ICCV #XXXXX

GTA: Guided Transfer of Spatial Attention from Self-supervised Models

Anonymous ICCV submission

Paper ID XXXXX

Abstract

Recently, self-supervised learning has enabled the pre-training of vision transformers (ViT) using vast amounts of unlabeled data to obtain rich representations. Using well-trained representations in transfer learning can lead to better performance and faster convergence compared to training from scratch. However, even if such good representations are transferred, a model can easily overfit the limited training dataset and lose the characteristics of the transferred representations. This phenomenon is more severe in ViT, which has low inductive bias. Through experimental analysis using attention maps in ViT, we observe that the rich representations deteriorate when trained on a small dataset. Motivated by this finding, we propose a novel and simple regularization method for ViT called guided transfer of spatial attention (GTA). Our proposed method regularizes the self-attention maps between source and target models. Through this explicit regularization, a target model can fully exploit the knowledge related to object localization properties. Our experimental results show that the proposed GTA consistently improves the accuracy across five benchmark datasets especially when the number of training data is small. As far as we know, there has been no previous study to improve transfer learning performance, specifically considering the ViT architecture.

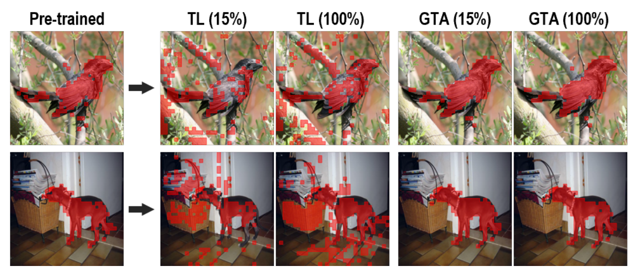

Figure 1. **Comparison of self-attention maps from pre-trained, naïvely fine-tuned, and GTA-traind models.** The self-attention maps of the multiple heads are aggregated with max values, and visualized in red color. Each column shows the attention maps from the models that are pre-trained using SSL, fine-tuned, and fine-tuned with GTA on 15% and 100% of training data, respectively. GTA shows that it is capable of fully leveraging object-centric representations learned by the SSL model.

1. Introduction

The Vision Transformer (ViT) has demonstrated impressive performance in a variety of computer vision tasks such as image classification [11, 35, 32, 34, 24, 39, 23], segmentation [34, 24, 23, 39], object detection [24, 23, 39], and image generation [6, 31, 41], surpassing traditional convolutional neural networks (CNNs). Unlike CNNs that rely entirely on convolution operations which are designed to capture locality, neighborhood structure, and translation equivariance, only the multi-layer perceptron (MLP) component in ViT is responsible for learning those characteristics. The main difference between ViT and CNNs is the self-attention mechanism in the multi-head self-attention (MSA) layer, which globally aggregates spatial features from input tokens with normalized importance [11]. ViT is known to have a lower inductive bias compared to CNNs, meaning that it requires more training data to obtain a well-performing model. As a result, when the available training data is limited, ViT generally shows lower performance than CNNs [21]. In a recent study [29], the authors argued that MSA has both advantages and disadvantages. The advantage is its ability to flatten the loss landscape, which can improve accuracy and robustness in large data regimes. On the other hand, the disadvantage is that MSA allows the negative Hessian eigenvalues when trained on limited training data. These negative Hessian eigenvalues can lead to a non-convex loss landscape, which can disturb model training. The study also demonstrated that self-attention can be interpreted as a *large-sized* and *data-specific* spatial kernel [29].

When training data is scarce, transfer learning (TL) has been considered as the de-facto paradigm in practice. Pre-trained models, which have been trained with supervised learning (SL) on large-scale datasets, have enabled faster training and high generalization performance in TL scenarios. Such SL models possess rich discriminative features that are effective in distinguishing between images, by using class labels during training. However, since the features are optimized for a specific large-scale dataset (e.g., ImageNet), they may not be as effective for various downstream datasets. For example, pre-trained models trained with a large-scale dataset consisting of animal images may not be suitable for downstream tasks in the medical domain. To maximize its effectiveness, large-scale datasets with labels should be readily available, and the domain of downstream data should be similar to that of pre-training data. Consequently, the conventional strategy of transferring the SL backbone has inherent limitations in terms of its applicability to a wide spectrum of downstream tasks.

Recently, self-supervised learning (SSL) has emerged as a promising alternative for learning visual representations without using class labels. Unlike SL, which focuses primarily on discriminative features, SSL can establish its own pretext tasks to produce richer representations that are helpful in describing the semantics of objects in images. Studies on SSL have demonstrated better TL performance than SL in various downstream tasks such as classification [17, 7, 9, 15, 44, 45, 16, 2, 12], localization [17, 15, 44, 45, 16], and segmentation [17, 15, 4, 44, 45, 16]. In addition, SSL enables to obtain the domain-oriented representations by training an unlabeled large-scale dataset related to the target domain of interest, e.g., SSL on large-scale medical images [3]. With these advantages, SSL can serve as a powerful alternative to SL, helping to address the domain discrepancies in various TL scenarios. The ViT architecture has recently proven advantageous for SSL due to its ability to fully leverage large-scale datasets. In particular, some studies have demonstrated high TL performance by utilizing accurate object-centric representation features that can be also helpful for semantic segmentation [4, 44]

Various TL techniques have been proposed to effectively learn target tasks by utilizing well-trained representations transferred from pre-trained models [28, 37, 8, 38, 33]. However, the majority of existing knowledge-exploiting methods are designed for CNNs [28, 37, 8, 38], and there are few effective TL methods that can leverage the characteristics of ViT [33]. When applying commonly used TL techniques to ViT, the object-centric representations from well-trained models may deteriorate. We experimentally confirmed that the quality of well-trained SSL features deteriorates after fine-tuning based on the visualization of self-attention maps from fine-tuned ViT models, and assessed the influence of the amount of training data (see Figure 1). Through the self-attention maps, we can visually see which image tokens are particularly attended to perform the target task. As shown in Figure 1, visualization results indicate that ViT trained with basic fine-tuning tends to overfit to the features corresponding to the background (i.e., non-object area). Even with a relatively sufficient amount of training data, ViT still focuses on non-object regions due to its low inductive bias. Motivating by this observation, we hypothesize that TL performance can be improved if we can prevent the degradation of attention quality of pre-trained SSL models.

In this paper, to address this issue, we propose the Guided Transfer of spatial Attention (GTA) method that effectively leverages pre-trained knowledge that contains object-centric attention to enhance TL performance of ViT, even with the limited size of the training dataset. Specifically, we explicitly regularize self-attention logits of a downstream network (i.e., a target network) through a simple squared L_2 distance. Using various benchmark datasets, we compare our proposed GTA with existing TL methods including a method designed specifically for ViT [33] to demonstrate its superiority over comparison targets. To evaluate the effectiveness and importance of guiding self-attention, we compare the performance of guiding other output features from ViT, e.g., outputs of MSA layers or transformer blocks. In addition, we experimentally evaluate whether we can expect a performance boost when GTA is used in conjunction with TransMix [5], a label-mixing augmentation method specifically designed for ViT based on attention scores. It differs from Mixup [42] and CutMix [40] which determine augmented labels based on randomly sampled mixing coefficients between two images. Finally, we evaluate the factors that can affect the performance of GTA including the use of SL as a guide model.

Our main contribution can be summarized as follows:

- We propose a simple yet effective TL technique for ViT named GTA. Our proposed GTA effectively improves performance by explicitly guiding self-attention logits. To the best of our knowledge, no prior work has proposed to improve the TL performance through a specific focus on the ViT architecture, particularly the MSA component.

- We demonstrate that as the amount of training data decreases, the likelihood of self-attention deviating from the pre-trained model and concentrating on non-object regions increases. Our experimental results show the critical importance of guiding self-attention during ViT training in TL settings, particularly when the amount of training data is limited.

2. Related Work

Transfer learning. TL is the most common and popular method in deep learning that can be applied to various downstream tasks [1, 14]. It not only improves performance but also ensures fast convergence of training by utilizing pre-trained models [18]. Some studies have proposed methods to exploit the pre-trained knowledge and improve performance by regularizing features [22, 8]. DELTA measures the importance of feature channels in the CNN model and

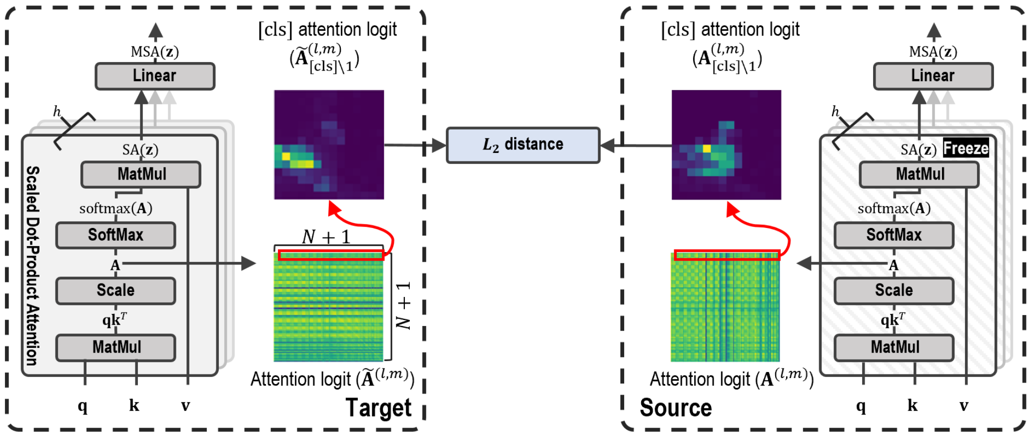

Figure 2. **The overall pipeline of the proposed GTA.** An image is first fed into both the frozen source model and the trainable target model. By minimizing the L_2 distance between the attention logits from each model, the target model is optimized for the current task while focusing on the image tokens that require attention by exploiting the source model.

regularizes the channels far from the pre-trained activations to leverage transferred knowledge [22]. BSS shows that small eigenvalues of transfer features cause negative transfer, and penalizing small eigenvalues during TL to suppress untransferable spectral components can improve performance [8]. Another method of exploiting prior knowledge is weight-based regularization, which controls the weight changes during downstream training [28, 37]. L_2 regularization penalizes changes in model weights [28], and L_2-SP utilizes L_2 constraints on the weights by using the pre-trained model as the starting point to leverage the learned inductive bias [37]. Co-tuning [38] has shown impressive performance improvements by leveraging the label relationship between the upstream and downstream tasks. However, in this work, to ensure ease of implementation and scalability, we only focus on methods that do not require additional data or pre-processing steps for training [22, 38]. Hence, we exclude previous approaches that utilize label information, as they cannot be used with annotation-free pre-trained models such as SSL. While many studies on TL have focused on CNNs, few studies have investigated the performance of TL with ViT [33]. In [33], it is shown that fine-tuning only the MSA layers can improve performance compared to full fine-tuning.

Self-supervised learning. SSL has received considerable attention due to its ability to learn meaningful representations without requiring human annotations [17, 7, 9, 15, 4, 44, 45, 16, 2, 12]. This is accomplished by engaging in self-imposed pretext tasks such as contrastive learning [7, 17], utilizing the teacher-student framework [4, 15], predicting pixels of masked patches [16] and a combination of pretext tasks [44, 45, 2]. Especially, there are two interesting SSL methods, DINO [4] and iBOT [44], that can provide valuable object-centric representations with ViT. DINO utilizes a distillation-based pretext that enables a model to understand the semantic layout of scenes. iBOT combines the masked image modeling task and pretext task used in DINO, and has shown improved attention quality and performance over DINO. However, there are few studies on how to effectively transfer those well-trained representations of ViT.

3. Method

This section presents our proposed approach, which aims to fully exploit the SSL representations from ViT for effective TL to unseen target datasets. We first provide a brief summary of the computations involved in ViT and then introduce the proposed GTA method.

3.1. Preliminaries

ViT consists of a stack of transformer blocks, each of which contains MSA and feed-forward layers. Let $\mathbf{z} \in \mathbb{R}^{(N+1)\times D}$ be input features of a specific transformer block, where N denotes the number of input features corresponding to image patches and D represents the dimensionality of features. Note that $\mathbf{z}$ has one extra dimension since the extra learnable `[cls]` token is typically used to aggregate patch-level features. The value of N can be calculated as $N = HW/P^2$, where H and W denote the height and width of an image, respectively, and P represents the size of patches.

The MSA layer computes a weighted sum of value embeddings, where the weights are computed with query and key embeddings. For a single attention head, these embeddings are obtained by the associated weights $\mathbf{W_q}$, $\mathbf{W_k}$, and $\mathbf{W_v}$, respectively. Specifically, a query $\mathbf{q}$, a key $\mathbf{k}$, and a value $\mathbf{v}$ are given by:

$$\mathbf{q} = \mathbf{z}\mathbf{W_q}, \mathbf{k} = \mathbf{z}\mathbf{W_k}, \mathbf{v} = \mathbf{z}\mathbf{W_v}, \tag{1}$$

i.e., $\mathbf{q}$, $\mathbf{k}$, and $\mathbf{v}$ are all $(N+1) \times k$ dimensional matrices where k denotes an embedding dimension of a single attention head. Typically, k is set to D/h when MSA has h attention heads. By computing a scaled dot product between q and k, we can obtain **the attention logit matrix A** as follows:

$$\mathbf{A} = \mathbf{q}\mathbf{k}^T/\sqrt{k}, \quad \mathbf{A} \in \mathbb{R}^{(N+1)\times(N+1)}. \tag{2}$$

It should be noted that this attention logit plays a crucial role in our GTA. Then, the output features $\mathrm{SA}(\mathbf{z}) \in \mathbb{R}^{(N+1)\times k}$ can be obtained by $\mathrm{softmax}(\mathbf{A})\mathbf{v}$ where $\mathrm{softmax}(\cdot)$ applies the softmax operation to every row of a matrix. Finally, MSA aggregates the outputs from h attention heads using the weight $\mathbf{W_{proj}} \in \mathbb{R}^{(h\cdot k)\times D}$ to compute the final MSA output:

$$\mathrm{MSA}(\mathbf{z}) = [\mathrm{SA}_1(\mathbf{z}), \cdots, \mathrm{SA}_h(\mathbf{z})]\mathbf{W_{proj}}. \tag{3}$$

Finally, position-wise feed-forward layers are employed to generate output features $\mathbf{z}'$ of a transformer block from $\mathrm{MSA}(\mathbf{z})$. Note that we have excluded layer normalization to simplify the explanation.

3.2. Spatial Attention Guidance

Inspired by the findings that ViT models pre-trained on large-scale datasets using SSL show remarkable foreground localization capabilities, and that MSA facilitates spatial mixing of input features, we propose a simple yet effective TL strategy that is tailor-made for ViT.

Given the attention logit matrix $\mathbf{A}^{(l,m)}$ (Eq. 2) of the l-th head in m-th transformer block, we focus on the attention logit values that relate to the `[cls]` token query. More specifically, given $\mathbf{A}^{(l,m)} = [\mathbf{A}^{(l,m)}_{\texttt{[cls]}}; \mathbf{A}^{(l,m)}_1; \cdots; \mathbf{A}^{(l,m)}_N]$, we only consider the `[cls]` attention vector, excluding the first element (which is simply a scaled norm of the `[cls]` query vector), denoted as $\mathbf{A}^{(l,m)}_{\texttt{[cls]}\setminus 1}$. This attention vector contains valuable information on which input patches should be attended to perform a given task.

Assuming that $\mathbf{A}^{(l,m)}_{\texttt{[cls]}\setminus 1}$ offers robust spatial mixing coefficients, leveraging this knowledge for TL on downstream tasks can be achieved through a straightforward implementation of constrained optimization, with the constraint that fine-tuned attention logits should be similar to those of initial models (e.g., pre-trained SSL models):

$$\min \mathcal{L}_{\mathrm{CE}} \quad \text{s.t. } \mathbf{A}^{(l,m)}_{\texttt{[cls]}\setminus 1} \approx \tilde{\mathbf{A}}^{(l,m)}_{\texttt{[cls]}\setminus 1} \quad \forall\, l, m \tag{4}$$

where $\mathcal{L}_{\mathrm{CE}}$ represents the cross entropy loss and $\tilde{\mathbf{A}}$ denotes an attention logit matrix of a target model trained during fine-tuning. To this end, we employ a simple squared L_2 distance for the constraint. Therefore, given a coefficient λ, our objective function $\mathcal{L}$ during fine-tuning reduces to:

$$\mathcal{L} = \mathcal{L}_{\mathrm{CE}} + \lambda \sum_{l,m} \left\| \mathbf{A}^{(l,m)}_{\texttt{[cls]}\setminus 1} - \tilde{\mathbf{A}}^{(l,m)}_{\texttt{[cls]}\setminus 1} \right\|_2^2 \tag{5}$$

Our regularization term, GTA, can be interpreted as transferring spatial kernels from a pre-trained model to a target model. That is, the target model tries to learn how to mix channel information while preserving the similarity of spatial mixing coefficients to those of the pre-trained model. It is worth noting that although GTA is motivated by the localization property of SSL models, it is also effective in TL with SL models since it allows the target model to selectively utilize pre-trained features.

Dataset	# category	# train	# test
CUB [36]	200	5994	5794
Cars [20]	196	8144	8041
Aircraft [26]	100	6667	3333
Dogs [19]	120	12000	8580
Pet [30]	37	3680	3669

Table 1. **Overview of dataset statistics.** Table shows the number of classes, and training and test images of each dataset used in our experiments.

4. Experimental Results

In this section, we evaluate the effectiveness of our method across multiple fine-grained datasets, which serve as standard benchmarks for assessing TL performance. Our experiments highlight the significance of applying regularization to the attention logits of the `[cls]` token. We also present segmentation results that demonstrate how the attention logits of the target model focus on objects that are relevant to the target task, rather than merely duplicating those of the source model. Furthermore, we assess the synergies between our method and the recent augmentation technique TransMix [5] that utilizes attention outputs in ViT. Finally, we conduct an ablation study to investigate the impact of key factors on the performance of our proposed method.

Datasets. We employ five widely used fine-grained datasets: CUB-200-2011 (CUB) [36], Stanford Cars (Cars) [20], FGVC-Aircraft (Aircraft) [26], Stanford Dogs (Dogs) [19], and Oxford-IIIT Pet (Pet) [30], which contain birds, cars, airplanes, dogs, and pets, respectively. Table 1 shows the data statistics for the datasets. We conduct experiments using four different configurations based on the amount of training data following [8, 38]. Each configuration consists of a varying percentage of randomly selected training samples for each category: 15%, 30%, 50%, and 100%.

Training configurations. We follow DINO fine-tuning configurations [4] and apply them across all methods, including the baseline (i.e., naïve fine-tuning). All methods are trained using AdamW optimizer with a momentum of

		Sampling Rates [Acc@1]			
Dataset	**Method**	**15%**	**30%**	**50%**	**100%**
CUB	Fine-tune (baseline)	41.376 ± 0.415	62.697 ± 0.552	75.158 ± 0.369	84.444 ± 0.166
	L_2-SP [37]	41.554 ± 1.020	63.261 ± 0.640	75.371 ± 0.345	84.898 ± 0.274
	BSS [8]	41.382 ± 0.787	62.870 ± 0.343	75.406 ± 0.147	84.501 ± 0.320
	Attention only (freeze FFN) [33]	42.636 ± 0.582	62.686 ± 0.511	75.175 ± 0.036	85.048 ± 0.232
	FFN only (freeze attention) [33]	37.349 ± 0.901	58.181 ± 0.121	71.839 ± 0.217	82.902 ± 0.138
	GTA	**51.525 ± 0.449**	**68.416 ± 0.419**	**78.058 ± 0.089**	**85.543 ± 0.320**
Cars	Fine-tune (baseline)	56.100 ± 0.675	78.502 ± 0.167	87.091 ± 0.132	93.065 ± 0.093
	L_2-SP [37]	56.676 ± 0.783	78.713 ± 0.316	87.257 ± 0.168	**93.276 ± 0.038**
	BSS [8]	56.154 ± 0.718	78.796 ± 0.131	87.170 ± 0.050	93.206 ± 0.044
	Attention only (freeze FFN) [33]	56.701 ± 0.521	77.872 ± 0.233	86.747 ± 0.256	92.414 ± 0.000
	FFN only (freeze attention) [33]	51.171 ± 0.799	75.418 ± 0.386	85.769 ± 0.273	92.671 ± 0.059
	GTA	**59.271 ± 0.248**	**79.488 ± 0.202**	**87.651 ± 0.111**	93.239 ± 0.097
Aircraft	Fine-tune (baseline)	52.115 ± 0.412	68.447 ± 0.647	76.848 ± 0.330	86.939 ± 0.076
	L_2-SP [37]	51.645 ± 0.465	68.777 ± 0.666	76.978 ± 0.625	**87.209 ± 0.121**
	BSS [8]	52.285 ± 0.291	68.677 ± 0.692	76.998 ± 0.330	87.129 ± 0.369
	Attention only (freeze FFN) [33]	50.735 ± 1.379	67.477 ± 0.505	76.098 ± 0.362	85.639 ± 0.522
	FFN only (freeze attention) [33]	51.195 ± 0.243	67.207 ± 0.390	75.198 ± 0.392	85.399 ± 0.809
	GTA	**54.635 ± 0.572**	**70.027 ± 0.778**	**77.548 ± 0.632**	86.989 ± 0.191
Dogs	Fine-tune (baseline)	59.775 ± 0.256	72.137 ± 0.220	78.131 ± 0.037	83.318 ± 0.007
	L_2-SP [37]	63.893 ± 0.477	75.715 ± 0.603	81.453 ± 0.338	85.264 ± 0.186
	BSS [8]	59.817 ± 0.303	72.253 ± 0.087	78.155 ± 0.219	83.570 ± 0.251
	Attention only (freeze FFN) [33]	62.747 ± 0.455	74.577 ± 0.298	80.113 ± 0.114	84.938 ± 0.205
	FFN only (freeze attention) [33]	57.502 ± 0.299	70.194 ± 0.095	77.253 ± 0.125	83.182 ± 0.273
	GTA	**69.196 ± 0.222**	**78.054 ± 0.194**	**81.803 ± 0.036**	**85.633 ± 0.192**
Pet	Fine-tune (baseline)	77.342 ± 0.382	86.418 ± 0.433	90.206 ± 0.096	93.123 ± 0.201
	L_2-SP [37]	81.185 ± 0.500	88.871 ± 0.220	92.169 ± 0.299	**94.276 ± 0.439**
	BSS [8]	77.478 ± 0.488	86.572 ± 0.450	90.597 ± 0.206	93.286 ± 0.417
	Attention only (freeze FFN) [33]	81.030 ± 0.666	88.698 ± 0.259	91.832 ± 0.306	93.786 ± 0.166
	FFN only (freeze attention) [33]	74.825 ± 0.886	84.755 ± 0.129	89.697 ± 0.382	92.723 ± 0.142
	GTA	**83.856 ± 0.063**	**89.906 ± 0.197**	**92.478 ± 0.245**	94.022 ± 0.246

Table 2. **Comparison of transfer learning methods.** The baseline refers to the naïvely fine-tuned model. "Attention only" and "FFN only" represent training of only attention layers and feed-forward network (FFN), respectively. GTA shows higher accuracy across all datasets and all sampling rates, with particularly significant improvements when the training data is limited. The best results are bold-faced.

0.9 during 3k iterations, and the learning rate is decreased by cosine annealing scheduler [25]. We set the batch size, weight decay, and initial learning rate to 768, 0.05, and 0.0001, respectively. Input images are resized to 224×224. RandAugment [10] is employed for augmentation. However, we do not use random erasing [43] since self-attention layers heavily focus on the areas erased by random erasing, which may lead to inaccurate attention guidance. All experiments are conducted with the ViT-small architecture. All weights are initialized with the ImageNet-1k pre-trained checkpoint of iBOT. We repeat each experiment three times with different random seeds to report performance variations.

4.1. Transfer Learning Performance

Firstly, we compare our method and previous TL methods (see Table 2) to verify their compatibility with ViT. Also, we evaluate the effectiveness of GTA in leveraging object-centric representations. To make the comparison as fair as possible, we mostly use the hyperparameter settings reported in each paper, but a regularization coefficient λ is tested with three values based on the default values of each TL method. Specifically, we train models with $0.1\times\alpha$, α, and $10\times\alpha$ when α is the default value. We report the best performance among the results obtained using three different λ values.

At the smallest sampling rate setting (i.e. 15%), GTA can significantly enhance performance compared to the baseline for all datasets. Specifically, each dataset shows an improvement of at least 2.52% and up to 10.15%. When the training data is insufficient, ViT tends to attend more to the background instead of foreground objects, making it challenging to classify images with different backgrounds in the test dataset. However, GTA addresses this issue by explicitly regularizing the attention on foreground objects. As the amount of training data increases, the degree of improvement decreases. For example, with the CUB dataset, the gaps between GTA and baseline are decreased as 15%: 10.149, 30%: 5.719, 50%: 2.900, and 100%: 1.099.

We also compare GTA with commonly used TL methods such as L_2-SP [37], BSS [8], and ViT-specific methods [33]. Our results demonstrate that GTA consistently outperforms comparison methods across all sampling rates, especially in cases where the training dataset is relatively small. Across all target datasets, the gap between GTA and the best-performing previous TL methods ranges from 2.35% to 8.89% at the 15% setting. While this trend remains at the 30% and 50% settings, the difference between GTA and other methods decreases, eventually becoming comparable at the 100% setting. For instance, The L_2-SP shows comparable results with GTA at the 100% configuration for Cars, Aircraft, and Pet datasets.

The L_2-SP is the most explicit and simple method to take advantage of a well-trained source model. However, it is uncertain whether the combination of ViT with L_2-SP, a method optimized for CNNs, is the reason for the relatively lower accuracy improvement. The BSS method has the advantage of excluding negative features from the pre-trained model, but it lacks regularization terms to leverage transferred knowledge, making it prone to overfitting to the target task, similar to the baseline. According to [33], training only attention layers yields better performance than end-to-end fine-tuning. While it is also observed in our experiments, the method shows lower performance than GTA. Similarly, the FFN-only method, which freezes the attention layers from the pre-trained model, shows poor performance since the frozen attention cannot be adapted to the target task.

4.2. The Importance of Attention Logits

Table 3 shows the importance of guiding attention logits compared to using other two outputs, the transformer block output $\mathbf{z}'$ and MSA output $\mathrm{MSA}(\mathbf{z})$ in ViT. We use L_2 regularization to those two outputs following Equation 5. Our experiments show that GTA outperforms the regularization of other outputs across all sampling rates and datasets. Such variants without careful consideration can lead to an acceleration of negative transfer. The guidance based on attention logits may not have a direct impact on training, but it would provide an appropriate inductive bias conditioned on well-trained representations, emphasizing only the areas that the model should attend to.

4.3. Segmentation Performance

In this experiment, we compare the segmentation results calculated by the GTA model with those of the SSL source model and fine-tuned model by evaluating segmentation performance on the PASCAL-VOC12 validation set based on the Jaccard index [13], following [4, 44, 27]. The visualization results show that the segmentation results from GTA are more accurate in focusing on the foreground object, as shown in Figure 3. Quantitatively, the GTA model also shows a higher Jaccard index compared to others (see Table 4). The fine-tuned model focuses on specific parts of the foreground but also attends to a significant amount of irrelevant background information. The SSL model performs well, but it also places attention on unimportant areas that are not relevant to the target class. While the segmentation results generated from GTA model do not perfectly replicate those of SSL model, it effectively focuses on the target object of the current task. Through these experiments, guiding based on attention logits has been also verified to be an effective method for focusing on informative areas while en-

		Sampling Rates	
Dataset	**Method**	**15%**	**100%**
CUB	baseline	41.376	84.444
	block output guide	46.859	85.077
	MSA output guide	46.519	84.904
	Attention logits (GTA)	**51.525**	**85.543**
Cars	baseline	56.100	93.065
	block output guide	58.960	93.098
	MSA output guide	59.039	93.023
	Attention logits (GTA)	**59.271**	**93.239**
Aircraft	baseline	52.115	86.939
	block output guide	54.485	86.999
	MSA output guide	54.225	87.039
	Attention logits (GTA)	**54.635**	**86.989**
Dogs	baseline	59.775	83.318
	block output guide	65.299	84.755
	MSA output guide	65.078	84.740
	Attention logits (GTA)	**69.196**	**85.633**
Pet	baseline	77.342	93.123
	block output guide	82.875	93.913
	MSA output guide	82.666	93.877
	Attention logits (GTA)	**83.856**	**94.022**

Table 3. **Effectiveness of different features for guidance.** The block output and MSA output guide indicate the guidance between source and target model with the transformer block output and the MSA layer output, respectively. Our proposed method, GTA, provide guidance to target model using attention logits. The proposed method shows higher accuracy across all dataset and sample rates. Best results are bold-faced.

Method	**Jarccard index**
baseline	0.367
pre-trained (SSL)	0.386
GTA	**0.399**

Table 4. **Quantitative evaluation of attention map guidance on segmentation task.** Baseline refers to simple fine-tuning, pre-trained denotes SSL models not yet train for the target task. The proposed GTA outperformed the others in terms of Jaccard index on PASCAL-VOC12 validation set. Best results are bold-faced.

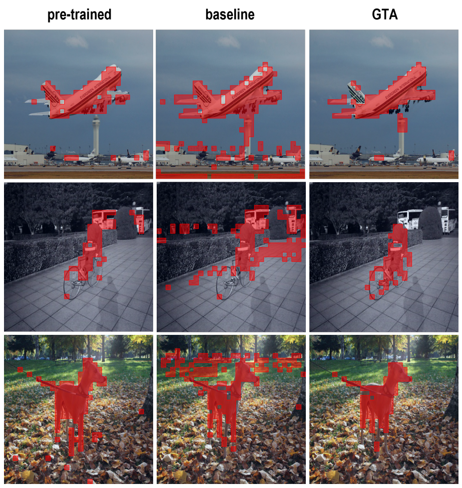

Figure 3. **Comparison of segmentation results on PASCAL-VOC12.** Pre-trained refers to the segmentation results obtained by the attention logits of the upstream SSL. Baseline represents the results obtained by fine-tuning the pre-trained model to target task. GTA denotes the results obtained by utilizing the GTA during fine-tuning. GTA shows optimized performance compared to the other results.

Dataset	Method	Sampling Rates 15%	100%
CUB	baseline	41.376	84.444
	baseline + TransMix	42.032	84.703
	GTA	51.525	85.543
	GTA + TransMix	54.361	85.755
Cars	baseline	56.100	93.065
	baseline + TransMix	56.117	93.139
	GTA	59.271	93.239
	GTA + TransMix	59.943	93.218
Aircraft	baseline	52.115	86.939
	baseline + TransMix	52.455	86.819
	GTA	54.635	86.989
	GTA + TransMix	55.166	87.369
Dogs	baseline	59.775	83.318
	baseline + TransMix	60.229	83.551
	GTA	69.196	85.633
	GTA + TransMix	70.004	85.793
Pet	baseline	77.342	93.123
	baseline + TransMix	77.396	93.268
	GTA	83.856	94.022
	GTA + TransMix	84.937	94.067

Table 5. **Quantitative evaluation of the boosting effect.** Baseline refers to the fine-tuned model without TransMix or GTA. +TransMix denote add TransMix augmentation on tranining. The combination of GTA and TransMix outperformed both the baseline and GTA alone. Best results are bold-faced.

suring the model to be optimized to the current target task.

4.4. Boosting Effect of Attention Guidance

As demonstrated in our previous experiment, we show that GTA improves the localization quality of the self-attention logits on the target object. To capitalize on this advantage, we investigate whether a boosting effect could be achieved by combining GTA with TransMix [5]. TransMix involves mixing images in a similar manner to CutMix [40], but without using the size ratio of the cropped box as a new label. Instead, a new label is calculated based on the self-attention ratio between the mixed images. Therefore, the effectiveness of TransMix relies on the ability of the target model to generate proper attention that is accurately focused on the foreground object. However, the authors argue that an attention map that accurately localizes objects cannot improve the performance of TransMix. It is based on the finding from the experiment using DINO as a parameter-frozen external model. The parameter-frozen external model has a limitation in that it can only generate mixing labels in a static manner, regardless of the training. In contrast, our proposed method allows for dynamic mixing labels while incorporating improved attention from an external model. This is because the parameter-frozen external model guides only the attention logit of the target model.

According to Table 5, TransMix shows better performance when it is combined with GTA rather than when it is used with the baseline. The gap between baseline and baseline+TransMix and between GTA and GTA+TransMix is significantly increased when the sampling rate is small. When trained with a small dataset, the background attention issue, as visualized in Figure 1, can hinder TransMix from generating the proper labels. However, as the amount of training data increases, the effect of attention improvement by GTA decreases, and consequently the boosting effect is also reduced. Since the combination of TransMix and GTA shows better results compared GTA alone, it demonstrates that GTA can be combined with other regularization methods to further improve the results.

4.5. Ablation Study

The performance of GTA can be influenced by two main factors: the selection of the pre-trained weight used as the source model and the appropriate regularization coefficient λ. In this section, we analyze these factors in detail.

Selection of guidance model. GTA is the method that guides the training of the target model using the source model. Therefore, the choice of which weights to use as the source model can affect the performance of GTA. In this experiment, we compare the performance of using SSL models and the commonly used SL model as the source model. Our results show that GTA consistently improves accuracy across all datasets, whether applied to SL or SSL

Dataset	Method	Sampling Rates 15%	100%
CUB	baseline (SL)	51.519	85.548
	GTA (SL)	**62.047**	**85.663**
	baseline (SSL)	41.376	84.444
	GTA (SSL)	51.525	85.543
Cars	baseline (SL)	45.894	91.382
	GTA (SL)	47.822	90.930
	baseline (SSL)	56.100	93.065
	GTA (SSL)	**59.271**	**93.239**
Aircraft	baseline (SL)	48.355	82.638
	GTA (SL)	49.635	82.558
	baseline (SSL)	52.115	86.939
	GTA (SSL)	**54.635**	**86.989**
Dogs	baseline (SL)	74.872	87.945
	GTA (SL)	**88.897**	**91.682**
	baseline (SSL)	59.775	83.318
	GTA (SSL)	69.196	85.633
Pet	baseline (SL)	81.466	93.123
	GTA (SL)	**91.524**	**94.967**
	baseline (SSL)	77.342	93.123
	GTA (SSL)	83.856	94.022

Table 6. **Comparison of GTA performance using different source model weights.** GTA consistently improved accuracy on all datasets using both SSL and SL weights as the source model. Best results are bold-faced.

(see Table 6). This suggests that GTA is not dependent on specific SSL weights, but rather can be applied to a variety of pre-trained models. However, there are performance differences depending on which weights are used. When using SL weights, we observe better performance on CUB, Dogs, and Pet datasets, whereas when using SSL weights, we observe better results on Cars and Aircraft compared to SL. These differences can be attributed to domain discrepancies between upstream and downstream data. Since the SL model is trained on ImageNet for classification, CUB, Dogs, and Pet are semantically close to the upstream domain, while Car and Aircraft are farther away, resulting in lower baseline performance. In contrast, SSL models show better generalization performance, leading to better results on Cars and Aircraft despite the fact that SSL is also trained on ImageNet.

Influence of lambda. We test four different λ values (0.1, 1.0, 10.0, 100.0) to find an optimal value for each dataset (see Figure 4). Our findings reveal that the optimal λ is varied depending on the amount of and characteristics of the dataset. Similar to the weight experiments above, we observe that the results of λ are also heavily influenced by the characteristics of the data domain. Specifically, datasets such as CUB, Dogs, and Pet that belong to the near-domain to upstream data show good performance with high λ values. In contrast, datasets such as Cars and Aircraft, belonging to the out-domain, show better results with low λ values. The difference could be attributed to the quality of the self-attention logits used for guidance. In the case of near-domain, even with high λ, the target task can be fitted well with minimal changes in the self-attention logits. However, in the out-domain, a considerable change in the self-attention logits is necessary to learn the target task. Therefore, as the target data are far from the upstream data domain, smaller λ values should be used, but too small λ values might result in overfitting similar to the baseline fine-tuning. As a result, our experiments show that for out-domain datasets, the optimal value of λ is consistently 1.0 regardless of the amount of training data. In contrast, a higher value of λ yields better accuracy as the amount of data decreases for near-domain datasets. At the 15% condition, 100.0 λ is appropriate, but for higher conditions, near 10.0 is found to be the optimal value. Hence, when applying GTA, it is necessary to set a parameter λ based on the characteristics and the amount of target data.

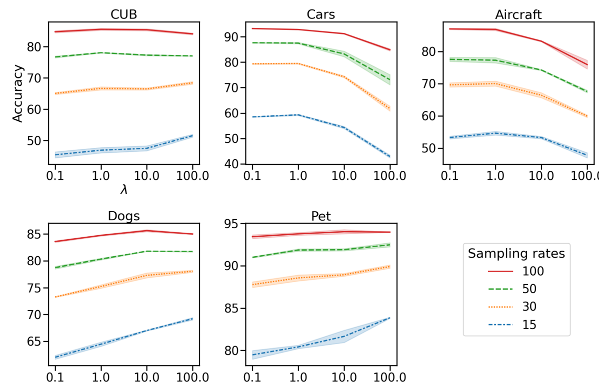

Figure 4. **The effect of different values of λ on GTA.** The optimal lambda value varies depending on the characteristics and amount of the target data.

5. Conclusion

In this paper, we propose a novel transfer learning method called GTA, which effectively utilizes SSL pre-trained knowledge to improve TL performance, specifically for ViT architecture. By applying explicit L_2 regularization between the attention logits of the target and source models, GTA can achieve significant performance improvements across various fine-grained datasets and sampling rates. Through extensive experiments, we show that imposing regularization on the attention logits in ViT is essential, and that GTA outperforms other comparison methods especially when the number of target training data is small. These results demonstrate that GTA is a simple and effective approach for improving the TL performance of ViT.

References

[1] Pulkit Agrawal, Ross Girshick, and Jitendra Malik. Abssnalyzing the performance of multilayer neural networks for object recognition. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII 13*, pages 329–344. Springer, 2014.

[2] Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Florian Bordes, Pascal Vincent, Armand Joulin, Mike Rabbat, and Nicolas Ballas. Masked siamese networks for label-efficient learning. In *European Conference on Computer Vision*, pages 456–473. Springer, 2022.

[3] Shekoofeh Azizi, Basil Mustafa, Fiona Ryan, Zachary Beaver, Jan Freyberg, Jonathan Deaton, Aaron Loh, Alan Karthikesalingam, Simon Kornblith, Ting Chen, et al. Big self-supervised models advance medical image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3478–3488, 2021.

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021.

[5] Jie-Neng Chen, Shuyang Sun, Ju He, Philip HS Torr, Alan Yuille, and Song Bai. Transmix: Attend to mix for vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12135–12144, 2022.

[6] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International conference on machine learning*, pages 1691–1703. PMLR, 2020.

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[8] Xinyang Chen, Sinan Wang, Bo Fu, Mingsheng Long, and Jianmin Wang. Catastrophic forgetting meets negative transfer: Batch spectral shrinkage for safe transfer learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[9] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649, 2021.

[10] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[12] Linus Ericsson, Henry Gouk, and Timothy M Hospedales. How well do self-supervised models transfer? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5414–5423, 2021.

[13] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–308, 2009.

[14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[15] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[16] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.

[17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[18] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4918–4927, 2019.

[19] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *Proc. CVPR workshop on fine-grained visual categorization (FGVC)*, volume 2. Citeseer, 2011.

[20] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013.

[21] Seung Hoon Lee, Seunghyun Lee, and Byung Cheol Song. Vision transformer for small-size datasets. *arXiv preprint arXiv:2112.13492*, 2021.

[22] Xingjian Li, Haoyi Xiong, Hanchao Wang, Yuxuan Rao, Liping Liu, and Jun Huan. Delta: Deep learning transfer using feature map with attention for convolutional networks. In *International Conference on Learning Representations*, 2019.

[23] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12009–12019, 2022.

[24] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer:

Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[25] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

[26] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[27] Muhammad Muzammal Naseer, Kanchana Ranasinghe, Salman H Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. *Advances in Neural Information Processing Systems*, 34:23296–23308, 2021.

[28] Andrew Y Ng. Feature selection, l 1 vs. l 2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, page 78, 2004.

[29] Namuk Park and Songkuk Kim. How do vision transformers work? In *International Conference on Learning Representations*, 2022.

[30] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012.

[31] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International conference on machine learning*, pages 4055–4064. PMLR, 2018.

[32] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. deit. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.

[33] Hugo Touvron, Matthieu Cord, Alaaeldin El-Nouby, Jakob Verbeek, and Hervé Jégou. Three things everyone should know about vision transformers. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV*, pages 497–515. Springer, 2022.

[34] Hugo Touvron, Matthieu Cord, and Herve Jegou. Deit iii: Revenge of the vit. *arXiv preprint arXiv:2204.07118*, 2022.

[35] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 32–42, 2021.

[36] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. 2010.

[37] LI Xuhong, Yves Grandvalet, and Franck Davoine. Explicit inductive bias for transfer learning with convolutional networks. In *International Conference on Machine Learning*, pages 2825–2834. PMLR, 2018.

[38] Kaichao You, Zhi Kou, Mingsheng Long, and Jianmin Wang. Co-tuning for transfer learning. *Advances in Neural Information Processing Systems*, 33:17236–17246, 2020.

[39] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 558–567, 2021.

[40] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[41] Bowen Zhang, Shuyang Gu, Bo Zhang, Jianmin Bao, Dong Chen, Fang Wen, Yong Wang, and Baining Guo. Styleswin: Transformer-based gan for high-resolution image generation. In *Proceedings of the IEEE/CVF sconference on computer vision and pattern recognition*, pages 11304–11314, 2022.

[42] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

[43] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020.

[44] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. Image bert pre-training with online tokenizer. In *International Conference on Learning Representations*, 2022.

[45] Pan Zhou, Yichen Zhou, Chenyang Si, Weihao Yu, Teck Khim Ng, and Shuicheng Yan. Mugs: A multi-granular self-supervised learning framework. *arXiv preprint arXiv:2203.14415*, 2022.