A Task-agnostic Pre-training and Task-aware Pre-training

Pre-training is the process of training a model on a large dataset to learn transferable features before fine-tuning it on a specific downstream task. It is a widely adopted strategy to enhance the performance of deep learning models. There are two main types of pre-training methods: task-agnostic and task-aware.

Task-agnostic pre-training focuses on learning general features without targeting any specific task. These pre-training methods [1, 2, 3, 4, 5] employ generic learning objectives to train models, enabling them to capture general patterns that are transferable across various tasks. Although effective for a wide range of applications, these methods may lack the fine-grained understanding needed for some tasks such as facial action analysis.

Task-aware pre-training, on the other hand, aligns the pre-training process with the downstream tasks' needs. This is achieved by modifying the network architecture, customizing loss functions, or leveraging pre-training data relevant to the downstream task. For example, some methods introduce region-of-interest (ROI) layers [6] or weighted losses [7] to emphasize task-relevant features, while others curated pre-training datasets [8] that align with the domain of the target task.

In conclusion, task-agnostic pre-training establishes a robust foundation for learning general-purpose features, making it suitable for a wide range of applications. In contrast, task-aware pre-training adopts a more specialized approach, enabling the model to capture fine-grained and task-specific patterns by incorporating domain knowledge, tailored objectives, and relevant data. The choice between these two kinds of methods depends on the specific needs of the downstream task and how much task-specific information is available during pre-training.

B Meta-Learning

Meta-learning, or "learning to learn," aims to enable models to quickly adapt to downstream tasks using only a small amount of labeled data. Meta-learning methods can be traced back to early works by [9, 10]. Recently, gradient-based meta learning has gained attention for solving bi-level optimization problems, and has been popularized by MAML [11], as well as subsequent studies [12, 13] for few-shot learning. Currently, meta learning has been transferred to various applications, like neural network teaching [14, 15], learning data augmentation and reweighting strategies [16, 17],and auxiliary task learning [18], showcasing its broad utility.

C Further Details of Calculating Meta-Gradients

The meta-gradients of the matching loss with respect to the adjuster parameters can be written as:

$$\frac{\partial \mathcal{L}_{match}}{\partial \boldsymbol{\lambda}^{(t)}} = \underbrace{\frac{\partial \mathcal{L}_{match}}{\partial \boldsymbol{\lambda}^{(t)}}}_{direct\ grad.} + \underbrace{\frac{\partial \mathcal{L}_{match}}{\partial \hat{\boldsymbol{\theta}}_e^{(t)}(\boldsymbol{\lambda}^{(t)})} \cdot \overbrace{\frac{\partial \hat{\boldsymbol{\theta}}_e^{(t)}(\boldsymbol{\lambda}^{(t)})}{\partial \boldsymbol{\lambda}^{(t)}}}^{Jacobian\ Matrix}}_{indirect\ grad.} \tag{1}$$

The direct gradients are zero in our case. The indirect gradient is the product of (a) the gradients of the matching loss with respect to the parameters of the base model's encoder, and (b) the Jacobian matrix of the encoder's parameters with respect to the adjuster's parameters. Term (a) can be efficiently obtained via standard backpropagation, as it involves differentiating the matching loss with respect to the encoder parameters. However, computing term (b) is more challenging due to the need to differentiate through the inner-loop optimization process. To efficiently compute term (b) and improve the stability of the meta-learning process, we adopt the approach proposed by Lorraine and Duvenaud [19], which leverages implicit differentiation for hyperparameter optimization. Specifically, we use the implicit function theorem (IFT) to re-express the Jacobian matrix term:

$$\frac{\partial \hat{\boldsymbol{\theta}}_e^{(t)}(\boldsymbol{\lambda}^{(t)})}{\partial \boldsymbol{\lambda}^{(t)}} = \left[\frac{\partial^2 \mathcal{L}_{weighted}}{\partial \hat{\boldsymbol{\theta}}_e^{(t)} \partial \left(\hat{\boldsymbol{\theta}}_e^{(t)}\right)^\top} \right]^{-1} \cdot \frac{\partial^2 \mathcal{L}_{weighted}}{\partial \hat{\boldsymbol{\theta}}_e^{(t)} \partial \left(\boldsymbol{\lambda}^{(t)}\right)^\top} \tag{2}$$

Here, the first term denotes the Hessian matrix of the weighted reconstruction loss $\mathcal{L}_{weighted}$ with respect to the encoder parameters, and the second term represents the mixed second-order partial derivatives with respect to the encoder and adjuster parameters.

In practice, the mixed derivatives can be efficiently computed via double backpropagation using modern automatic differentiation frameworks. To approximate the inverse Hessian term, we apply the **Neumann series expansion**:

$$H^{-1} \approx \gamma \sum_{i=0}^{\infty} (I - \gamma H)^i \tag{3}$$

where γ is a scaling factor that controls the convergence and numerical stability. The sufficient condition for convergence is that the spectral radius of $I - \gamma H$ satisfies:

$$\rho(I - \gamma H) < 1 \tag{4}$$

Here, $\rho(\cdot)$ denotes the **spectral radius** of a matrix, i.e., the largest absolute value among its eigenvalues.

Let $H \in \mathbb{R}^{n \times n}$ be a symmetric positive-definite matrix with eigenvalues $\sigma_1 \leq \cdots \leq \sigma_n$, then there exists an orthogonal matrix Q and a diagonal matrix $\Lambda = \text{diag}(\sigma_1, \ldots, \sigma_n)$ such that $H = Q\Lambda Q^T$. The eigenvalues of $I - \gamma H$ are:

$$\text{eig}(I - \gamma H) = \{1 - \gamma\sigma_i\}_{i=1}^{n}$$

and its spectral radius is:

$$\rho(I - \gamma H) = \max\{|1 - \gamma\sigma_1|,\ |1 - \gamma\sigma_n|\} \tag{5}$$

To minimize the spectral radius and ensure convergence, we choose the optimal learning rate γ^* such that:

$$\gamma^* = \frac{2}{\sigma_1 + \sigma_n} \tag{6}$$

and the corresponding spectral radius becomes:

$$\rho(I - \gamma^* H) = \frac{\sigma_n - \sigma_1}{\sigma_n + \sigma_1} < 1 \tag{7}$$

This setting ensures the convergence of the Neumann series while minimizing the approximation error, leading to improved convergence speed and stability in the meta-gradient computation.

D Algorithm

TAME's pre-training strategy is summarized in Alg. 1. Each iteration of the pre-training process consists of two loops: the Inner Loop and the Outer Loop. During these loops, the parameters of the adjuster, validator, and base model are alternately updated to progressively align the learned features with the downstream Facial Action Analysis (FAA) tasks.

Can we update the components in TAME simultaneously? Doing so can lead to trivial solutions. For example, if the base model and the adjuster are optimized at the same time, the adjuster might simply assign zero-valued meta weights to drive the weighted reconstruction loss towards zero, thereby blocking meaningful gradients for the base model.

E Datasets

In the pre-training phase for the three FAA downstream tasks, four datasets are utilized to alternately optimize the parameters of the MAE base model, the validator and the adjuster.Below are the introductions of the datasets.

AffectNet [20] is by far the largest human expression dataset in the wild, containing approximately 300,000 labeled facial images annotated with 7 expression categories (including neutral). In addition, it includes another 300,000 unlabeled facial images. We use only the raw image data to pre-train

Algorithm 1 TAME's Dual-Loop Meta-Optimization Strategy

Input: Unlabeled dataset $\mathcal{S}_u$, fine-tuning dataset $\mathcal{S}_{ft}$, validation dataset $\mathcal{S}_{val}$, and number of iterations T.
Output: Base model parameters $\boldsymbol{\theta}^{(T)}$.
Initialization: Initialize base model parameters $\boldsymbol{\theta}^{(0)}$, adjuster parameters $\boldsymbol{\lambda}^{(0)}$, and validator parameters $\boldsymbol{\phi}^{(0)}$.

for $t = 0$ to $T - 1$ **do**
  Inner Loop:
    Sample $\boldsymbol{x}_u \sim \mathcal{S}_u$, compute the candidate update $\hat{\boldsymbol{\theta}}_e^{(t)}(\boldsymbol{\lambda}^{(t)})$ using Eq. (3) of the main text.
  Outer Loop:
    repeat
      Sample $(\boldsymbol{x}_{ft}, \boldsymbol{y}_{ft}) \sim \mathcal{S}_{ft}$.
      Optimize $\boldsymbol{\phi}^{(t)}$ using Eq. (4) of the main text.
      Compute matching loss $\mathcal{L}_{match}$ on $\mathcal{S}_{val}$.
    until $\mathcal{L}_{match} \to \mathcal{L}^*_{match}(\boldsymbol{\lambda}^{(t)})$, yielding $\boldsymbol{\phi}^{(t+1)}$.
  Sample $(\boldsymbol{x}_{val}, \boldsymbol{y}_{val}) \sim \mathcal{S}_{val}$, and update $\boldsymbol{\lambda}^{(t)} \to \boldsymbol{\lambda}^{(t+1)}$ using Eq. (6) of the main text.
  Update $\boldsymbol{\theta}^{(t)} \to \boldsymbol{\theta}^{(t+1)}$ using Eq. (9) of the main text with sampled $\boldsymbol{x}_u$.
end for
Return $\boldsymbol{\theta}^{(T)}$.

our MAE [4] base model in the Inner Loop. For CLIP pre-training [5], the discrete expression labels are converted into short descriptive sentences to align with its vision-language framework. For example, an image labeled as "happy" is transformed into the textual prompt: "This person has a happy expression."

BP4D [21] dataset is a spontaneous facial AU dataset containing 328 videos from 41 subjects (23 females and 18 males). Each subject participates in 8 sessions, during which their spontaneous facial expressions are recorded. The presence or absence of 12 AUs is annotated for all video frames. The dataset contains approximately 140,000 AU-labeled images. To evaluate model performance, we conduct 3-fold cross-validation on the dataset.

DISFA [22] dataset comprises 26 participants, with AUs annotated on an intensity scale from 0 to 5. In total, it contains approximately 130,000 AU-labeled frames. To evaluate model performance, we perform 3-fold cross-validation, splitting the dataset into three folds based on subject IDs.

UNBC-McMaster [23] dataset comprises 200 brief videos (usually <10 sec) and about 48,000 frames across 129 participants. Each video (image sequence) is annotated with pain intensity scores from 0 to 4. A 10-fold cross-validation is conducted on the dataset.

F Evaluation of Pre-trained Representations

To evaluate the discriminative and task-adaptive quality of TAME's representations, we freeze the pre-trained encoder and attach a lightweight task-specific head (e.g., a multi-label classifier for AU detection, a regression head for AU intensity, or a classifier for pain estimation). Only the head is trained, while the encoder remains fixed. This setup avoids end-to-end fine-tuning and provides an objective measure of feature quality across different pre-training methods. The results are shown in Tab. 1.

a) Comparison with Self-Supervised Pre-training Methods

Under the frozen-backbone evaluation, TAME consistently outperforms self-supervised baselines across all three FAA tasks, indicating that its pre-trained features are both discriminative and transferable. Unlike standard approaches such as MAE [4] and MoCo v3 [25], TAME introduces task-aware guidance into feature learning process.

Table 1: Evaluation of pre-trained representations on facial action analysis tasks

Categories	Methods	Backbones	Pre-training Datasets	AU Detection F1 Score ↑	AU Intensity Est. ICC ↑	Pain Est. Acc. ↑
SSL Methods	MAE [4]	ViT-B[24]	AffectNet[20]	0.60	0.58	0.86
	MoCo v3 [25]	ViT-B[24]	AffectNet[20]	0.54	0.45	0.85
Foundation Models	CLIP [26]	ViT-B[24]	LAION-2B[27]	0.53	0.38	0.84
	CLIP [26]	ViT-B[24]	AffectNet[20]	0.57	0.49	0.85
Ours	TAME	ViT-B[24]	AffectNet[20]	**0.67**	**0.65**	**0.91**

Conventional self-supervised methods learn general-purpose features but lack alignment with task requirements. As a result, they fail to focus on task-relevant regions or semantics, which limits their effectiveness in downstream tasks that demand fine-grained and localized feature representations.

TAME addresses this issue by introducing a task-aware feedback mechanism during pre-training. The validator provides task-matching signals that guide the adjuster to reweight the reconstruction loss spatially, encouraging the encoder to attend to regions that are more informative for downstream tasks. Through meta-optimization, this process continuously aligns feature learning with task demands, enabling the encoder to produce highly discriminative representations even without fine-tuning.

b) Comparison with Vision Foundation Models

The comparison with vision foundation models further highlights the task-adaptivity of TAME at the representation level. With the backbone frozen and only the task head trained, CLIP [26] consistently underperforms TAME across all three facial action analysis tasks. This indicates that, despite CLIP's strong open-domain generalization enabled by large-scale image–text pre-training (e.g., LAION-2B [27]), its representations are not well-suited for the fine-grained structural discrimination required in FAA.

In contrast, TAME's region-weighted pre-training directs attention toward task-relevant facial areas, enhancing local discriminability while preserving general feature quality. As a result, TAME provides more effective task-adapted representations and achieves superior downstream performance.

G Comparison with Semi-supervised Methods

Tab. 2 shows that TAME outperform conventional semi-supervised methods on three FAA tasks. In semi-supervised approaches, the base model is updated using both labeled and unlabeled data simultaneously, which often leads to quick convergence on the small labeled set. This premature convergence restricts the model's learned representation space, diminishing the potential advantages of large-scale unlabeled data. TAME mitigates this overfitting issue by first training the base model solely on unlabeled data, allowing it to learn robust and generalizable features. The limited labeled data is then introduced separately for task-aware refinement through the adjuster and validator, preventing the small labeled set from prematurely constraining the representation space.

Table 2: Comparison with semi-supervised methods on facial action analysis tasks

Methods	Backbones	Unlabeled Datasets	AU Detection F1 score ↑	AU Intensity Est. ICC ↑	Pain Est. Acc. ↑
Sup-MAE [28]	ViT-B[24]	AffectNet	0.58	0.40	0.85
FixMatch [29]	ViT-B[24]	AffectNet	0.54	0.31	0.85
Meta Pseudo Label [30]	ViT-B[24]	AffectNet	0.56	0.35	0.88
TAME	ViT-B[24]	AffectNet	**0.69**	**0.68**	**0.92**

H Impact of Model and Data Scaling on Downstream Performance

This section evaluates how TAME's performance scales with respect to base model size, adjuster size, and pre-training data volume, aiming to assess its scalability and practical applicability.

a) Impact of Base Model Scaling

As shown in Table 3, increasing the encoder size from ViT-B to ViT-L improves performance on AU detection and pain estimation, while the improvement on AU intensity estimation is marginal. This indicates that a larger backbone can benefit tasks that require stronger global representations, but may offer limited advantages for tasks focused on subtle local variations.

Facial action analysis often relies on fine-grained, localized features rather than broad contextual understanding. As a result, simply increasing model size does not guarantee consistent performance gains. Importantly, TAME delivers strong performance with the mid-sized ViT-B, highlighting its efficiency and suitability for deployment in resource-constrained real-world settings.

Table 3: Performance impact of scaling the base model capacity

Foundation Models	AU Detection F1 Score ↑	AU Intensity Est. ICC ↑	Pain Est. Acc. ↑
ViT-B[24]	0.69	0.68	0.92
ViT-L[24]	**0.70**	**0.68**	**0.93**

b) Impact of Adjuster Capacity

Table 4 presents TAME's performance when the adjuster is scaled from ViT-S to ViT-B. The results show negligible performance differences across all three downstream tasks, indicating that a small-size adjuster is already sufficient for providing effective guidance.

This result aligns with the adjuster's intended role in TAME. As an auxiliary module, the adjuster generates region-weighting signals based on task feedback, rather than directly learning visual representations. Since it is not responsible for feature extraction, its effectiveness depends less on model capacity. Therefore, a lightweight adjuster is sufficient to perform its function, while also reducing computational cost.

Table 4: Performance impact of scaling the adjuster capacity

Adjuster Structure	AU Detection F1 Score ↑	AU Intensity Est. ICC ↑	Pain Est. Acc. ↑
ViT-S[24]	0.69	0.68	0.92
ViT-B[24]	**0.69**	**0.68**	**0.92**

c) Impact of Pre-training Data Volume

To examine the effect of pre-training data volume, we double the amount of pre-training data from 300,000 to 600,000 images. As shown in Table 5, this expansion leads to consistent improvements across all three tasks: AU detection F1 score increases from 0.69 to 0.71, AU intensity estimation ICC improves from 0.68 to 0.69, and pain estimation accuracy rises from 0.92 to 0.95.

These results demonstrate that enlarging the pre-training dataset enhances the expressiveness and transferability of learned representations, which is particularly beneficial for fine-grained facial action analysis.

Table 5: Performance impact of pre-training data volume

Number of Unlabeled Images	AU Detection F1 Score ↑	AU Intensity Est. ICC ↑	Pain Est. Acc. ↑
300,000	0.69	0.68	0.92
600,000	**0.71**	**0.69**	**0.95**

I Impact of Downstream Labeled Data Proportion in Pre-training

Table 6 presents the average F1 score on the AU detection task when TAME is pre-trained using 0%, 20%, 40%, 60%, 80%, and 100% of the downstream labels. Here, "proportion" refers to the percentage of labeled samples provided to the adjuster and validator during pre-training, relative to the total amount of downstream supervision.

With no supervision (0%), TAME achieves an F1 score of 0.631, identical to that of the MAE [4] baseline. As the label proportion increases from 20% to 80%, performance steadily improves from 0.676 to 0.695, indicating that even limited supervision enables the adjuster to guide the encoder toward task-relevant regions. Notably, with just 20% of the downstream labels, TAME already surpasses its unsupervised counterpart (0.676 vs. 0.631), demonstrating its ability to efficiently leverage limited task supervision. With full supervision (100%), the F1 score further rises to 0.700, reflecting the model's full potential under complete task guidance.

Table 6: TAME performance with varying proportions of labeled data in pre-training

Proportion	0%	20%	40%	60%	80%	100%
F1 Score Avg.	0.631	0.676	0.685	0.691	0.695	0.700
Δ	-	+7.13%	+1.33%	+0.88%	+0.58%	+0.72%

J Computational Resource Consumption

Table 7 compares the computational resource usage and downstream performance of standard MAE [4] and TAME. TAME achieves F1 scores of 0.69 (AU detection), 0.68 (AU intensity estimation), and 0.92 (pain estimation), consistently outperforming MAE's corresponding scores of 0.63, 0.60, and 0.89. This performance gain comes at a modest computational cost: TAME's pre-training takes approximately 40 hours, compared to 32 hours for MAE, and peak GPU memory usage increases from 18 GB to 32 GB, primarily due to the dual-loop optimization and additional components (adjuster and validator) introduced for meta-learning.

To mitigate the added computational overhead, TAME adopts a Neumann-series approximation based on the Implicit Function Theorem (Eq. 2), which reduces the complexity of second-order gradient computation from $O(n^3)$ to $O(n^2)$. In addition, a progressive update strategy—such as performing a full meta-update once every four standard SSL iterations—can further reduce training time and bring its efficiency closer to that of MAE [4].

Table 7: Comparison of computational resource consumption across methods

Methods	Number of Pre-training Data	Batch Size	Training Epochs	Pre-training Time (Hours)	Single-GPU Memory Usage(GB)	AU Detection F1 Score ↑	AU Intensity Est. ICC ↑	Pain Est. Acc. ↑
MAE [4]	300,000	32	200	$\tilde{3}2$	$\tilde{1}8$	0.63	0.60	0.89
TAME	300,000	32	200	$\tilde{4}0$	$\tilde{3}2$	**0.69**	**0.68**	**0.92**

Although the memory requirement rises to around 32 GB per GPU (at batch size 32), this remains feasible on modern hardware such as the NVIDIA A100. Further memory optimization is possible

through techniques like mixed-precision training (e.g., FP16), which can effectively reduce peak GPU memory consumption without sacrificing performance.

K Visualization

We utilize Attention Rollout [31] to visualize attention maps of pre-trained models, including MAE [4], TAME pre-trained for AU detection, and TAME pre-trained for pain estimation. Results are presented in Fig. 1.

For MAE [4], the attention is mainly focused on the central region of the face. This suggests that MAE [4] learns a global representation of the facial image, making it effective at capturing general features but potentially less precise when analyzing subtle details.

In contrast, TAME for AU detection exhibits a more structured and localized attention pattern, with emphasis on specific facial regions associated with Action Units (AUs). Action Units refer to the smallest visible facial movements linked to expressions. The pre-trained model appears to have learned to focus on key areas such as the eyes, eyebrows, and mouth. This concentrated attention suggests that TAME (AU) is more specialized in capturing the subtle muscle movements that contribute to different expressions.

For pain estimation, the attention of TAME is mainly focused on the regions typically associated with pain expression, such as the nose and cheek. This aligns with research in pain expression analysis, which highlights that facial features such as tightened eyelids, wrinkled nose, and contracted cheeks are critical indicators of discomfort. By prioritizing these regions, the model enhances its sensitivity to pain-related expressions, allowing for more accurate and reliable pain recognition.

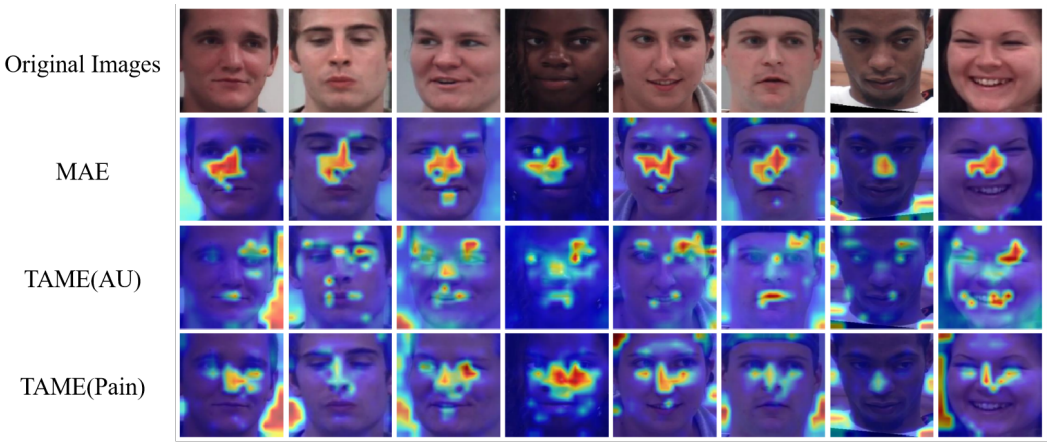

Figure 1: Comparison of attention maps of pretrained MAE and TAME models pretrained for different facial action analysis tasks.

References

[1] 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.

[2] 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.

[3] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

[4] 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.

[5] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. *arXiv preprint arXiv:2212.07143*, 2022.

[6] Yanan Chang and Shangfei Wang. Knowledge-driven self-supervised representation learning for facial action unit recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20417–20426, 2022.

[7] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3024–3033, 2021.

[8] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

[9] Jürgen Schmidhuber. A neural network that embeds its own meta-levels. In *IEEE International Conference on Neural Networks*, pages 407–412. IEEE, 1993.

[10] Sebastian Thrun. Lifelong learning algorithms. *Learning to learn*, 8:181–209, 1998.

[11] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[12] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[13] Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-learning with implicit gradients. *Advances in neural information processing systems*, 32, 2019.

[14] Yang Fan, Fei Tian, Tao Qin, Xiang-Yang Li, and Tie-Yan Liu. Learning to teach. *arXiv preprint arXiv:1805.03643*, 2018.

[15] Yang Fan, Yingce Xia, Lijun Wu, Shufang Xie, Weiqing Liu, Jiang Bian, Tao Qin, and Xiang-Yang Li. Learning to reweight with deep interactions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7385–7393, 2021.

[16] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International conference on machine learning*, pages 2304–2313. PMLR, 2018.

[17] Ryuichiro Hataya, Jan Zdenek, Kazuki Yoshizoe, and Hideki Nakayama. Meta approach to data augmentation optimization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2574–2583, 2022.

[18] Shikun Liu, Andrew Davison, and Edward Johns. Self-supervised generalisation with meta auxiliary learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[19] Jonathan Lorraine, Paul Vicol, and David Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In *International Conference on Artificial Intelligence and Statistics*, pages 1540–1552. PMLR, 2020.

[20] Ali Mollahosseini, Behzad Hasani, and Mohammad H Mahoor. Affectnet: A database for facial expression, valence, and arousal computing in the wild. *IEEE Transactions on Affective Computing*, 10(1):18–31, 2017.

[21] Xing Zhang, Lijun Yin, Jeffrey F Cohn, Shaun Canavan, Michael Reale, Andy Horowitz, and Peng Liu. A high-resolution spontaneous 3d dynamic facial expression database. In *2013 10th IEEE international conference and workshops on automatic face and gesture recognition*, pages 1–6. IEEE, 2013.

[22] S Mohammad Mavadati, Mohammad H Mahoor, Kevin Bartlett, Philip Trinh, and Jeffrey F Cohn. Disfa: A spontaneous facial action intensity database. *IEEE Transactions on Affective Computing*, 4(2):151–160, 2013.

[23] Patrick Lucey, Jeffrey F Cohn, Kenneth M Prkachin, Patricia E Solomon, and Iain Matthews. Painful data: The unbc-mcmaster shoulder pain expression archive database. In *2011 IEEE International Conference on Automatic Face & Gesture Recognition (FG)*, pages 57–64. IEEE, 2011.

[24] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[25] 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.

[26] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2829, 2023.

[27] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022.

[28] Feng Liang, Yangguang Li, and Diana Marculescu. Supmae: Supervised masked autoencoders are efficient vision learners. *arXiv preprint arXiv:2205.14540*, 2022.

[29] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020.

[30] Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V Le. Meta pseudo labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11557–11568, 2021.

[31] Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. *arXiv preprint arXiv:2005.00928*, 2020.