Appendix

1. Logits Normalization

In this section, we first analyze the effect of temperature hyperparameter in the naive logits distillation loss and give the mathematical demonstration of Theorem 1.

1.1. Mathematical Derivation

Suppose that pre-softmax logits produced by teacher and student networks are $v_i^+ = C_t(q_i^+)$ and $v_i = C_s(q_i)$. The traditional knowledge distillation methods [2,3] usually use a softmax layer to produce the posterior distillation p_i, *e.g.*, given the input v_i, the posterior distillation is:

$$p_i{}^{(k)} = \frac{\exp(v_i{}^{(k)}/\tau)}{\sum_{j=1}^{K}\exp(v_i{}^{(j)}/\tau)}, k = 1, 2, ..., K, \quad (1)$$

where K is the class number, k is the class index. $v_i{}^{(k)}$ and $p_i{}^{(k)}$ is the predicted logit value and the probability of the k_{th} class, respectively. $\tau > 0$ is a temperature scaling parameter that controls the sharpness of the output distribution. According to Eq.1, we obtain the soft probability distributions of teacher and student as follows:

$$p_i^{+(k)} = \frac{\exp(v_i^{+(k)}/\tau)}{\sum_{j=1}^{K}\exp(v_i^{+(j)}/\tau)}, k = 1, 2, ..., K, \quad (2)$$

$$p_i{}^{(k)} = \frac{\exp(v_i^{(k)}/\tau)}{\sum_{j=1}^{K}\exp(v_i^{(j)}/\tau)}, k = 1, 2, ..., K. \quad (3)$$

The naive logits distillation loss $\mathcal{L}_{ld}$ can be expressed as:

$$\mathcal{L}_{ld}(p_i, p_i^+) = D_{KL}(p_i^+||p_i) = \sum_{k=1}^{K} p_i^{+(k)} \log(\frac{p_i^{+(k)}}{p_i{}^{(k)}}), \quad (4)$$

where $D_{KL}(p_i^+||p_i)$ denotes the KL divergence between p_i^+ and p_i. The gradient of the loss $\mathcal{L}_{ld}$ relative to the student's logits $v_i{}^{(k)}$ is computed as:

$$\begin{aligned}\frac{\partial \mathcal{L}_{ld}}{\partial v_i{}^{(k)}} &= \frac{1}{\tau}(p_i{}^{(k)} - p_i^{+(k)}) \\ &= \frac{1}{\tau}(\frac{\exp(v_i{}^{(k)}/\tau)}{\sum_{j=1}^{K}\exp(v_i{}^{(j)}/\tau)} - \frac{\exp(v_i^{+(k)}/\tau)}{\sum_{j=1}^{K}\exp(v_i^{+(j)}/\tau)}).\end{aligned} \quad (5)$$

If the temperature τ is high compared with the magnitude of the logits $v_i{}^{(k)}$, we have:

$$\exp(v_i{}^{(k)}/\tau) \approx v_i{}^{(k)}/\tau + 1, \tau \gg v_i{}^{(k)}. \quad (6)$$

Then we can approximate Eq.5 to:

$$\begin{aligned}\frac{\partial \mathcal{L}_{ld}}{\partial v_i{}^{(k)}} &\approx \frac{1}{\tau}(\frac{v_i{}^{(k)}/\tau + 1}{\sum_{j=1}^{K}(v_i{}^{(j)}/\tau + 1)} - \frac{v_i^{+(k)}/\tau + 1}{\sum_{j=1}^{K}(v_i^{+(j)}/\tau + 1)}) \\ &= \frac{1}{\tau}(\frac{v_i{}^{(k)}/\tau + 1}{K + \sum_{j=1}^{K}(v_i{}^{(j)}/\tau)} - \frac{v_i^{+(k)}/\tau + 1}{K + \sum_{j=1}^{K}(v_i^{+(j)}/\tau)}).\end{aligned} \quad (7)$$

Assuming that the logits have been zero-meaned separately for each sample [2] so that $\sum_{j=1}^{K} v_i{}^{(j)} = \sum_{j=1}^{K} v_i^{+(j)} = 0$, Eq.7 simplifies to:

$$\begin{aligned}\frac{\partial \mathcal{L}_{ld}}{\partial v_i{}^{(k)}} &\approx \frac{1}{\tau}(\frac{v_i{}^{(k)}/\tau}{K} - \frac{v_i^{+(k)}/\tau}{K}) \\ &= \frac{1}{K\tau^2}(v_i{}^{(k)} - v_i^{+(k)}).\end{aligned} \quad (8)$$

Denoting the normalized logits of student and teacher as $\bar{v}_i$ and $\bar{v}_i^+$, respectively:

$$\bar{v}_i = \frac{v_i}{||v_i||_2}, \bar{v}_i^+ = \frac{v_i^+}{||v_i^+||_2}, \quad (9)$$

where $||\cdot||_2$ refers to $L2$ norm of the vector, Eq.8 can be formed as:

$$\begin{aligned}\frac{\partial \mathcal{L}_{ld}}{\partial v_i{}^{(k)}} &\approx \frac{1}{K\tau^2}(v_i{}^{(k)} - v_i^{+(k)}) \\ &= \frac{||v_i^+||}{K\tau^2}(\frac{||v_i||}{||v_i^+||}\bar{v}_i{}^{(k)} - \bar{v}_i^{+(k)}).\end{aligned} \quad (10)$$

From Eq.10, we find that $||v_i||$ and $||v_i^+||$ dynamically change during training and the temperature τ reacts with the continuously updated $||v_i^+||$ that can be considered as the compensate for $||v_i^+||$. We argue that if the magnitude of the teacher and student logits are normalized, the temperature τ used to compensate for this magnitude needs no more consideration (τ always equals 1). What's more, the magnitude of the teacher and student logits bring obstacles to the optimizing process, and the temperature τ correspondingly has a great impact on the distillation performance.

1.2. Invariant Classification Results

Next, we prove the invariance of classification results before and after logits normalization. Suppose that the pre-softmax logits produced by a classification network is $v_i \in \mathbb{R}^K$ and the soft probability distribution is $p_i \in \mathbb{R}^K$, where K is the class number. We can get the indexes of the descending ordered elements of a vector by using the function $argsort(\cdot)$. The descending ordered indexes of the elements in v_i and p_i are the same, that is:

$$argsort(v_i) = argsort(p_i). \quad (11)$$

Dimention	AWA1			AWA2			CUB			FLO			APY		
	U	*S*	*H*	*U*	*S*	*H*	*U*	*S*	*H*	*U*	*S*	*H*	*U*	*S*	*H*
w/o projector	63.4	78.2	70.0	64.3	78.0	70.5	65.0	55.2	59.7	63.5	76.3	69.3	43.1	38.5	40.7
256	61.1	**83.3**	70.5	60.8	**85.3**	71.0	**68.0**	54.9	60.8	62.6	78.9	69.8	39.6	**46.9**	42.9
512	**67.4**	81.2	**73.6**	**65.3**	82.3	**72.8**	67.3	**65.5**	**66.4**	**66.1**	**86.5**	**74.9**	**45.2**	46.3	**45.7**
1024	64.6	79.9	71.5	58.1	83.4	68.5	66.4	56.3	60.9	64.8	79.3	71.3	41.1	39.8	40.5
2048	66.5	79.6	72.5	64.6	79.9	71.4	67.2	59.6	63.2	64.8	83.5	73.0	33.1	43.6	37.6

Table 1. The results of ICCE built without (w/o) or with different dimensional projectors. The best results are marked in **bold**.

Case		AWA1			AWA2			CUB			FLO			APY		
		U	*S*	*H*	*U*	*S*	*H*	*U*	*S*	*H*	*U*	*S*	*H*	*U*	*S*	*H*
†	$\tau = 1$	65.0	79.7	71.6	65	79.4	71.5	67.9	58.6	62.9	65.1	80.3	71.9	34.0	45.1	38.8
	$\tau = 10$	67.2	80.4	73.2	62.1	83.3	71.2	**71.7**	57.5	63.8	65.5	79.3	71.7	39.3	31.3	34.9
	$\tau = 20$	64.4	**81.8**	72.0	62.4	83.7	71.5	69.7	57.8	63.2	64.2	86.0	73.6	35.1	47.7	40.4
	$\tau = 50$	65.7	81.1	72.6	60.6	**83.9**	70.4	71.1	60.0	65.1	64.4	84.0	72.9	31.9	**57.9**	41.1
	$\tau = 100$	66.6	80.5	72.9	62.3	83.2	71.2	68.1	63.5	65.7	65.8	80.8	72.5	38.5	49.2	43.2
	$\tau = 150$	64.7	81.0	72.0	61.7	82.2	70.5	70.0	61.6	65.5	65.1	83.3	73.1	**47.6**	40.2	43.6
	$\tau = 250$	66.7	77.6	71.7	62.5	81.6	70.8	70.8	56.7	63.0	65.8	81.0	72.6	37.1	37.5	37.3
‡	$\tau = 1$	**67.4**	81.2	**73.6**	**65.3**	82.3	**72.8**	67.3	**65.5**	**66.4**	**66.1**	**86.5**	**74.9**	45.2	46.3	**45.7**

Table 2. The impact of temperature hyperparameter in the logits distillation loss. † and ‡ denote two ways to produce soft posterior distribution. † indicates that straightly applying softmax operation with different temperatures τ on the teacher and student output logits. ‡ denotes using regular softmax operation ($\tau = 1$) on the normalized logits. The best results are marked in **bold**.

Decay rate (ξ)	AWA1	AWA2	CUB	FLO	APY
0.9	72.1	71.1	64.6	72.1	**45.7**
0.91	72.1	71.7	**66.4**	74.2	43.2
0.93	71.6	71.7	65.0	**74.9**	41.0
0.95	72.6	71.2	64.7	73.4	41.4
0.97	72.2	72.4	63.8	73.0	42.3
0.98	71.0	**72.8**	65.3	73.7	44.9
0.99	**73.6**	70.5	63.9	72.1	43.8

Table 3. The effect of different teacher decay rates (ξ) on harmonic mean H, the best results are marked in **bold**.

Zooming out each element in v_i by $||v_i||$, the relative scale of them will not be changed, that is:

$$\bar{v_i}^{(k)} = \frac{{v_i}^{(k)}}{||v_i||}, k = 1, 2, ..., K, \quad argsort(\bar{v_i}) = argsort(v_i). \tag{12}$$

The descending ordered indexes of the soft probability distribution $\bar{p_i}$ produced by the normalized logits $\bar{v_i}$ keep the same with that of $\bar{v_i}$:

$$argsort(\bar{v_i}) = argsort(\bar{p_i}). \tag{13}$$

Combining Eq.11, Eq.12, Eq.13, we can get:

$$argsort(p_i) = arsort(\bar{p_i}) = argsort(v_i) = argsort(\bar{v_i}). \tag{14}$$

From the above equation, it can be inferred that the predicted classification results stay the same whether normalizing the logits or not.

2. Additional Ablations

Existence of the projector. In Table 1, we discuss the GZSL classification results of our ICCE without or with different dimensional projectors. As introduced in [1], the projection improves the representation quality of the input embeddings; we find that including a projection head is crucial. When adopting a projector, the accuracies of seen class are further boosted, indicating that more information is maintained in the embedding space. From Table 1, we notice that 512 dimensional projector constrains the embedding better, significantly improving the performance.

Effect of temperature hyperparameter. In Table 2, we study the impact of temperature hyperparameter in the logits distillation loss. † and ‡ denote two ways to produce soft posterior distribution: (1) straightly applying softmax operation with different temperatures τ on the output logits (denotes as †); (2) applying regular softmax operation ($\tau = 1$) on the normalized logits (denotes as ‡). From Table 2, we find out that τ greatly impacts the GZSL classification performance. Searching for a proper τ is quite difficult

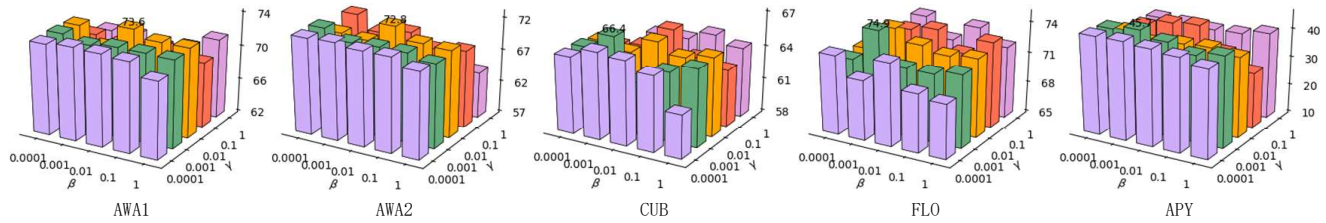

Figure 1. The results of harmonic mean H in GZSL with respect to different loss weights β and γ. The best H on each dataset is annotated.

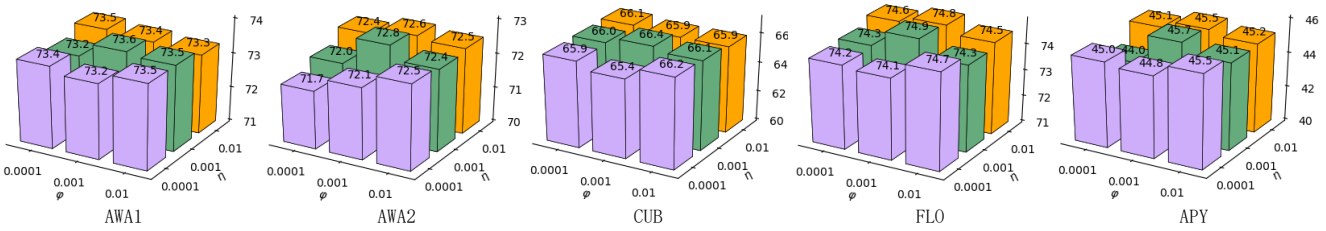

Figure 2. The results of harmonic mean H in GZSL with respect to different loss weights η and φ.

and time-consuming. However, if we normalize the output logits, the temperature used to compensate logits magnitude needs no more consideration (τ always equals 1), and the results achieve the best in all cases.

Effect of teacher decay rate. In Table 3, we compare the harmonic mean H effected by different teacher decay rates (ξ) on five datasets. We observe that our method benefits from different ξ on different datasets. For instance, on AWA1 and AWA2, a large decay rate will lead to best results (0.99 for AWA1, 0.98 for AWA2), while on CUB, FLO, and APY, a smaller ξ achieves better performance (0.91 for CUB, 0.93 for FLO, 0.9 for APY).

Influence of β and γ. In Figure 1, we explore the influence of loss weight β and γ in the self-distillation embedding module. We cross-validate the two parameters in $[0.0001, 0.001, 0.01, 0.1, 1]$ and plot the H values with respect to various parameter pairs. When applying different parameters, the results change relatively. Our method achieves the best results when $\beta = \gamma = 0.01$ on AWA1 and AWA2, $\beta = \gamma = 0.001$ on CUB, FLO, and APY.

Exploration of η and φ. In Figure 2, we investigate the loss weight η and φ in the overall objective function of ICCE (Eq.13 in the main paper). We cross-validate the two parameters in $[0.0001, 0.001, 0.01]$ and plot the H values with respect to various parameter pairs. Our method achieves the best results when $\eta = \varphi = 0.001$ on AWA1, AWA2, CUB, FLO, and APY.

References

[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 2

[2] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv:1503.02531*, 2015. 1

[3] Sukmin Yun, Jongjin Park, Kimin Lee, and Jinwoo Shin. Regularizing class-wise predictions via self-knowledge distillation. In *CVPR*, 2020. 1