# Learning Imbalanced Data with Vision Transformers
# Supplementary Material

## A. Missing Proofs and Derivations

### A.1. Proof to Theorem 1

**Theorem 1.** Logit Bias of Balanced CE. Let $\pi_{\mathbf{y}_i} = n_{\mathbf{y}_i}/N$ be the training label $\mathbf{y}_i$ distribution. If we implement the balanced cross-entropy loss via logit adjustment, the bias item of logit $\mathbf{z}_{\mathbf{y}_i}$ will be $\mathcal{B}^{\text{ce}}_{\mathbf{y}_i} = \log \pi_{\mathbf{y}_i}$, *i.e.*,

$$\mathcal{L}_{\text{Bal-CE}} = \log[1 + \sum_{\mathbf{y}_j \neq \mathbf{y}_i} e^{(\mathbf{z}_{\mathbf{y}_j} + \log \pi_{\mathbf{y}_j}) - (\mathbf{z}_{\mathbf{y}_i} + \log \pi_{\mathbf{y}_i})}].$$

***Proof.***

Following the notions in Section Preliminaries, we simplify a model $\mathcal{M}_\theta$ with parameters $\theta$, which attempts to learn the joint probability distribution of images and labels $\mathcal{P}(\mathcal{X}, \mathcal{Y})$. Due to its agnostic, one may try to get the maximum posterior $\mathcal{P}(\mathcal{Y}|\mathcal{X})$ as an approximation solution from the Bayesian estimation view. To this end, if we Maximize A Posterior (MAP) to optimize $\theta$, we have:

$$\hat{\theta} = \arg\max_\theta \mathcal{P}(\mathcal{Y}|\mathcal{X}) = \arg\max_\theta \frac{\mathcal{P}(\mathcal{X}|\mathcal{Y}) \cdot \mathcal{P}(\mathcal{Y})}{\mathcal{P}(\mathcal{X})} = \arg\max_\theta \mathcal{P}(\mathcal{X}|\mathcal{Y}) \cdot \mathcal{P}(\mathcal{Y}),$$

where $\mathcal{P}(\mathcal{X}|\mathcal{Y})$ is the likelihood function, $\mathcal{P}(\mathcal{Y})$ is the prior distribution of $\mathcal{Y}$, and $\mathcal{P}(\mathcal{X})$ is the evidence factor, which is $\theta$ irrelevant. Then, if we reasonably view $\mathcal{P}(\mathcal{Y})$ as the class distribution (typically class label frequency $\pi_{\mathbf{y}_i}$ as approximations), the MAP is equivalent to maximizing the likelihood function $\mathcal{P}(\mathcal{X}|\mathcal{Y};\theta)$. Considering both training $\mathcal{P}^s(\mathcal{X}, \mathcal{Y})$ and test datasets $\mathcal{P}^t(\mathcal{X}, \mathcal{Y})$, the MAP shall hold on to both of them, *i.e.*,

$$\begin{cases} \hat{\theta} = \arg\max_\theta \mathcal{P}^s(\mathcal{Y}|\mathcal{X}) = \arg\max_\theta \mathcal{P}^s(\mathcal{X}|\mathcal{Y};\theta) \cdot \mathcal{P}^s(\mathcal{Y}) \\ \hat{\theta} = \arg\max_\theta \mathcal{P}^t(\mathcal{Y}|\mathcal{X}) = \arg\max_\theta \mathcal{P}^t(\mathcal{X}|\mathcal{Y};\theta) \cdot \mathcal{P}^t(\mathcal{Y}) \end{cases}$$

With model parameters $\theta$ learned on the training set $\mathcal{P}^s(\mathcal{X}, \mathcal{Y})$, the likelihood function will be consistent. To obtain the maximization posterior on the test dataset (the best accuracy performance), we can derive that:

$$\mathcal{P}^t(\mathcal{Y}|\mathcal{X};\theta) \quad \propto \quad \mathcal{P}^t(\mathcal{X}|\mathcal{Y};\theta)\mathcal{P}^t(\mathcal{Y}) \quad \propto \quad \frac{\mathcal{P}^s(\mathcal{Y}|\mathcal{X};\theta)}{\mathcal{P}^s(\mathcal{Y})} \cdot \mathcal{P}^t(\mathcal{Y})$$

Since MAP is equivalent to maximizing the likelihood function $\mathcal{P}(\mathcal{X}|\mathcal{Y};\theta)$, we further decouple the test MAP as regulation terms to achieve the Structural Risk Minimization:

$$\arg\max_\theta \mathcal{P}^t(\mathcal{Y}|\mathcal{X};\theta) \;=\; \arg\max_\theta \log \mathcal{P}^t(\mathcal{Y}|\mathcal{X};\theta) \;=\; \arg\max_\theta \log \mathcal{P}^s(\mathcal{X}|\mathcal{Y};\theta) - \log \mathcal{P}^s(\mathcal{Y}) + \log \mathcal{P}^t(\mathcal{Y})$$

Notice that $\mathcal{P}^s(\mathcal{Y})$ and $\mathcal{P}^t(\mathcal{Y})$ are both $\theta$ irrelevant according to our previous hypothesis. Hence, we can compensate the regulation terms $-\log \mathcal{P}^s(\mathcal{Y}) + \log \mathcal{P}^t(\mathcal{Y})$ during the training procession as $+\log \pi^s(\mathbf{y}) - \log \pi^t(\mathbf{y})$. In addition, if we adopt the *Softmax* for probability normalization, we will have:

$$\mathcal{P}^s(\mathbf{x}_i|\mathbf{y}_i;\theta) = \frac{e^{\mathbf{z}_{\mathbf{y}_i}}}{\sum_{\mathbf{y}_j \in C} e^{\mathbf{z}_{\mathbf{y}_j}}} \quad \Longrightarrow \quad \log \mathcal{P}^s(\mathbf{x}_i|\mathbf{y}_i;\theta) = \log \frac{e^{\mathbf{z}_{\mathbf{y}_i}}}{\sum_{\mathbf{y}_j \in C} e^{\mathbf{z}_{\mathbf{y}_j}}} \quad \propto \quad \log e^{\mathbf{z}_{\mathbf{y}_i}} = \mathbf{z}_{\mathbf{y}_i}$$

Thus, the $\log \mathcal{P}^s(\mathcal{X}|\mathcal{Y};\theta)$ is equivalent to the output logits $\mathbf{z} := \mathcal{M}(\mathbf{x}|\theta)$ and we immediately deduce that the training regulation shall be $\mathbf{z}_\mathbf{y} + \log \pi^s_\mathbf{y} - \log \pi^t_\mathbf{y}$. For the balanced test datasets, $-\log \pi^t_\mathbf{y} = \log C$ and can be ignored for all classes. Hence, we derive the final bias as:

$$\mathcal{B}^{\text{ce}}_{\mathbf{y}_i} = \log \pi^s_{\mathbf{y}_i}$$

□

## A.2. Proof to Theorem 2&3

**Theorem 2&3.** Logit Bias of Balanced BCE with Test Prior. Let $\pi^s_{\mathbf{y}_i}$ and $\pi^t_{\mathbf{y}_i}$ be the label $\mathbf{y}_i$ training and test distribution. If we implement the balanced cross-entropy loss via logit adjustment, the bias item of logit $\mathbf{z}_{\mathbf{y}_i}$ will be:

$$\mathcal{B}^{\text{bce}}_{\mathbf{y}_i} = (\log \pi^s_{\mathbf{y}_i} - \log \pi^t_{\mathbf{y}_i}) - (\log(1 - \pi^s_{\mathbf{y}_i}) - \log(1 - \pi^t_{\mathbf{y}_i}))$$

***Proof.***

In this paper, we propose the balanced binary cross entropy loss in Thm. 2 and further extend it with the test prior (test label distribution) in Thm. 3. As we discussed, the bias in Thm. 2 is derived from re-balancing with training instance numbers like [51] do. Here, we give another proof from the Bayesian estimation view like Thm. 1. We mainly give the proof to the Thm. 3 and derive the Thm. 2 as a special case of Thm. 3. Following the notions in the proof to Thm. 1, BCE loss treats the long-tailed recognition task as $C$ independent binary classification problems. For every single problem, the derivation in Thm. 1 still holds if $\mathcal{Y} := \{0, 1\}$:

$$\arg\max_{\theta} \mathcal{P}^t(\mathcal{Y}|\mathcal{X};\theta) \;=\; \arg\max_{\theta} \log \mathcal{P}^t(\mathcal{Y}|\mathcal{X};\theta) \;=\; \arg\max_{\theta} \log \mathcal{P}^s(\mathcal{X}|\mathcal{Y};\theta) - \log \mathcal{P}^s(\mathcal{Y}) + \log \mathcal{P}^t(\mathcal{Y})$$

If we adopt the *Sigmoid* for probability normalization, we will have:

$$\mathcal{P}^s(\mathbf{x}_i|\mathbf{y}_i;\theta) = \frac{1}{1 + e^{-\mathbf{z}_{\mathbf{y}_i}}} \quad \Longrightarrow \quad \frac{e^{\mathbf{z}_{\mathbf{y}_i}}}{e^{\mathbf{z}_{\mathbf{y}_i}} + e^0}$$

Similar to the *Softmax*, for the binary classification, we consider the $e^{\mathbf{z}_{\mathbf{y}_i}}/(e^{\mathbf{z}_{\mathbf{y}_i}} + e^0)$ as the likelihood for $\mathcal{Y} = 1$ and $e^0/(e^{\mathbf{z}_{\mathbf{y}_i}} + e^0)$ for $\mathcal{Y} = 0$. Then, we can derive that:

$$\log \mathcal{P}^s(\mathbf{x}_i|\mathbf{y}_i;\theta) = \log \frac{e^{\mathbf{z}_{\mathbf{y}_i}}}{e^{\mathbf{z}_{\mathbf{y}_i}} + e^0} \quad \propto \quad \log e^{\mathbf{z}_{\mathbf{y}_i}} = \mathbf{z}_{\mathbf{y}_i}$$

Different from CE, which just punishes the positive term, BCE shall take the negative terms into consideration as well. If we take the statistical label frequency $\pi^s_{\mathbf{y}}$ and $\pi^t_{\mathbf{y}}$ as the prior, we can deduce that the bias should be:

$$\begin{cases} \log \pi^s_{\mathbf{y}} - \log \pi^t_{\mathbf{y}} & \textit{for positive item } \mathbf{z}_{\mathbf{y}_i} \\ \log(1 - \pi^s_{\mathbf{y}}) - \log(1 - \pi^t_{\mathbf{y}}) & \textit{for negative item } 0 \end{cases}$$

Hence, for a single binary classification, the unbiased *Sigmoid* operation is required to compensate for each term:

$$\sigma(\mathbf{z}_{\mathbf{y}_i}) = \frac{e^{\mathbf{z}_{\mathbf{y}_i}}}{e^{\mathbf{z}_{\mathbf{y}_i}} + e^0} \Longrightarrow \frac{e^{\mathbf{z}_{\mathbf{y}_i} + \log \pi^s_{\mathbf{y}_i} - \log \pi^t_{\mathbf{y}_i}}}{e^{\mathbf{z}_{\mathbf{y}_i} + \log \pi^s_{\mathbf{y}_i} - \log \pi^t_{\mathbf{y}_i}} + e^{0 + \log(1 - \pi^s_{\mathbf{y}_i}) - \log(1 - \pi^t_{\mathbf{y}_i})}}$$

To match the Logit Adjustment requirement [47], we convert all bias to the logit $\mathbf{z}_{\mathbf{y}_i}$:

$$\begin{aligned} \sigma(\mathbf{z}_{\mathbf{y}_i}) = \frac{e^{\mathbf{z}_{\mathbf{y}_i}}}{e^{\mathbf{z}_{\mathbf{y}_i}} + e^0} &\Longrightarrow \frac{e^{\mathbf{z}_{\mathbf{y}_i} + \log \pi^s_{\mathbf{y}_i} - \log \pi^t_{\mathbf{y}_i} - (\log(1 - \pi^s_{\mathbf{y}_i}) - \log(1 - \pi^t_{\mathbf{y}_i}))}}{e^{\mathbf{z}_{\mathbf{y}_i} + \log \pi^s_{\mathbf{y}_i} - \log \pi^t_{\mathbf{y}_i} - (\log(1 - \pi^s_{\mathbf{y}_i}) - \log(1 - \pi^t_{\mathbf{y}_i}))} + e^0} \\ &= \frac{1}{1 + e^{-[\mathbf{z}_{\mathbf{y}_i} + (\log \pi^s_{\mathbf{y}_i} - \log \pi^t_{\mathbf{y}_i}) - (\log(1 - \pi^s_{\mathbf{y}_i}) - \log(1 - \pi^t_{\mathbf{y}_i}))]}} \end{aligned}$$

Hence, we get the final bias with train and test label prior knowledge:

$$\mathcal{B}^{\text{bce}}_{\mathbf{y}_i} = (\log \pi^s_{\mathbf{y}_i} - \log \pi^t_{\mathbf{y}_i}) - (\log(1 - \pi^s_{\mathbf{y}_i}) - \log(1 - \pi^t_{\mathbf{y}_i}))$$

For the balanced test dataset, $\pi^t_{\mathbf{y}_i} = 1/C$ and the $\mathcal{B}^{\text{bce}}_{\mathbf{y}_i}$ will be the form in Thm. 2 if we ignore constant terms.

$$\mathcal{B}^{\text{bce}}_{\mathbf{y}_i} = (\log \pi^s_{\mathbf{y}_i} - \log \frac{1}{C}) - (\log(1 - \pi^s_{\mathbf{y}_i}) - \log(1 - \frac{1}{C})) = \log \pi^s_{\mathbf{y}_i} - \log(1 - \pi^s_{\mathbf{y}_i}) + \log(C - 1)$$

□

### A.3. Fisher Consistency with Test Prior

Menon *et al.* show how to verify whether a pair-wise loss ensures Fisher consistency for the balanced error (see the Theorem 1 in [47]). Here, we extend it to test the prior available situations.

$$\mathcal{L}(\mathbf{y}_i, \mathcal{M}(\mathbf{x})) = \alpha_{\mathbf{y}_i} \cdot \log[1 + \sum_{\mathbf{y}_j \neq \mathbf{y}_i} Exp(\Delta_{\mathbf{y}_i\mathbf{y}_j}) \cdot Exp(\mathcal{M}_{\mathbf{y}_j}(\mathbf{x}) - \mathcal{M}_{\mathbf{y}_i}(\mathbf{x}))]$$

**Theorem 4.** For any $\delta^s, \delta^t \in \mathbb{R}_+^C$, the pairwise loss is Fisher consistent with weights and margins:

$$\alpha_{\mathbf{y}_i} = \frac{\delta^s_{\mathbf{y}_i} \cdot \pi^t_{\mathbf{y}_i}}{\delta^t_{\mathbf{y}_i} \cdot \pi^s_{\mathbf{y}_i}} \quad \Delta_{\mathbf{y}_i\mathbf{y}_j} = \log\left(\frac{\delta^s_{\mathbf{y}_j} \cdot \delta^t_{\mathbf{y}_i}}{\delta^t_{\mathbf{y}_j} \cdot \delta^s_{\mathbf{y}_i}}\right)$$

With $\delta^s_{\mathbf{y}_i} = \pi^s_{\mathbf{y}_i}$ and $\delta^t_{\mathbf{y}_i} = \pi^t_{\mathbf{y}_i}$, we deduce that Bal-BCE is Fisher consistent between train (s) and test (t) set.

***Proof.***

Let $\Delta_{\mathbf{y}_i\mathbf{y}_j} = \log\left(\frac{\delta^s_{\mathbf{y}_j} \cdot \delta^t_{\mathbf{y}_i}}{\delta^t_{\mathbf{y}_j} \cdot \delta^s_{\mathbf{y}_i}}\right)$ and $\alpha_{\mathbf{y}_i} = 1$, we have:

$$\mathcal{L}(\mathbf{y}_i, \mathcal{M}(\mathbf{x})) = -\log \frac{e^{\mathbf{z}_{\mathbf{y}_i} + \log \delta^s_{\mathbf{y}_i} - \log \delta^t_{\mathbf{y}_i}}}{\sum_{\mathbf{y}_j \in \mathcal{Y}} e^{\mathbf{z}_{\mathbf{y}_j} + \log \delta^s_{\mathbf{y}_j} - \log \delta^t_{\mathbf{y}_j}}}$$

If $\eta_{\mathbf{y}_i}(\mathbf{x})$ represents the posterior possibility $\mathcal{P}^s(\mathbf{y}_i|\mathbf{x})$, the Bayes-optimal score will satisfy:

$$\mathbf{z}^*_{\mathbf{y}_i} + \log \delta^s_{\mathbf{y}_i} - \log \delta^t_{\mathbf{y}_i} = \log \eta_{\mathbf{y}_i}(\mathbf{x}) \quad \Longrightarrow \quad \mathbf{z}^*_{\mathbf{y}_i} = \log\left(\frac{\eta_{\mathbf{y}_i}(\mathbf{x})}{\delta^s_{\mathbf{y}_i}} \cdot \delta^t_{\mathbf{y}_i}\right)$$

Now consider adding weights $\alpha_{\mathbf{y}_i}$ to the loss term, the corresponding risk shall be:

$$\mathbb{E}_{\mathbf{x},\mathbf{y}}\left[\mathcal{L}_{\alpha_{\mathbf{y}_i}}\right] \quad = \quad \sum_{\mathbf{y}_i \in \mathcal{Y}} \pi^s_{\mathbf{y}_i} \cdot \mathbb{E}_{\mathbf{x}|\mathbf{y}=\mathbf{y}_i}\left[\mathcal{L}_{\alpha_{\mathbf{y}_i}}\right] \quad = \quad \sum_{\mathbf{y}_i \in \mathcal{Y}} \pi^s_{\mathbf{y}_i} \cdot \alpha_{\mathbf{y}i} \cdot \mathbb{E}_{\mathbf{x}|\mathbf{y}=\mathbf{y}_i}[\mathcal{L}] \quad \propto \quad \sum_{\mathbf{y}_i \in \mathcal{Y}} \bar{\pi}^s_{\mathbf{y}_i} \cdot \mathbb{E}_{\mathbf{x}|\mathbf{y}=\mathbf{y}_i}[\mathcal{L}]$$

where $\bar{\pi}^s_{\mathbf{y}_i} \propto \pi^s_{\mathbf{y}_i} \cdot \alpha_{\mathbf{y}_i}$. Hence training with the weighted loss amounts to training with the original loss on the new label distribution $\bar{\pi}$. The posterior probability $\bar{\eta}_{\mathbf{y}_i}(\mathbf{x})$ on the altered label distribution is:

$$\bar{\eta}_{\mathbf{y}_i}(\mathbf{x}) = \overline{\mathcal{P}}\left(\mathbf{y}_i|\mathbf{x}\right) \quad \propto \quad \mathcal{P}\left(\mathbf{x}|\mathbf{y}_i\right) \cdot \bar{\pi}^s_{\mathbf{y}_i} \quad \propto \quad \eta_{\mathbf{y}_i}(\mathbf{x}) \cdot \frac{\bar{\pi}^{\mathrm{s}}_{\mathbf{y}_i}}{\pi^s_{\mathbf{y}_i}} \quad \propto \quad \eta_{\mathbf{y}_i}(\mathbf{x}) \cdot \alpha_{\mathbf{y}_i}$$

When we set $\alpha_{\mathbf{y}_i} = \frac{\delta^s_{\mathbf{y}_i} \cdot \pi^t_{\mathbf{y}_i}}{\delta^t_{\mathbf{y}_i} \cdot \pi^s_{\mathbf{y}_i}}$, the Bayes-optimal score will satisfy:

$$\begin{aligned}
\arg\max_{\mathbf{y}_i \in \mathcal{Y}} \mathbf{z}^*_{\mathbf{y}_i} \quad &= \quad \arg\max_{\mathbf{y}_i \in \mathcal{Y}} \log\left(\frac{\bar{\eta}_{\mathbf{y_i}}(\mathbf{x})}{\delta^s_{\mathbf{y}_i}} \cdot \delta^t_{\mathbf{y}_i}\right) \\
&= \quad \arg\max_{\mathbf{y}_i \in \mathcal{Y}} \log\left(\frac{\eta_{\mathbf{y}_i}(\mathbf{x}) \cdot \alpha_{\mathbf{y}_i}}{\delta^s_{\mathbf{y_i}}} \cdot \delta^t_{\mathbf{y}_i}\right) \\
&= \quad \arg\max_{\mathbf{y}_i \in \mathcal{Y}} \log\left(\frac{\eta_{\mathbf{y}_i}(\mathbf{x})}{\pi^s_{\mathbf{y}_i}} \cdot \pi^t_{\mathbf{y}_i}\right)
\end{aligned}$$

□

## B. Analysis to Proposed Bias

For Bal-CE, Ren *et al.* [51] propose the balanced softmax as a strong baseline for long-tailed recognition while Menon *et al* [47] deploy it by adding extra logit margins. The following works [22, 70] further extend it with test prior knowledge, which can be written as:

$$\mathcal{B}^{\text{ce}}_{\mathbf{y}_i} = \log \pi^s_{\mathbf{y}_i} + \log C$$

To improve the performance of balanced binary cross-entropy loss in long-tailed recognition, we propose an unbiased version of *Sigmoid* to eliminate the inherent bias to the head class. Inspired by Logit Adjustment [47], we implement it as a bias $\mathcal{B}^{\text{bce}}_{\mathbf{y}_i}$ to the model logits and extend to test prior as well, which can be written as:

$$\mathcal{B}^{\text{bce}}_{\mathbf{y}_i} = \log \pi^s_{\mathbf{y}_i} - \log(1 - \pi^s_{\mathbf{y}_i}) + \log(C - 1)$$

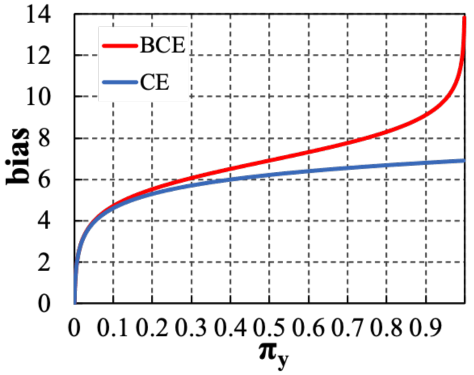


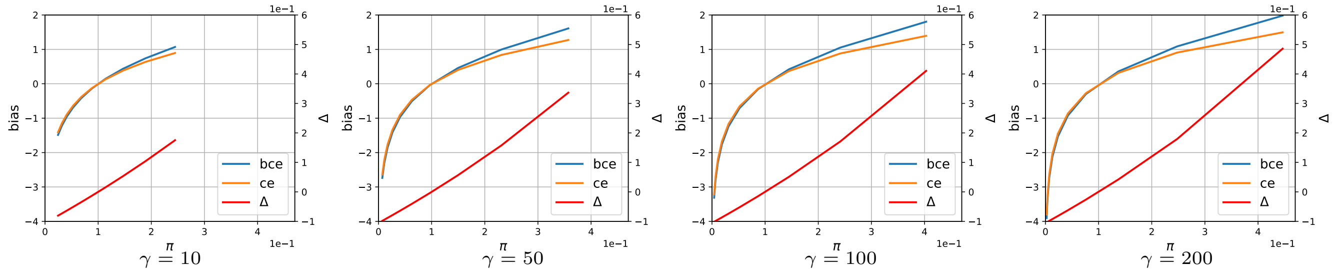
Figure 4. $\mathcal{B}^{\text{ce}}_{\mathbf{y}_i}$ and $\mathcal{B}^{\text{bce}}_{\mathbf{y}_i}$ *w.r.t.* $\pi_{\mathbf{y}_i}$ ($C$=1,000).

Fig. 4 shows the difference between $\mathcal{B}^{\text{ce}}_{\mathbf{y}_i}$ and $\mathcal{B}^{\text{bce}}_{\mathbf{y}_i}$. Notice that $\mathcal{B}^{\text{bce}}_{\mathbf{y}_i}$ is closed to $\mathcal{B}^{\text{ce}}_{\mathbf{y}_i}$ when $\pi_{\mathbf{y}_i}$ is small, which indicates that both $\mathcal{B}^{\text{ce}}_{\mathbf{y}_i}$ and $\mathcal{B}^{\text{bce}}_{\mathbf{y}_i}$ help the models to pay more attention to learn the tail. However, $\mathcal{B}^{\text{bce}}_{\mathbf{y}_i}$ gives larger biases to the head and makes the inter-class distance of the head smaller. Such a modification allows Bal-BCE to show more tolerance to the head compared to Bal-CE. To be more specific, CE utilizes *Softmax* to emphasize mutual exclusion, where large head bias will damage corresponding performance severely. In contrast, BCE calculates independent class-wise probability with *Sigmoid* function, where the original task is considered as a series of binary classification tasks. Hence, the head bias will not influence the tail. In addition, larger biases will not hurt the head as CE does because it hedges the over-suppression for negative labels. CE can not benefit from it because of its mutual exclusion.

$$\begin{cases} \dfrac{\partial \mathcal{L}_{\text{Bal-CE}}\left(\mathbf{z}_{\mathbf{y}_j}, \mathbb{1}(\mathbf{y}_j)\right)}{\partial\left(\mathbf{z}_{\mathbf{y}_j}\right)} = \dfrac{e^{\mathbf{z}_{\mathbf{y}_j} + \mathcal{B}^{\text{ce}}_{\mathbf{y}_j}}}{\sum_{\mathbf{y}_i \in \mathcal{C}} e^{\mathbf{z}_{\mathbf{y}_i} + \mathcal{B}^{\text{ce}}_{\mathbf{y}_i}}}, \quad \dfrac{\partial \mathcal{L}_{\text{Bal-BCE}}\left(\mathbf{z}_{\mathbf{y}_j}, \mathbb{1}(\mathbf{y}_j)\right)}{\partial\left(\mathbf{z}_{\mathbf{y}_j}\right)} = \dfrac{e^{\mathbf{z}_{\mathbf{y}_j} + \mathcal{B}^{\text{bce}}_{\mathbf{y}_j}}}{1 + e^{\mathbf{z}_{\mathbf{y}_j} + \mathcal{B}^{\text{bce}}_{\mathbf{y}_j}}}, & \mathbb{1}(\mathbf{y}_j) = 0 \\ \dfrac{\partial \mathcal{L}_{\text{Bal-CE}}\left(\mathbf{z}_{\mathbf{y}_j}, \mathbb{1}(\mathbf{y}_j)\right)}{\partial\left(\mathbf{z}_{\mathbf{y}_j}\right)} = \dfrac{e^{\mathbf{z}_{\mathbf{y}_j} + \mathcal{B}^{\text{ce}}_{\mathbf{y}_j}}}{\sum_{\mathbf{y}_i \in \mathcal{C}} e^{\mathbf{z}_{\mathbf{y}_i} + \mathcal{B}^{\text{ce}}_{\mathbf{y}_i}}}, \quad \dfrac{\partial \mathcal{L}_{\text{Bal-BCE}}\left(\mathbf{z}_{\mathbf{y}_j}, \mathbb{1}(\mathbf{y}_j)\right)}{\partial\left(\mathbf{z}_{\mathbf{y}_j}\right)} = -\dfrac{1}{1 + e^{\mathbf{z}_{\mathbf{y}_j} + \mathcal{B}^{\text{bce}}_{\mathbf{y}_j}}}, & \mathbb{1}(\mathbf{y}_j) = 1 \end{cases}$$

From the optimization view, as the above equation shows, we can also observe that $\mathcal{B}^{\text{bce}}_{\mathbf{y}_i}$ will not affect class $\mathbf{y}_j$'s gradients. However, for Bal-CE, the optimization step would be rather small once the logit for the positive class is much higher than those of the negative ones. With the dominance of head labels, larger head biases will make the networks fall into even worse situations. In contrast, for the Bal-BCE, the above larger head biases will act as a regularization to overcome the over-suppression while avoiding damage to the head classes themselves.

In addition, $\mathcal{B}^{\text{bce}}_{\mathbf{y}_i}$ will be more important when the datasets become more skewed. As Fig. 5 shows, the difference will be larger when the imbalance factor $\gamma$ increases. It means the performance will get worse if we adopt $\mathcal{B}^{\text{ce}}_{\mathbf{y}_i}$ for BCE loss. Notice that the gap between $\mathcal{B}^{\text{ce}}_{\mathbf{y}_i}$ and $\mathcal{B}^{\text{bce}}_{\mathbf{y}_i}$ has consistent diminution when the class number $C$ is getting bigger. However, $\mathcal{B}^{\text{bce}}_{\mathbf{y}_i}$ still bring obvious performance gain in this circumstance.



$\gamma = 10$ $\gamma = 50$ $\gamma = 100$ $\gamma = 200$

Figure 5. Visualization of the bias in CIFAR10-LT dataset. A larger $\gamma$ indicates a severer imbalance situation. $\Delta$ is the difference between the two biases, which is shown in right *y*-axis. With $\gamma$ increases, the $\Delta$ becomes more important to the final bias.

## C. Datasets

We conduct experiments on CIFAR-LT [32], ImageNet-LT [52], iNat18 [57], and Places-LT [82]. With different imbalanced factors $\gamma$, we build the long-tailed version of CIFAR by discarding training instances following the rule given in [13] and keeping the original validation set for all datasets. To investigate the MGP performance on LT data, we build a balanced ImageNet-1K subset called ImageNet-BAL. It contains the same training instance number as ImageNet-LT while keeping class labels balanced. Notice that both LT and BAL adopt the same validation set. We demonstrate MGP is robust enough for long-tailed data via quantitative and qualitative experiments on the BAL and LT. iNat18 is the largest benchmark of the long tail community. Our LiVT ameliorates vanilla ViTs most significantly because of the data scale and fine-grained problems. Places-LT is created from large-scale dataset Places [82] by [44]. The train set contains just 62K images with a high imbalance factor, which makes it challenging for data-hungry Transformers.

Table 8. Detailed information of datasets motioned in the main paper.

| Dataset | CIFAR-10-LT | | CIFAR-100-LT | | ImageNet-LT | ImageNet-BAL | iNat18 | PlaceLT |
|---|---|---|---|---|---|---|---|---|
| | Imbalance Factor ($\gamma$) | | | | | | | |
| | 100 | 10 | 100 | 10 | | | | |
| Training Images | 12,406 | 20,431 | 10,847 | 19,573 | 115,846 | 160,000 | 437,513 | 62,500 |
| Classes Number | 10 | 10 | 100 | 100 | 1,000 | 1,000 | 8,142 | 365 |
| Max Images | 5,000 | 5,000 | 500 | 500 | 1,280 | 160 | 1,000 | 4,980 |
| Min Images | 50 | 500 | 5 | 50 | 5 | 160 | 2 | 5 |
| Imbalance Factor | 100 | 10 | 100 | 10 | 256 | 1 | 500 | 996 |

## D. Implementation Details

### D.1. Augmentations in Algorithm.1

In Alg. 1, LiVT adopts different augmentations in two stages, *i.e.*, $\mathcal{A}_{pt}$ & $\mathcal{A}_{ft}$. The reason is from our observations that the strong data augmentations in MGP will not contribute to higher performance while bringing extra calculation burden. Some augmentations like Color Jitter may lead to wired reconstruction results *w.r.t.* the augmented images. For the BFT stage, we adopt more general data augmentations for stable training procession. The AutoAug improves performance on ImageNet-LT/BAL remarkably and slightly in iNat18 / Places-LT, which is consistent with the observation in [12]. Mixup and Cutmix make the training more smooth, and RandomErease regulates the model with better performance.

Table 9. The detailed augmentations adopted in Alg. 1.

| Augmentation | Masked Generative Pretraining ($\mathcal{A}_{pt}$) | Balanced Fine Tuning ($\mathcal{A}_{ft}$) |
|---|---|---|
| RandomResizedCrop | ✓ | ✓ |
| RandomHorizontalFlip | ✓ | ✓ |
| AutoAug | × | (9,0.5) |
| Mixup | × | 0.8 |
| Cutmix | × | 1.0 |
| RandomErease | × | 0.25 |
| Normalize | ✓ | ✓ |

### D.2. Configure Settings for Table 1

In Tab. 1, we implement different ViT training recipes on long-tailed and balanced ImageNet-1K subsets. Specially, we reproduce vanilla ViTs according to Tab.11 in [18], DeiT III according to Tab.1 in [55], and MAE according to Tab.9 in [18]. All recipes train ViTs with more epochs (800) compared to ResNets (typically 90 or 180). However, the performance is far from catching up with ResNet baselines and severely deteriorate when it becomes imbalanced because the dataset is relatively small for data-hungry ViTs compared to ImageNet-1K or ImageNet22K and the long-tailed labels bias the ViTs heavily.

### D.3. Configure Settings for the Main Comparisons

We conduct experiments on ImageNet-LT, iNat18, and Places-LT. For fair comparisons, we train all models from scratch following previous LTR work. To balance the performance and computation complexity trade-off, we adopt a small image size for the large-scale dataset and adopt 800 epochs for MGP. Thanks to the masked tokens, MGP trains ViTs much faster than vanilla ViT and DeiT. We transfer the hyper-parameters of ImageNet-LT to other benchmarks and just finetune the $\tau$ of Bal-BCE loss slightly. Notice that Places-LT is a small dataset and we just finetune 30 epochs to avoid over-fitting.

Table 10. The LiVT configurations on three main benchmarks. We mainly transfer the hyper-parameters of ImageNet-LT to other benchmarks without wide changes. Tuning hyper-parameters will bring further improvement.

| Configuration | ImageNet-LT | iNaturalist 2018 | Places-LT |
|---|---|---|---|
| Masked Generative Pretraining. | | | |
| Epoch | 800 | 800 | 800 |
| Warmup Epoch | 40 | 40 | 40 |
| Effective Batch Size | 4096 | 4096 | 4096 |
| Optimizer | AdamW(0.9,0.95) | AdamW(0.9,0.95) | AdamW(0.9,0.95) |
| Learning Rate | 1.5e-4 | 1.5e-4 | 1.5e-4 |
| LR schedule | cosine(min=0) | cosine(min=0) | cosine(min=0) |
| Weight Decay | 5e-2 | 5e-2 | 5e-2 |
| Mask Ratio | 0.75 | 0.75 | 0.75 |
| Input Size | 224 | 128 | 224 |
| Balanced Fine Tuning. | | | |
| Epoch | 100 | 100 | 30 |
| Warmup Epoch | 10 | 10 | 5 |
| Effective Batch Size | 1024 | 1024 | 1024 |
| Optimizer | AdamW(0.9,0.99) | AdamW(0.9,0.99) | AdamW(0.9,0.99) |
| Learning Rate | 1e-3 | 1e-3 | 1e-3 |
| LR schedule | cosine(min=1e-6) | cosine(min=1e-6) | cosine(min=1e-6) |
| Weight Decay | 5e-2 | 5e-2 | 5e-2 |
| Layer Decay | 0.75 | 0.75 | 0.75 |
| Input Size | 224 | 224 | 224 |
| Drop Path | 0.1 | 0.2 | 0.1 |
| $\tau$ of Bal-BCE | 1 | 1 | 1.05 |

## E. Additional Experiments

### E.1. DeiT with Bal-BCE

In the DeiT III [55], Touvron *et al.* propose to train ViTs with binary cross entropy loss. With our proposed bias $\mathcal{B}^{\text{bce}}$, we can further boost its recipe when collaborating with long-tailed distributed data. As Tab. 11 shows, Bal-BCE rebalances the performance of ViT-Small over three groups and improves the overall accuracy significantly. It is worth noticing that the few-shot gets ameliorated remarkably, while the many-shot is sacrificed to some extent. Compared to the results in Tab.6, we get a meticulous observation that Bal-BCE improves all groups' performance when adopting MGP as the pretrain manner, and even the many-shot (head) classes get compelling growth, especially on the small models. The aforementioned phenomenon may indicate that the MGP learns more generalized and unbiased features compared to supervised manners, which helps $\mathcal{B}^{\text{bce}}$ to calibrate more misclassification cases instead of the over-confident but right cases.

Table 11. Ablation study of proposed bias on DeiT III. Experiments are conducted with ViT-Small on ImageNet-LT for 400 epochs.

| Loss Type | Many | Δ | Med. | Δ | Few | Δ | Acc | Δ |
|---|---|---|---|---|---|---|---|---|
| BCE w/o $\mathcal{B}^{\text{bce}}$ | 64.2 | - | 32.2 | - | 9.0 | - | 41.4 | - |
| BCE w/ $\mathcal{B}^{\text{bce}}$ | 60.3 | **-4.0** | 40.8 | **+8.7** | 23.8 | **+14.7** | 46.0 | **+4.6** |

## E.2. Performance with higher resolution

With the FixRes effect [56], LiVT can reach further performance gains with minor computational overhead, which only increases the resolution in the 2nd stage with a few epochs. As a comparison, ResNet-based methods require extra effort to modify the network with heavy computational overhead. Hence, we only provide LiVT* in Tab. 2-4. Note that LiVT with 224 resolution already achieves SOTA performance (except tiny Places-LT). We additionally show ViT-based methods with 384 resolution in Tab. 12. ViT-based methods typically show lower performance than ResNet-based ones due to ViTs' data hungry in the tiny dataset (Tab. 4). Noteworthy, our Bal-BCE loss remarkably improves performance (Acc **+10.5**% & Few **+18.8**% compared to MAE). While tuning hyper-parameters (e.g., $\tau$ in Alg. 1 or parameters in Tab. 9&10) can further boost the performance (Fig. 3), we keep consistent settings with Tab. 2&3 to report the LiVT performance in Tab. 4.

Table 12. Top-1 Accuracy of ViT-B-16 pretrained on iNat18 dataset. We fine-tune models for 100 epochs with 384 resolution.

| Resolution | ViT | DeiT | MAE | LiVT |
|---|---|---|---|---|
| 224 × 224 | 54.6 | 61.0 | 69.4 | 76.1 |
| 384 × 384 | 56.3 **+1.7** | 63.7 **+2.7** | 72.9 **+3.5** | 81.0 **+4.9** |

## E.3. Negative-Tolerant Regularization

Recently, there are some other works to improve the performance of BCE loss. For instance, Wu *et al.* [66] propose to leave more Negative Tolerant Regularization (NTR) in the BCE loss. In long-tailed recognition, the tail class samples are usually learned as negative pairs resulting from the head class dominance. Here, for clear and concise expression, we call the logit $z_{\mathbf{y}_i}$ positive logit and $z_{\mathbf{y}_j}, (j \neq i)$ negative logits for the label $\mathbf{y}_i$. For *Softmax* operation, the gradient of the negative logits will be relatively small due to its mutual exclusion when the positive logit is large. However, *Sigmoid* acts differently from *Softmax*. The *Sigmoid* always maintains relatively large gradients for negative logits despite the positive logit value. This property of BCE leads to the output tail class logits being smaller, which incurs that the model only overfits a few tail-positive samples in the training set.

To overcome this problem, Wu *et al.* propose the NT-BCE loss to alleviate the dominance of negative labels. With a hyper-parameter $\lambda$ to control the strength of negative tolerance regularization, the NT-BCE can be written as:

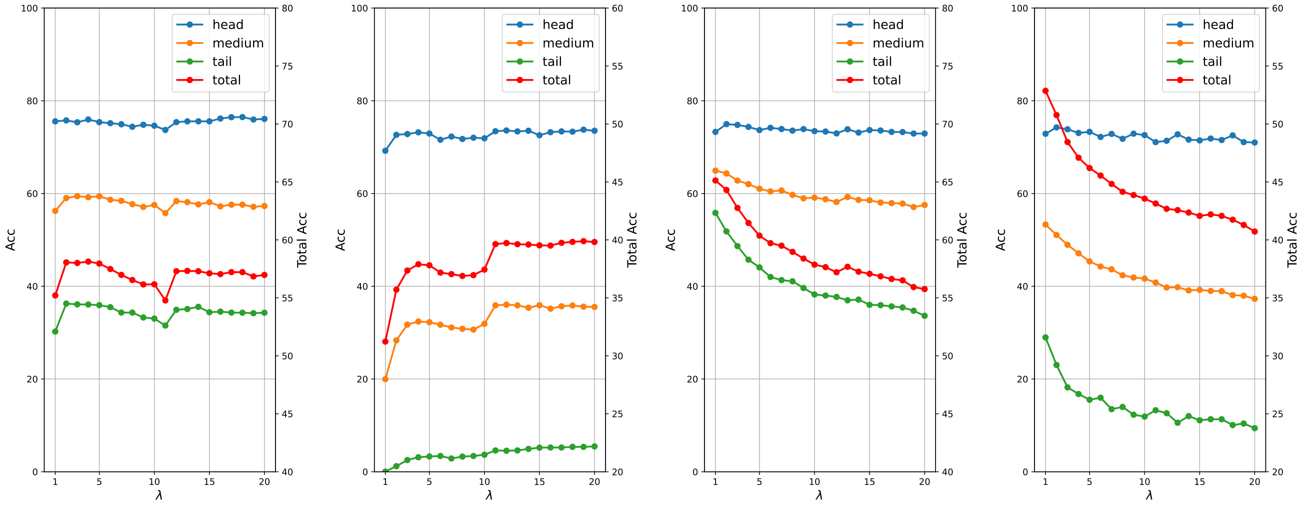
$$\mathcal{L}_{\text{NT-BCE}} = -\sum_{\mathbf{y}_i \in \mathcal{C}} [\mathbb{1}(\mathbf{y}_i) \cdot \log \frac{1}{1+e^{-\mathbf{z}_{\mathbf{y}_i}}} + \frac{1}{\lambda}(1-\mathbb{1}(\mathbf{y}_i)) \cdot \log(1-\frac{1}{1+e^{-\lambda \mathbf{z}_{\mathbf{y}_i}}})]$$

To collaborate with it, we add our proposed bias $\mathcal{B}^{\text{bce}}_{\mathbf{y}_i} = \log \pi_{\mathbf{y}_i} - \log(1-\pi_{\mathbf{y}_i})$ to the above loss and derive that:

$$\mathcal{L}^*_{\text{NT-BCE}} = -\sum_{\mathbf{y}_i \in \mathcal{C}} [\mathbb{1}(\mathbf{y}_i) \cdot \log \frac{1}{1+e^{-(\mathbf{z}_{\mathbf{y}_i}+\mathcal{B}^{\text{bce}}_{\mathbf{y}_i})}} + \frac{1}{\lambda}(1-\mathbb{1}(\mathbf{y}_i)) \cdot \log(1-\frac{1}{1+e^{-\lambda(\mathbf{z}_{\mathbf{y}_i}+\mathcal{B}^{\text{bce}}_{\mathbf{y}_i})}})]$$

For more in-depth observations, we train ViT-B on CIFAT-100-LT with both $\mathcal{L}_{\text{NT-BCE}}$ and $\mathcal{L}^*_{\text{NT-BCE}}$ and show the experiment results in Fig. 6. The NTR ameliorates the vanilla BCE loss with large $\lambda$ by benefiting medium and tail classes. However, the performance of $\mathcal{L}_{\text{NT-BCE}}$ is hard to catch up with $\mathcal{L}^*_{\text{NT-BCE}}$. What's worse, the NTR consistently deteriorates the performance of $\mathcal{L}^*_{\text{NT-BCE}}$ when $\lambda$ gets larger. The best is achieved at $\lambda = 1$, which indicates that NTR can not work well with our bias.

To explain it, we revisit the purpose of NTR, which aims to reduce the gradient of tail negative logits. While optimizing the tail class as negative logits, if the logit is small, the corresponding gradient will also be small to keep the logit from over-minimization. However, it is contradictory to our proposed bias. Typically, the margin-based loss makes the network pay attention to certain categories by increasing the corresponding difficulty with larger margins. As the margins for all classes, our bias $\mathcal{B}^{\text{bce}}$ makes the tail (head) class harder (easier) to learn, where the initial head logits are larger than tail ones, as shown in Fig.5. With NTR, tail classes will converge more slowly because larger $\lambda$ tends to slow down the optimization of tail logits, which finally results in unsatisfying tail performance. Although We *et al.*, add a similar bias in [66], they ignore its effect because of the little difference between the training and test label distribution of their datasets. More explorations are still required to make NTR and $\mathcal{B}^{\text{bce}}$ complement each other in long-tailed recognition.



(a) LT-10 with $\mathcal{L}_{\text{NT-BCE}}$ (b) LT-100 with $\mathcal{L}_{\text{NT-BCE}}$ (c) LT-10 with $\mathcal{L}^{*}_{\text{NT-BCE}}$ (d) LT-100 with $\mathcal{L}^{*}_{\text{NT-BCE}}$

Figure 6. Performance of BCE loss with NTR and $\mathcal{B}^{\text{bce}}$ on CIFAR100-LT. The total accuracy (red) is shown in right $y$-axis for better visualizations. (a)(b) NTR boosts vanilla BCE loss by benefiting medium and tail classes. (c)(d) NTR fails to collaborate with our bias.

# F. More Discussions

## F.1. About the two-stage pipeline.

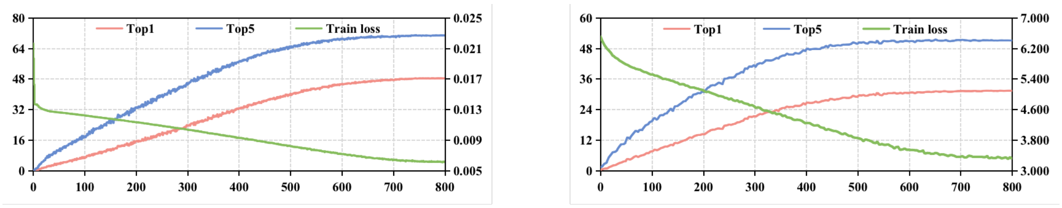
Although one stage is a promising direction, the two-stage frameworks (e.g., c-RT [29], MiSLAS [80], and our LiVT) typically achieve much better performance. For ViTs in LTR tasks, the difficulty is to learn the inductive bias and label statistical bias simultaneously. We manage the challenge by decoupling the two biases and learning the inductive bias in the MGP stage and the statistical bias in the BFT stage separately.

## F.2. Test prior to model performance.

The Bal-CE implementation in previous work [51] contains the test prior (i.e., $\pi_i^t = 1/C$) by default. With balanced test data, it is equal to eliminate the test prior bias item $-\log \pi_i^t = \log C$ with *Softmax* operation (Eq. 2). However, for Bal-BCE, the test prior term cannot be reduced in *Sigmoid* operation. As we discussed in Thm. 3, although ignoring this term does not influence the optimization direction, it will reduce the loss value, especially when $C$ is large (c.f. derivation in Supp. A.2). Therefore, the test prior is essential to ensure stability during training in Bal-BCE (e.g., models trained in the iNat18 dataset cannot converge without this item).

## F.3. About the baseline performance.

One may consider overfitting as a possible reason for the poor performance of the ViT baseline. Hence, we visualize the training log in Fig. 7. Either in the tiny Places-LT or the large-scale iNat18 (similar scale to ImageNet-1K), ViTs exhibit biased performance (Tab. 2-4). The unsatisfactory performance of ViT-based baselines (direct supervision) mainly accounts for the long-tailed problems rather than the overfitting issues (Tab. 1). Even under the same setting with these baselines, Bal-BCE improves MAE (Tab. 2-4) and DeiT (Supp. E.1) consistently in few and overall performance.



(a) Vanilla ViT (CE loss) (b) DeiT III (BCE loss)

Figure 7. The training log of ViT-B on ImageNet-LT. Left y-axis: validation accuracy. Right y-axis: training loss value. The validation accuracy is consistent with training loss, which means that overfitting does not occur during the training process.

## G. Visualization of MGP Reconstruction

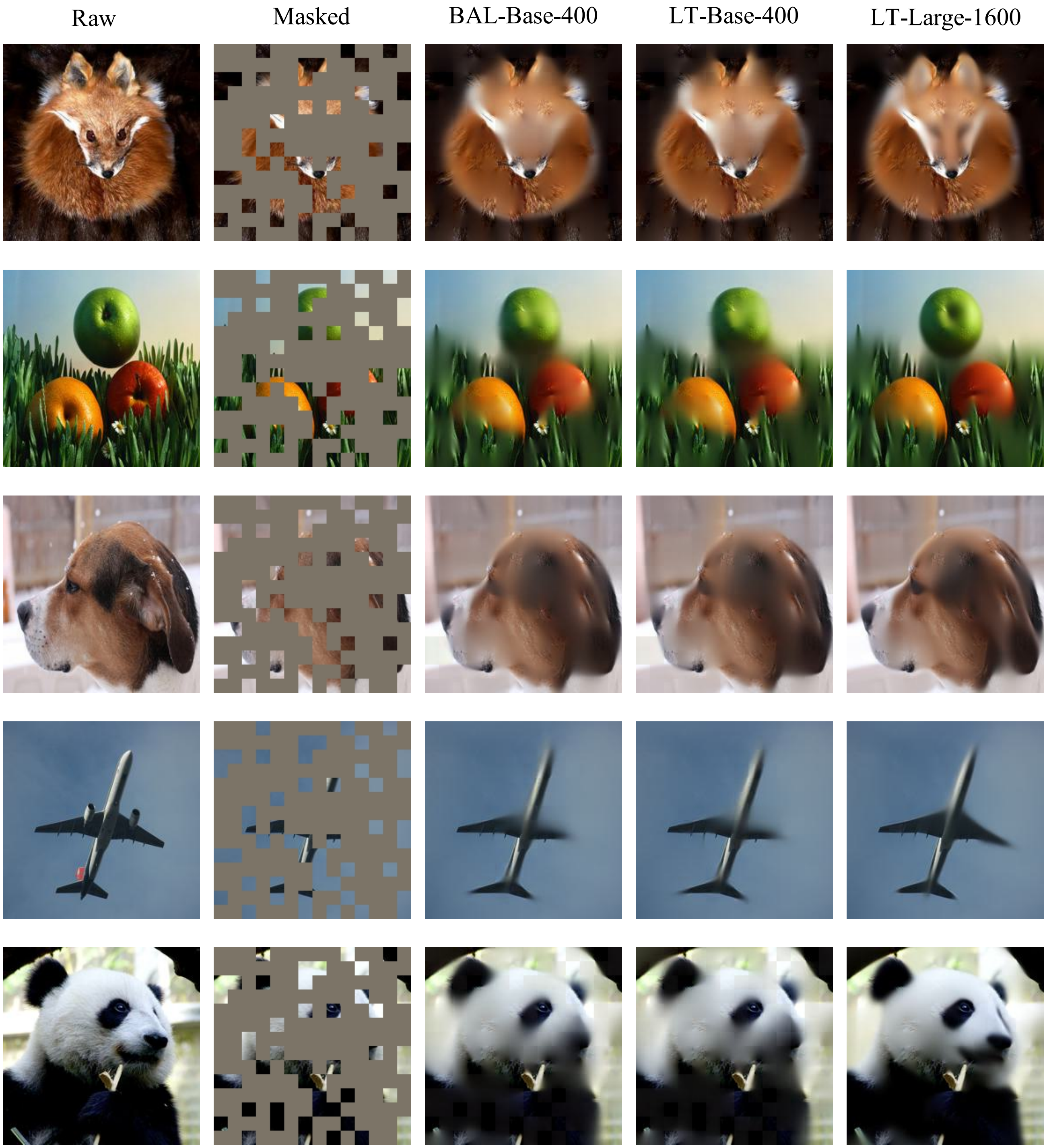


Figure 8. MGP Reconstruction comparisons. Raw: input images. Masked: we fix all masks for intuitive comparisons. BAL-Base-400: ViT-Base-16 trained on ImageNet-BAL for 400 epochs. LT-Base-400: ViT-Base-16 trained on ImageNet-LT for 400 epochs. LT-Large-1600: ViT-Large-16 trained on ImageNet-LT for 1600 epochs. With the same training instance number and implementation settings, the ViT-B models trained with both LT and BAL datasets show comparable reconstruction ability. With the ImageNet-LT data, we can further get better reconstruction results with a bigger model and longer MGP epochs, as the column LT-Large-1600 shows.