as:

$$q_{i|j} = \frac{K(\mathbf{x}_i^{(S)}, \mathbf{x}_j^{(S)}; 2\sigma_s^2)}{\sum_{k=1, k \neq j} K(\mathbf{x}_k^{(S)}, \mathbf{x}_j^{(S)}; 2\sigma_s^2)}, \quad (2)$$

where the representation extracted by a layer of the student model is denoted by $\mathbf{x}_i^{(S)}$.

To avoid the need for kernel bandwidth tuning in kernel density estimation, a cosine similarity-based affinity metric is usually employed to estimate conditional probabilities [11], [12]. Therefore, the kernel is defined as:

$$K_{\cosine}(\mathbf{a}, \mathbf{b}) = \frac{1}{2} \left(\frac{(\mathbf{a}^T \mathbf{b})}{\|\mathbf{a}\|^2 \|\mathbf{b}\|^2} + 1 \right), \quad (3)$$

where \mathbf{a} and \mathbf{b} two input vectors. Then, Kullback-Leibler (KL) divergence can be employed for measuring the distance between these two distributions. Therefore, the loss function for training the student model is calculated as follows:

$$L = \sum_{i=1}^N \sum_{j=1, j \neq i}^N p_{j|i} \log \left(\frac{p_{j|i}}{q_{j|i}} \right) \quad (4)$$

Note that KL divergence is not symmetric, giving more weight to minimizing the divergence for neighboring pairs of data points.

B. Vision Transformers and Probabilistic Knowledge Transfer

Applying this methodology directly to a representation extracted by a layer of a Transformer model is not straightforward. The Transformer model's decision-making primarily relies on the classification ("cls") token, making it a straightforward choice for applying the Probabilistic Knowledge Transfer (PKT) method. However, this approach could lead to losing valuable information regarding the way the models function, which is encoded in the other tokens. Therefore, instead of focusing solely on the "cls" token, we can combine all tokens into a vector and employ this vector for knowledge distillation.

However, such an approach would not fully exploit the wealth of information contained within the Transformer, while it could also lead to high dimensional representations that can lead to less effective density estimation. Each patch, thanks to the attention layers, carries valuable positional and relational information with respect to other layers. To harness this information effectively, we adopted a patch-based strategy. We applied PKT independently for each patch within each layer and aggregated these individual losses to formulate the proposed method, allowing for matching the geometry of tokens at each layer of the student model with the geometry of tokens at corresponding layers of the teacher model.

Specifically, our method involves the application of PKT between the hidden representations (tokens) of each layer (l) of the teacher and student Transformers at patch level. We treat each token in the hidden representation as an instance (i), resulting in as many instances as tokens in the input image. The dimension of each instance is equal to the size of the token after the linear projection. To facilitate PKT, we compare the hidden representations (instances) of the student

and teacher models. For each layer, we consider the tokens of the student model ($\mathbf{x}_i^{(S)}$) and the tokens of the teacher model ($\mathbf{x}_i^{(T)}$) in their respective hidden dimensions and we define the corresponding loss, denoted by L_{PKT}^l . The goal is to align the geometry representation of the hidden dimensions in the student model with that of the teacher model through PKT knowledge transfer. Therefore, the loss used for knowledge transfer is defined as:

$$L = \sum_{l=1}^M \alpha_l L_{PKT}^l \quad (5)$$

where M is the the number of layers, L_{PKT}^l is the loss between the teacher and student for the corresponding layer and α_l is a factor that adjusts the importance layer l during the distillation process. PKT is applied individually to each image in the batch during training and then the loss is aggregated.

There are several different approaches that can be used for selecting the weighting factors α_l . Perhaps the simplest one is to set α_l to 1 for all layers. This implies that the importance of transferring knowledge from all layers is equal. However, literature suggest that this is not always the case [12], highlighting the impact of both early and late layers on the knowledge distillation process.

Indeed, for late layers, the importance of knowledge distillation is evident, as also the original knowledge distillation approach emphasized [3]. Transferring knowledge at (or near) the classification layer can immediately affect the decision boundary of the classifier leading to improved performance. However, at the same time, difficulties in back-propagating gradients in the early layers of the network, along with the information processing inequality [13], which suggests that information lost in the early layers of a network cannot be recovered later, suggest that early layer might be equally important as well.

Motivated by these observations, we present a scheme, in which the last and the early layers have the same contribution in the knowledge transfer process (i.e., $\alpha_l = 1$), while we propose linearly decaying the weighting factor for the intermediate layers. Indeed, as we demonstrate in Section III-C, this approach can lead to more effective knowledge transfer, improving the accuracy of the models. We also provide additional details on how such an approach can be applied to the examined architecture in Section III-C.

III. EXPERIMENTAL EVALUATION

In this section, we present a comprehensive evaluation of the proposed method leveraging the CIFAR-10 and Tiny-Imagenet datasets. Then, we proceed by presenting the experimental evaluation results. Finally, we present an evaluation of different weighting strategies for selecting the impact of different layers on the knowledge distillation approach.

A. Experimental Setup

In this subsection we first outline the employed model architectures. Then, we proceed by presenting the employed training setup.

a) Model Architecture: For our experiments, we utilize two Transformer-based models: a Large Vision Transformer serving as the teacher, boasting 85 million parameters, and a small Vision Transformer, which is a compact version of the teacher, acting as the student with 1.5 million parameters. As far as the teacher is concerned we finetuned the pretrained Transformer ViT-Base [1] for the CIFAR-10 [14] and Tiny-Imagenet [15] datasets. To ensure a direct correspondence between both models, we maintain an identical number of layers, and heads. However, a key divergence arises in the linear projection layer, wherein the student model radically reduces the dimensionality of tokens compared to the teacher. Specifically, after the linear projection, the new token in the student model has only 96 features, while the corresponding patch in the teacher model contains 768 features. Consequently, this dimensionality reduction leads to smaller attention matrices, with dimensions of 96×96 , as opposed to the original 768×768 matrices in the teacher model. The details of the employed architectures are summarized in Table I.

TABLE I
COMPARING THE ARCHITECTURE OF TEACHER AND STUDENT MODELS

	Layers	Heads	Patches	Proj. Dim.	Parameters
Teacher	12	12	14×14	768	85M
Student	12	12	14×14	96	1.5M

b) Training Setup: Regarding the training setup, for the Cifar-10 dataset the models were trained for 40 epochs, utilizing a carefully crafted strategy. We initially trained the model for 30 epochs, employing a batch size of 64 and a learning rate of 0.0003. Subsequently, to enhance convergence and model performance, we reduced the learning rate to 0.0002 for the final 10 epochs. The same setup was used for all the evaluated methods.

In the case of TinyImageNet, we adopt a more intricate training procedure to address instabilities that occur during the training process. We initiate training with a learning rate of 0.0003, continuing for an initial span of 90 epochs. Following the first 10 epochs, we gradually reduce the influence of the proposed knowledge distillation loss on the overall loss by a factor of 0.1, sustained over the subsequent 20 epochs. Subsequently, we enact a further reduction in the impact of the proposed knowledge distillation, again by a factor of 0.1, extending over the subsequent 10 epochs. Upon reaching the 40th epoch, one more reduction to 0.001 takes place. To fine-tune the process, we subsequently decrease the learning rate to 0.0002. This adjusted learning rate is employed for the final phase of training, spanning 10 epochs, during which we reduce the influence of the proposed knowledge distillation loss on the overall loss using a weight of 0.0001. This multi-stage training approach aims to effectively balance and enhance the training process, mitigating potential instabilities and contributing to the model's improved performance.

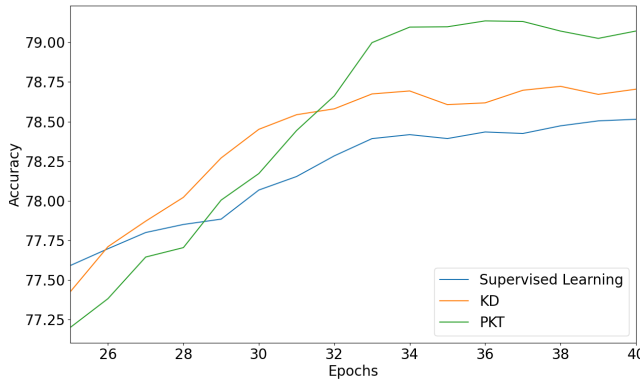


Fig. 1. The figure provides a comparison of results on the Cifar10 dataset among different methods. The blue line represents supervised learning without any additional approach used, the orange line represents knowledge distillation, while the green line represents the proposed method.

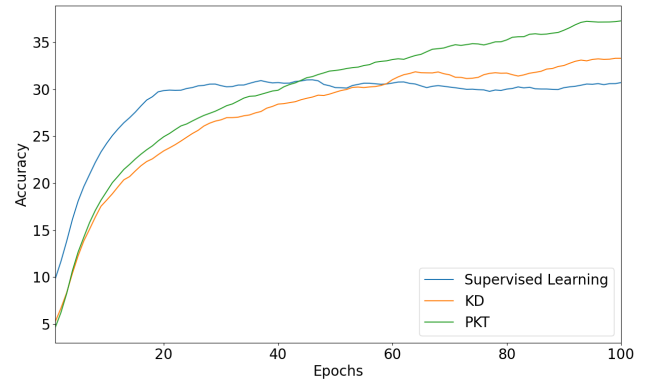


Fig. 2. The figure provides a comparison of results on the TinyImagenet dataset among different methods. The blue line represents supervised learning without any additional approach used, the orange line represents knowledge distillation, while the green line represents the proposed method.

We initiated our evaluation by employing classic supervised training as the baseline approach. During this process, we train the student model using standard cross-entropy loss. In addition, building upon the baseline, we proceed to investigate knowledge distillation, a popular technique for transferring knowledge from a large teacher model to a smaller student model. We employ the mean squared error loss to match the soft labels generated by the teacher with the corresponding soft labels produced by the student. For each evaluated model, we performed 5 training runs and we report the average accuracy.

B. Experimental Evaluation

For the CIFAR-10 dataset, the advantage of our proposed approach lies in its consistent improvements over the knowledge distillation method beyond the 30th epoch, as shown in Fig. 1. In particular, we have observed a mean improvement of 0.7% compared to the scenario where the model was solely trained using supervised learning. In contrast, the mean improvement achieved with conventional knowledge distillation is 0.15%. Regarding the TinyImagenet dataset, the knowledge distillation approach yields an increase of 2.3%, as illustrated in Fig. 2. In contrast, the proposed method achieves a remarkable improvement of 6.3%, demonstrating its notable effectiveness.

It is worth noting that the model trained without knowledge distillation produced inferior results, reaffirming the crucial role of knowledge distillation in enhancing the student model's performance. By leveraging the teacher model's knowledge, our approach effectively guides the student model towards learning critical features and decision boundaries, leading to a more accurate and refined classification process.

C. Evaluation of Different Layer Weighting Strategies

A key aspect of the proposed method is the selection of weighting factors for the loss for each layer during the knowledge transfer process. We have meticulously experimented

with various weight factors for individual layers, recognizing that different layers play distinct roles during training [12].

The results are presented in Fig. 3, where we present findings from three distinct models. In the first model, all layers are assigned equal weights of 1 ("Model 1"). For the second model ("Model 2") we wanted to examine the effect of having initial layers with minimal impact in order to enhance knowledge transfer in layers closer to the final decision (output). To this end, we linearly distribute the weights starting from 1.0 for the 12th layer to 0.1 for the 3rd layer and before. Finally, in the third model ("Model 3"), we emphasize the fundamental role of the first two layers in comprehending input features. Consequently, we assign unity weights to these two first layers, while applying the same weight-reduction logic as the second model to the remaining layers. We also attach particular importance to the last layer of the student model, as it produces the crucial output that feeds into the fully connected network responsible for classification. Hence, we use a weight factor of one upon this ultimate layer to ensure its accurate alignment with the teacher's knowledge. The weights used in these three setups for the different layers are summarized in Table II. The evaluation results reported in Fig. 3, demonstrate that this final strategy, which is the one employed for the experiments conducted in this paper, facilitates a finely tuned knowledge distillation process, where information flow is optimized for each layer's unique contributions to the model's learning, as also first noted in [12].

IV. CONCLUSION

This paper introduced and investigated the use of Probabilistic Knowledge Transfer (PKT) for knowledge distillation within Vision Transformers models. By extending knowledge transfer across all layers of the model architecture, PKT diverges from conventional techniques and employs probabilistic modeling techniques to capture the intricate data geometry. To this end, a patch-based knowledge distillation

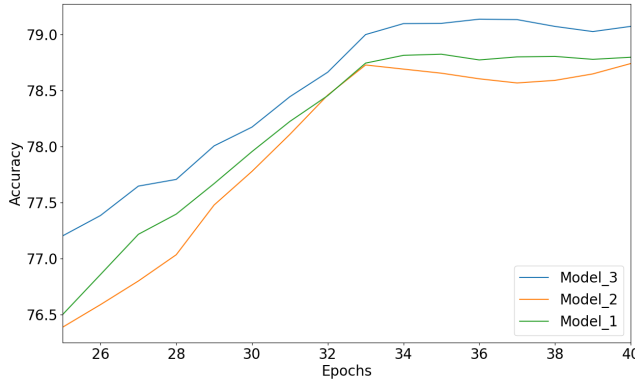


Fig. 3. The figure presents accuracy results on the CIFAR-10 dataset using for the three different weighting strategies for the impact of different layers in the knowledge distillation process.

TABLE II
DISTILLATION WEIGHTS α_l FOR DIFFERENT LAYERS AND SETUPS, WHICH CORRESPOND TO MODELS OF FIG. 3.

Layer	Setup 1	Setup 2	Setup 3
1	1	0.1	1
2	1	0.1	1
3	1	0.1	0.1
4	1	0.2	0.2
5	1	0.3	0.3
6	1	0.4	0.4
7	1	0.5	0.5
8	1	0.6	0.6
9	1	0.7	0.7
10	1	0.8	0.8
11	1	0.9	0.9
12	1	1	1

approach was developed building upon PKT and targeting Vision Transformers architectures. We proposed separately distilling the knowledge encoded by tokens that correspond to different patches, overcoming the aforementioned difficulties. The effectiveness of the proposed approach was demonstrated through comprehensive experiments on CIFAR-10 and Tiny-Imagenet datasets, as well as experimental studies that demonstrate the impact of various design choices on the performance of the proposed approach.

The proposed method opens several interesting future research directions. First, can the factors used for weighting the different layers during the distillation process be adjusted automatically by observing the information flow in the network and learning stage? To this end, meta-learning approaches could also be employed [16]. Furthermore, the proposed method can be expanded to handle large language models based on

Transformer-like architectures [17], or even Language-Vision-Action models used in robotics applications [18].

REFERENCES

- [1] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proceedings of the International Conference on Learning Representations*, 2021.
- [2] Q. Xu, X. Wei, Y. Lu, H. Wang, P. Zhao, and X. Liu, "Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.
- [3] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *NIPS Deep Learning and Representation Learning Workshop*, 2015.
- [4] J. Kim, D. Joo, J. Kim, and J.-W. Ha, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7130–7138.
- [5] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1789–1819, 2021.
- [6] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," in *Proceedings of the International Conference on Learning Representations*, 2015.
- [7] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4133–4141.
- [8] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2215–2223.
- [9] T. Wolf, V. Sanh, J. Chaumond, C. Delangue, and A. Moi, "Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019.
- [10] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, and Q. Liu, "Tinybert: Distilling bert for natural language understanding," in *Proceedings of the Findings of the Association for Computational Linguistics*, 2020.
- [11] N. Passalis and A. Tefas, "Learning deep representations with probabilistic knowledge transfer," in *Proceedings of the European Conference on Computer Vision*, 2018.
- [12] N. Passalis, M. Tzelepi, and A. Tefas, "Heterogeneous knowledge distillation using information flow modeling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.
- [13] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," in *2015 IEEE Information Theory Workshop*, 2015, pp. 1–5.
- [14] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.
- [15] Y. Le and X. Yang, "Tiny imagenet visual recognition challenge," *CS 231N*, vol. 7, no. 7, p. 3, 2015.
- [16] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, "Meta-learning in neural networks: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 5149–5169, 2021.
- [17] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving language understanding by generative pre-training," 2018.
- [18] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *Proceedings of the Conference on Robot Learning*, 2023, pp. 2165–2183.