More Reliable Pseudo-labels, Better Performance: A Generalized Approach to Single Positive Multi-label Learning

Supplementary Material

1. Characteristics of the Weight Functions

The parameters $\alpha = [\sigma, \mu]$ of $v^o(p;\alpha)$, for $o = 1, 2, 3, 4$, are linearly updated over the training epochs, following [2], as shown below:

$$\mu^{(t)} = \mu^{(0)} + \left(\mu^{(T)} - \mu^{(0)}\right) \cdot \frac{t}{T},$$
$$\sigma^{(t)} = \sigma^{(0)} + \left(\sigma^{(T)} - \sigma^{(0)}\right) \cdot \frac{t}{T},$$

where $t = 1, 2, \ldots, T$ denotes the current training epoch, T is the total number of training epochs, $\mu^{(0)}$ and $\sigma^{(0)}$ are initial values, $\mu^{(0)}$, $\sigma^{(0)}$, $\mu^{(T)}$, $\sigma^{(T)}$ are hyperparameters.

Recall that $v^1(p;\alpha) = 1$ is used to handle given single-positive labels. In contrast, $v^2(p;\alpha) = \exp\left(-\frac{(p-\mu)^2}{2\sigma^2}\right)$ is used for undefined labels, i.e., labels that are neither observed in the dataset nor recognized by the pseudo-labeling method $\mathcal{M}$.

The formulation of $v^2(p;\alpha)$ depends on the output confidence p. When an undefined label has a p value close to 1, the instance is likely a false negative or an outlier; thus, its weight should be reduced. Similarly, in cases of imbalance between positive and negative samples, the weight of easy samples should be reduced appropriately. For undefined labels with p close to 0, the instance is likely easy, so its weight should be small. Conversely, if p is near μ, the weight should be higher due to the potential presence of semi-hard examples.

As negative pseudo-labels are derived from undefined labels, we use the same weight function, $v^3(p;\alpha) = v^2(p;\alpha)$. For positive pseudo-labels, we use the weight function

$$v^4(p;\alpha) = \min\left(\max\left(1 - v^3(p;\alpha), \lambda_1\right), \lambda_2\right),$$

which operates in reverse of $v^3(p;\alpha)$.

2. Proof of theorem 4.1

To prove Theorem 4.1, we demonstrate that under the conditions $\max(C(\mathcal{M}), |m' - \hat{m}|) \to 0$, the proposed GPR Loss reduces to the GR Loss.

Definitions Recap:

- $C(\mathcal{M}) = \exp\left(\sum_{n=1}^{N}\sum_{i=1}^{C}\mathbb{I}_{[y_{n,i}=1]}\log P(l_{n,i} = 1|x_n, \mathcal{M})\right)$.
- $m' = \frac{1}{N'}\sum_{n=1}^{N'}\sum_{i=1}^{C}\mathbb{I}_{[y_{n,i}=1]}$ (validation set average positives).
- $\hat{m} = \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{C}p_{n,i}$ (predicted average positives).

Key Observations:

1. Regularization Term R:

$$R = \left(\frac{\hat{m} - m}{C}\right)^2.$$

As $|m' - \hat{m}| \to 0$ and $m' \to E_{pos}$ (Remark 2), $\hat{m} \to E_{pos}$. Hence, $R \to 0$.

2. Pseudo-Label Confidence $C(\mathcal{M})$: $C(\mathcal{M}) \to 0$ implies the pseudo-labeling method $\mathcal{M}$ is unconfident. This occurs when $P(l_{n,i} = 1|x_n, \mathcal{M}) \to 0$ for true positives ($y_{n,i} = 1$), forcing $l_{n,i} = 0$ for all n, i (undefined labels).

Under $C(\mathcal{M}) \to 0$ and $l_{n,i} = 0$:

- Loss Terms $\mathcal{L}_{n,i}^{new}$: Since $l_{n,i} = 0$, the terms $\mathcal{L}_{n,i}^3$ (negative pseudo-labels) and $\mathcal{L}_{n,i}^4$ (positive pseudo-labels) are inactive. Only $\mathcal{L}_{n,i}^1$ (observed positives) and $\mathcal{L}_{n,i}^2$ (Assume Negative) remain, matching the GR Loss terms $\mathcal{L}_{n,i}^{old}$.
- Weight Terms $v^{\text{new}}(p_{n,i};\alpha)$: Similarly, v^{new} collapses to v^{old}, as only the original Assume Negative labels ($\hat{y}_{n,i}$) contribute.

With $R \to 0$ and $C(\mathcal{M}) \to 0$, the GPR Loss in Eq. (4) simplifies to the GR Loss in Eq. (3) as follow:

$$\mathcal{L}^{\text{GPR}} \to \frac{1}{NC}\sum_{n=1}^{N}\sum_{i=1}^{C}v^{\text{old}}(p_{n,i};\alpha)\cdot\mathcal{L}_{n,i}^{\text{old}} = \mathcal{L}^{\text{GR}}.$$

3. False Negative Pseudo-label Estimator

The formulation of $\hat{k}(p;\beta)$ relies on two assumptions: initially, $\hat{k}(p;\beta)$ behaves almost like a constant function, but it gradually transforms into a monotonically increasing function towards the final training stage [2]. This function is modeled using a logistic function as follows:

$$\hat{k}(p;\beta) = \frac{1}{1 + \exp\{-(w\cdot p + b)\}},$$

where $\beta = [w, b]$. To ensure $\hat{k}(p;\beta)$ meets the required properties, both w and b are assumed to increase linearly with training epochs, expressed as:

$$w^{(t)} = w^{(0)} + \left(w^{(T)} - w^{(0)}\right)\cdot\frac{t}{T},$$
$$b^{(t)} = b^{(0)} + \left(b^{(T)} - b^{(0)}\right)\cdot\frac{t}{T},$$

where $w^{(0)}$ and $b^{(0)}$ are initial values, $w^{(0)}$, $b^{(0)}$, $w^{(T)}$, $b^{(T)}$ are hyperparameters.

4. Graph Convolutional Network

In this section, we introduce the process to obtain the adjacent matrix A^*, following [5].

From a set of C classes, label features are derived by feeding a prompt template, *"A photo of a* $\{class\}$*"*, into the CLIP text encoder, resulting in label features $\bar{z}_i$ for each class i-th. The correlation prior among labels, $A = (a_{ij})_{C \times C}$, is computed as:

$$a_{ij} = \mathrm{sim}(\bar{z}_i, \bar{z}_j)$$

where $\mathrm{sim}(\cdot, \cdot)$ is the cosine similarity. For each row a_i, the top u elements are selected and the rest are set to zero, resulting in a sparse matrix, $A' = (a'_{ij})_{C \times C}$:

$$a'_{ij} = \begin{cases} a_{ij}, & j \in \mathrm{TopK}(a_i, u), \\ 0, & j \notin \mathrm{TopK}(a_i, u). \end{cases}$$

We mitigate over-smoothness of graph representation by adjusting the sparse graph A' as follows:

$$\bar{a}_{ij} = \begin{cases} \left(\frac{e}{\sum_{i \neq j} a'_{ij'}} \right) \times a'_{ij}, & \text{if } i \neq j \\ 1 - e, & \text{if } i = j \end{cases}$$

where e is a hyper-parameter that determines weights assigned to a node itself and its neighboring nodes. The label correspondence graph G is derived as:

$$a^*_{ij} = \frac{\mathbb{1}_{[}\bar{a}_{ij} \neq 0] \exp(\bar{a}_{ij} / \tau')}{\sum_j \mathbb{1}_{[}\bar{a}_{ij} \neq 0] \exp(\bar{a}_{ij} / \tau')}$$

where τ' controls distribution smoothness. The adjacency matrix of G is denoted as $A^* = (a^*_{ij})_{C \times C}$.

The importance of each node is emphasized in A^*, with weights assigned to other nodes based on their relationships, thus encoding label correspondence in the structured graph A^*.

For a GCN with L layers, the weight matrix W_l, with a shape of $d_{\text{in}} \times d_{\text{out}}$, is typically initialized using a uniform distribution with bounds determined by the size of the output features:

$$W_l \sim \text{Uniform}\left(-\frac{1}{\sqrt{d_{\text{out}}}}, \frac{1}{\sqrt{d_{\text{out}}}} \right).$$

This GCN will be used to add controlled noise based on label-to-label correspondence into the CLIP text encoder of DAMP. We conjecture that this can help DAMP remove pseudo-labels with low to medium confidence, retaining only the pseudo-labels with high confidence, without requiring additional training efforts.

5. Datasets

In this study, the proposed method is evaluated using environmental experiments similar to those described in [2, 4, 8, 9] across four standard benchmark datasets: PASCAL VOC 2012 (VOC), MS-COCO 2014 (COCO), NUS-WIDE (NUS), and CUB-200-2011 (CUB). The aim is to assess the effectiveness of various Single-Positive Multi-Label Learning (SPML) methods. To do this, fully labeled multi-label image datasets are initially used and then systematically reduced by discarding some annotations.

To simulate single-positive training data, one positive label is randomly chosen to be retained for each training example. This simulation is conducted once per dataset, ensuring that the same label set is consistently used for all comparisons within that dataset. Thus, every image in a batch retains the same single positive label throughout the process. Twenty percent of the training set for each dataset is set aside for validation purposes. Both the validation and test sets are fully labeled. VOC12 consists of 5,717 training images and 20 classes, with results reported on the official validation set containing 5,823 images. COCO includes 82,081 training images and 80 classes, with results also reported on the official validation set of 40,137 images.

The complete NUS dataset is not available online, necessitating a re-scraping from Flickr. Consequently, not all original images were obtained. A total of 126,034 training images and 84,226 test images were collected, covering 81 classes according to [4]. The training and test sets were combined, with 150,000 images randomly selected for training and the remaining 60,260 images used for testing. The CUB dataset comprises 5,994 training images and 5,794 test images. Each CUB image is annotated with a vector indicating the presence or absence of 312 binary attributes. Although subsets of these attributes are known to be mutually exclusive, this information is not utilized in this study. Additional statistics on the datasets are provided in Table 1.

Table 1. Data statistics on four benchmark datasets.

Statistics		VOC	COCO	NUS	CUB
# Classes		20	80	81	312
# Images	Training	4,574	65,665	120,000	4,795
	Validation	1,143	16,416	30,000	1,199
	Test	5,823	40,137	60,260	5,794
# Labels per training image	Positive	1.5	2.9	1.9	31.4
	Negative	18.5	77.1	79.1	280.6

6. DAMP

In this section, we introduce a Dynamic Augmented Multi-focus Pseudo-labeling (DAMP) approach for SPML.

6.1. CLIP Inference

As introduced in [12], given an image input x and the i-th class from a set of C classes, the corresponding visual embedding and textual embedding are $h = E_v(x) \in \mathbb{R}^K$ and $t_i = E_t(\mathcal{P}_i) \in \mathbb{R}^K$, respectively. Here, E_v and E_t are the image and text encoders of CLIP model with dimension K, and $\mathcal{P}_i$ is a predefined prompt for class i, such as *"a photo of a* $\{$*class*$\}$*"*. The cosine similarity score $\hat{s}_i$ between the i-th class and image x is computed as follows:

$$\hat{s}_i = \frac{h^\top t_i}{\|h\| \cdot \|t_i\|} \tag{1}$$

where $\|\cdot\|$ denotes the Euclidean norm. These scores are normalized with a temperature parameter τ as follows:

$$s_i = \frac{\exp\left(\hat{s}_i/\tau\right)}{\sum_{i=1}^{C} \exp\left(\hat{s}_i/\tau\right)} \tag{2}$$

We denote $S = \{s_1, s_2, \ldots, s_C\}$ as the probability distribution for the input x across C classes.

6.2. Strengthening CLIP Inference with Noise

Several works, including [7, 10, 13], have studied enhancing model performance during fine-tuning by adding controlled noise to model embeddings. Additionally, in [3, 5], label-to-label relationships are presented by GCN. Inspired by this, we propose adding controlled label-to-label correspondence noise to the text embeddings of the CLIP model, defined as $t_i = G(E_t(\mathcal{P}_i)) + E_t(\mathcal{P}_i))$, where $G(\cdot)$ is a GCN with L layers, updated as follows: $H_{l+1} = \text{LeakyReLU}(A^* H_l W_l)$, for $l \in [0, L)$ and $H_0 = \{E_t(\mathcal{P}_i) \mid 1 \leq i \leq C\}$. The graph G remains frozen during training, and weights W_l are initialized from a uniform distribution. The adjacency matrix A^* is derived from cosine similarity scores between the text embeddings of the classes produced by the CLIP text encoder. Further implementation details are in the supplementary materials.

6.3. Dynamic Augmented Multi-focus Pseudo-labeling

Augmentor. Let I be a given image of size $H \times W$. We first divide the image I into smaller patches $\{P_z\}_{z=1,2,\ldots,R}$ using a $g \times g$ grid, where g is the grid size and $R = g^2$ is the total number of patches. Each patch has nominal dimensions of $\frac{H}{g} \times \frac{W}{g}$, and its size is increased by a random ratio r, creating an overlap between adjacent patches. We then process the image and its patches using a transformation pipeline $\mathrm{T}(\cdot)$, which includes standard preprocessing and weakly data augmentation techniques to generate various views for CLIP as follows: $x^{global} = \mathrm{T}(I)$, $x_z^{local} = \mathrm{T}(P_z)$.

Global and local views. Following Secs. 6.1 and 6.2, from x^{global} and x_z^{local}, we obtain the probability distributions $S^{global} = \{s_1^{global}, s_2^{global}, \ldots, s_C^{global}\}$ and $S_z^{local} = \{s_{z,1}^{local}, s_{z,2}^{local}, \ldots, s_{z,C}^{local}\}$, respectively. For simplicity, we use the same temperature parameter τ for both the image and its patches.

Local threshold based on single positives. Let $\hat{c}$ be the given single positive label, according to the SPML setting in **??**, for the image I. The local threshold ζ^{local}, which defines the patches to be trusted, is adjusted based on $\zeta^{local} = \min(s_{\hat{c}}^{global}, \nu)$, where ν is the general local threshold, set as a hyperparameter. In some cases, if ν is set too high to recognize hard positives, we should consider the scores above $s_{\hat{c}}^{global}$, as these can be meaningful, since $\hat{c}$ is one of the true labels of the global view.

Aggregator. From the distributions $\{S_z^{\text{local}}\}_{z=1,2,\ldots,R}$, we aggregate a unified local distribution S^{agg} following [1]. For each class c, we compute $\omega_c = \max_{z=1,\ldots,R} s_{z,c}^{\text{local}}$ and $\psi_c = \min_{z=1,\ldots,R} s_{z,c}^{\text{local}}$, defining the aggregation score s_c^{agg} for class c as

$$s_c^{\text{agg}} = \mathbb{1}_{[\omega_c \geq \zeta^{\text{local}}]}\omega_c + \mathbb{1}_{[\omega_c < \zeta^{\text{local}}]}\psi_c,$$

where $\mathbb{1}_{[\cdot]}$ is the indicator function. This yields $S^{\text{agg}} = \{s_1^{\text{agg}}, s_2^{\text{agg}}, \ldots, s_C^{\text{agg}}\}$, the soft aggregation vector for each input image.

Positive pseudo-labels. To extract reliable positive pseudo-labels, we integrate both global and aggregated local similarities into $S^{final} = \frac{1}{2}\left(S^{global} + S^{agg}\right)$. Let $Q' = \{l'_1, l'_2, \cdots, l'_C\}$ be the pseudo labels of the image I. We convert the soft similarity scores S^{final} into hard pseudo-labels as follows:

$$l'_c = \begin{cases} 1, & s_c^{final} \in \text{TopK}(S^{final}, k) \;\;\&\;\; s_c^{final} \geq \zeta^{global} \\ 0, & \text{otherwise,} \end{cases}$$

where ζ^{global} is the global threshold for high-confidence positive pseudo-labels, set as a hyperparameter, and k limits the number of positive pseudo-labels.

Negative pseudo-labels. To identify potential negative pseudo-labels we compute average similarity scores as:

$$S^{avg} = \frac{1}{2}\left(S^{global} + \frac{1}{R}\sum_{z=1}^{R} S_z^{local}\right).$$

We use S^{avg} to refine Q' by assigning negative pseudo-labels, producing the final pseudo-labels $Q = \{l_1, l_2, \cdots, l_C\}$. A class c is designated as a negative pseudo-label if its score s_c^{avg} falls within the lowest $\Delta_{\text{neg}}\%$

of values in S^{avg}. Assuming a potential negative pseudo-label has low scores in both image I and every patch P_z according to the VLM, we define the assignment as:

$$l_c = \begin{cases} -1, & s_c^{\text{avg}} \leq \theta_{\Delta_{\text{neg}}}(S^{\text{avg}}) \\ l'_c, & \text{otherwise} \end{cases}$$

where $\theta_{\Delta_{\text{neg}}}(S^{\text{avg}})$ denotes the Δ_{neg}-th-percentile of S^{avg}, serving as the threshold to identify the lowest $\Delta_{\text{neg}}\%$ of values in S^{avg} as negative pseudo-labels.

6.4. Performance

As shown in Tab. 2, the difference in precision on positive pseudo-labels generated by DAMP across epochs is generally small, demonstrating the stability of the training process when applying the randomization-based pseudo-labeling approach in our proposed method.

Table 2. DAMP performance reported in average precision across epochs.

Metrics	VOC	COCO	NUS	CUB
Average Precision	65.13 ± 0.53	84.79 ± 0.05	37.09 ± 0.08	19.07 ± 0.02

7. Hyperparameters Settings

The hyperparameters, including the temperature τ that controls the softmax prediction scores, the range of weights for the positive pseudo-label loss (λ_1, λ_2), the top k highest scores used to extract positive pseudo-labels, the global threshold ζ^{global}, the general local threshold ν, and the loss coefficients, are described in Table 3.

Table 3. The hyperparameters of AEVLP conducted on backbone ResNet-50 [6]

Hyperparameters	VOC	COCO	NUS	CUB
$(w^{(0)}, b^{(0)})$	(0, -2)	(0, -2)	(0, -2)	(0, -4)
$(w^{(T)}, b^(T))$	(2, -2)	(10, -8)	(10, -8)	(10, -8)
$(\mu^{(0)}, \sigma^{(0)})$	(0.5, 2)	(0.5, 2)	(0.5, 2)	(0.5, 2)
$(\mu^{(T)}, \sigma^{(T)})$	(0.8, 0.5)	(0.8, 0.5)	(0.8, 0.5)	(0.8, 0.5)
ν	0.1	0.3	0.3	0.005
k	3	3	3	32
ζ^{global}	0.5	0.5	0.6	0.005
τ	0.01	0.01	0.01	0.1
q_1	1	1	1	1
q_2	0.01	0.01	0.01	0.01
q_3	0.9	0.9	0.9	0.9
(λ_1, λ_2)	(0.4, 0.7)	(0.5, 0.7)	(0.5, 0.7)	(0.2, 0.8)
η	0.05	0.001	0.001	0.01

8. Visualization

The dynamic patching mechanism of the Augmentor module, which helps the CLIP model attend to more details in the input images, is visualized in Fig. 1. We also provide our output predictions on the COCO test dataset to demonstrate the effectiveness of our method in Fig. 2.

References

[1] Rabab Abdelfattah, Qing Guo, Xiaoguang Li, Xiaofeng Wang, and Song Wang. Cdul: Clip-driven unsupervised learning for multi-label image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1348–1357, 2023. 3

[2] Yanxi Chen, Chunxiao Li, Xinyang Dai, Jinhuan Li, Weiyu Sun, Yiming Wang, Renyuan Zhang, Tinghe Zhang, and Bo Wang. Boosting single positive multi-label classification with generalized robust loss. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 3825–3833. International Joint Conferences on Artificial Intelligence Organization, 2024. Main Track. 1, 2

[3] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. Multi-label image recognition with graph convolutional networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5177–5186, 2019. 3

[4] Elijah Cole, Oisin Mac Aodha, Titouan Lorieul, Pietro Perona, Dan Morris, and Nebojsa Jojic. Multi-label learning from single positive labels. In *CVPR-21*, 2021. 2

[5] Zixuan Ding, Ao Wang, Hui Chen, Qiang Zhang, Pengzhang Liu, Yongjun Bao, Weipeng Yan, and Jungong Han. Exploring structured semantic prior for multi label recognition with incomplete labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3398–3407, 2023. 2, 3

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4, 6

[7] Neel Jain, Ping-yeh Chiang, Yuxin Wen, John Kirchenbauer, Hong-Min Chu, Gowthami Somepalli, Brian R. Bartoldson, Bhavya Kailkhura, Avi Schwarzschild, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Neftune: Noisy embeddings improve instruction finetuning. *CoRR*, abs/2310.05914, 2023. 3

[8] Youngwook Kim, Jae Myung Kim, Zeynep Akata, and Jungwoo Lee. Large loss matters in weakly supervised multi-label classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14156–14165, 2022. 2

[9] Youngwook Kim, Jae Myung Kim, Jieun Jeong, Cordelia Schmid, Zeynep Akata, and Jungwoo Lee. Bridging the gap between model explanations in partially annotated multi-label classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3408–3417, 2023. 2

[10] Kezhi Kong, Guohao Li, Mucong Ding, Zuxuan Wu, Chen Zhu, Bernard Ghanem, Gavin Taylor, and Tom Goldstein. Robust optimization as data augmentation for large-scale

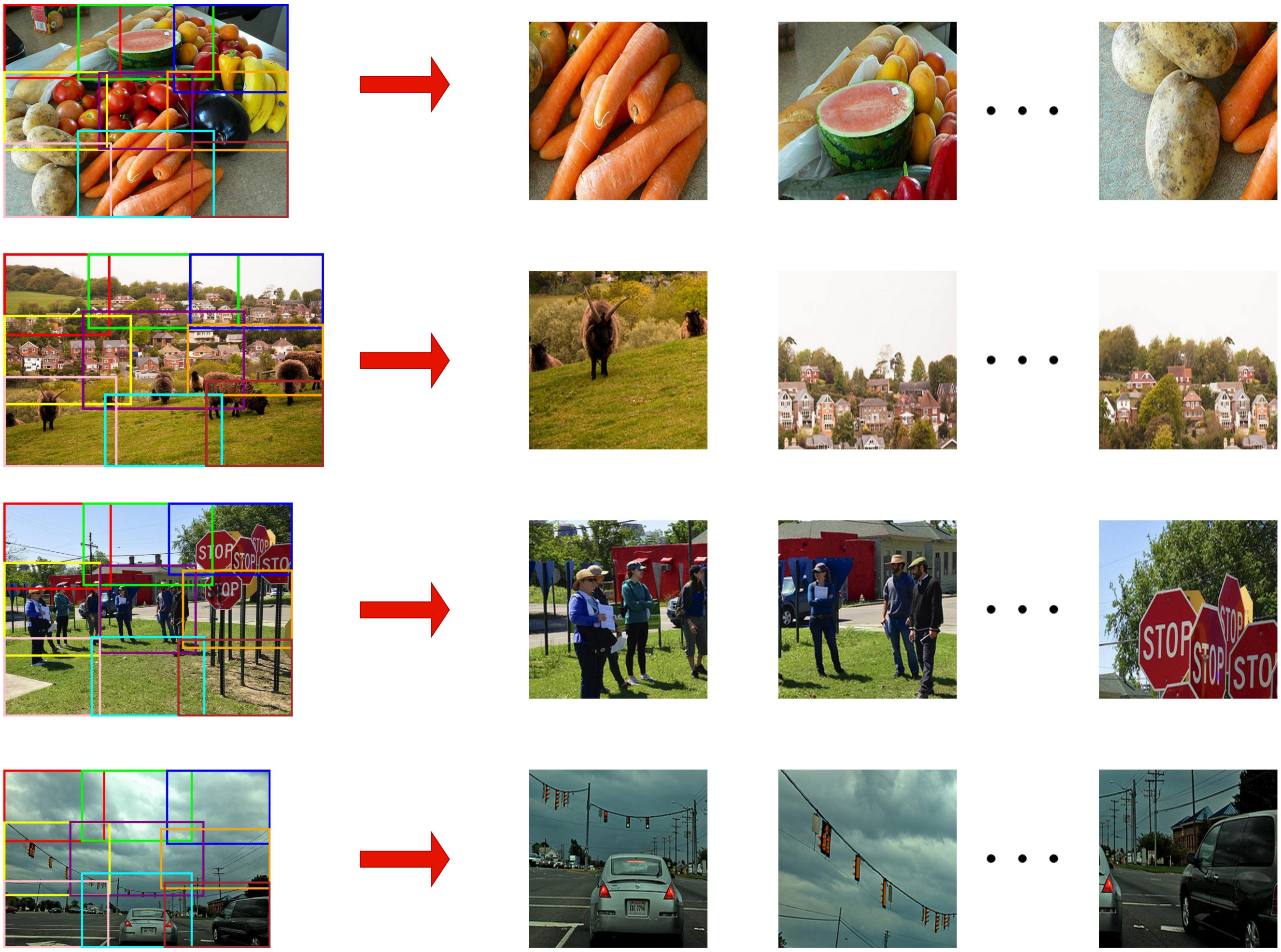

Figure 1. Visualization of patching mechanism in Augmentor.

graphs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 60–69, 2022. 3

[11] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 6

[12] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3

[13] Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. Freelb: Enhanced adversarial training for natural language understanding. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. 3

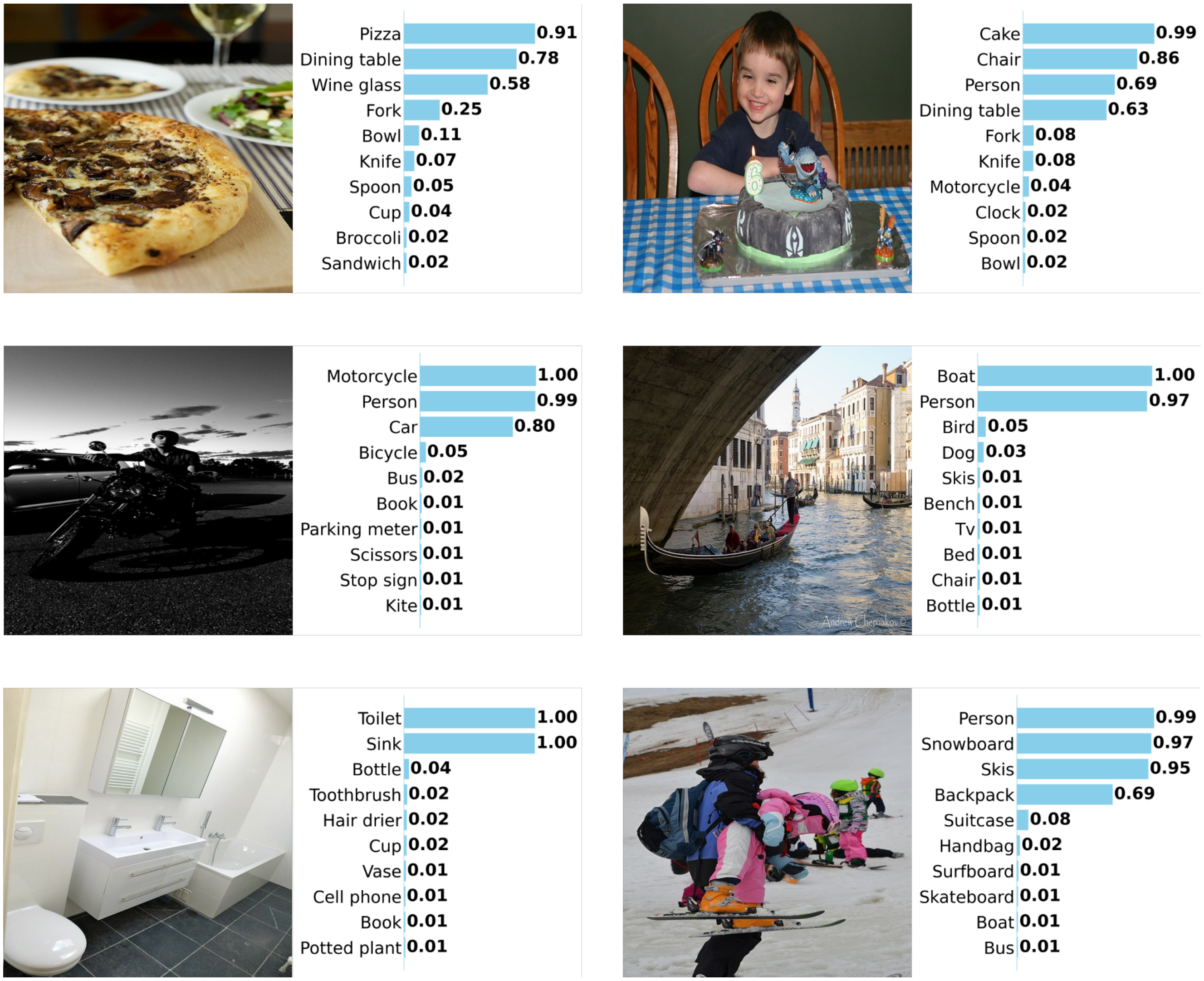

Figure 2. Inference of ResNet-50 [6] model trained with our AEVLP method on images from COCO [11] test dataset.