A APPENDIX

A.1 IMPLEMENTATION DETAILS

For all CIFAR-10 and CIFAR-100 comparison experiments, we used an 18-layer PreActResNet (He et al., 2016) as the baseline network following the setups in (Li et al., 2020), unless otherwise specified. The model was trained using SGD with a momentum of 0.9, a weight decay of 0.0005, and a batch size of 256 for CIFAR-100 and 512 for CIFAR-10. The network was trained from scratch for 300 epochs. We set the learning rate as 0.15 initially with a cosine annealing decay. Following (Li et al., 2020), we set the warm up period as 10 epochs for both CIFAR-10 & CIFAR-100. The optimizer and the learning rate schedule remained the same for both the main and the meta model. Gradient clipping is applied to stabilize training. All experiments were conducted with one V100 GPU, except for the experiments on Clothing 1M which were conducted with one RTX A6000 GPU.

For ISIC2019 experiments, we used ResNet-50 with ImageNet pretrained weights. A batch size of 64 was used for training with an initial learning rate of 0.01. The network was trained for 30 epochs in total with the warmup period as 1 epoch. All other implementation details remained the same as above. For Clothing 1M experiments, we used an ImageNet pre-trained 18-layer ResNet (He et al., 2016) as our baseline. We finetuned the network with a learning rate of 0.005 for 300 epochs. The model was trained using SGD with a momentum of 0.9, a weight decay of 0.0005, and a batch size of 256. Following (Li et al., 2020), to ensure the labels (noisy) were balanced, for each epoch, we sampled 250 mini-batches from the training data.

A.2 ALLEVIATE POTENTIAL OVERFITTING TO NOISY EXAMPLES.

We also plot the testing accuracy curve under different noise fractions in Figure 3, which shows that our proposed L2B would help preventing potential overfitting to noisy samples compared with standard training. Meanwhile, compared to simply sample reweighting (L2RW), our L2B introduces pseudo-labels for bootstrapping the learner and is able to converge to a better optimum.

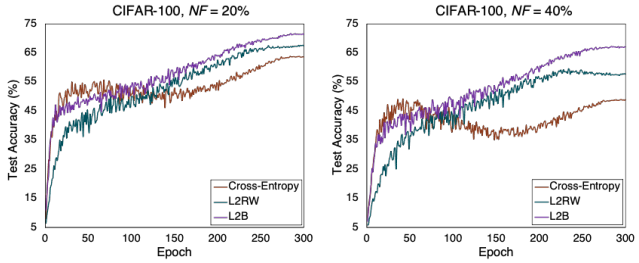

Figure 3: Test accuracy v.s. number of epochs on CIFAR-100 under the noise fraction of 20% and 40%.

A.3 QUALITATIVE RESULTS

We also demonstrate a set of qualitative examples to illustrate how our proposed L2B benefits from the joint instance and label reweighting paradigm. In Figure 4, we can see that when the estimated pseudo label is of high-quality, i.e., the pseudo label is different from the noisy label but equal to the clean label, our model will automatically assign a much higher weight to β for corrupted training samples. On the contrary, α can be near zero in this case. This indicates that our L2B algorithm will pay more attention to the pseudo label than the real noisy label when computing the losses. In addition, we also show several cases where the pseudo label is equal to the noisy label, where we can see that α and β are almost identical under this circumstance since the two losses are of the same value. Note that the relatively small values of α and β are due to that we use a large batch size (i.e., 512) for CIFAR-10 experiments. By normalizing the weights in each training batch (see Eq. 9), the value of α and β can be on the scale of 10^{-4}.

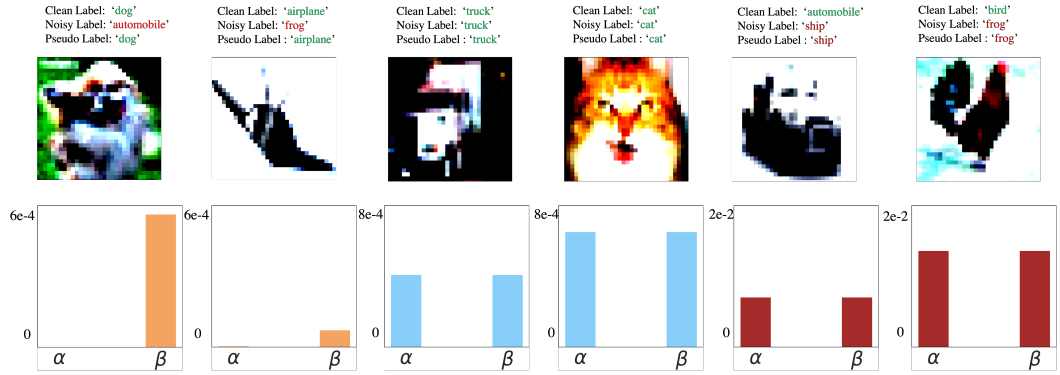

Figure 4: Examples of α and β on CIFAR-10 with asymmetric noise fraction of 20%. When the estimated pseudo label is of high-quality, i.e., the pseudo label is different from the noisy label but equal to the clean label, our model will automatically assign a much higher weight to β than to α for corrupted training samples. When the pseudo label is equal to the noisy label (i.e., the two loss terms are equal to each other), α and β are almost identical.

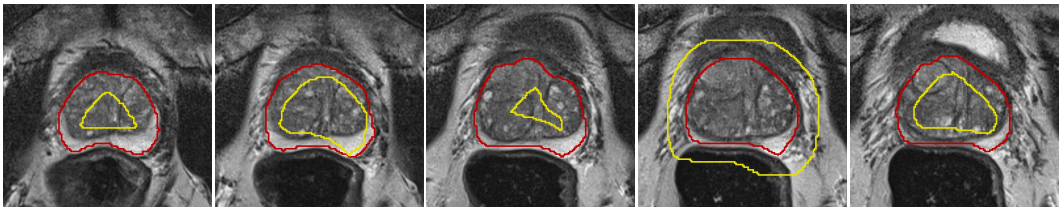

Figure 5: Visual comparison of prostate MRI images with noisy (contoured in yellow) and accurate (contoured in red) segmentation masks to demonstrate the discrepancy in segmentation quality between the two.

A.4 Details of the generalized segmentation task

PROMISE12 dataset contains 50 3D transversal T2-weighted MR images of the prostate with manual binary prostate gland segmentation and is obtained from multiple centers with different acquisition protocols. Following Soerensen et al. (2021); Wang et al. (2021), we utilized 2D slices in the axial view for both training and testing. All images are resized to 144×144 and splits are randomized. Noisy labels used in Table 5 were synthesized using random rotation, erosion, or dilation, achieving approximately a 60% corruption ratio and an average Dice coefficient of 0.6206. And visualizations of the corrupted noisy labels (shown in yellow) as well as the ground-truth (shown in red) are illustrated in Figure 5. Furthermore, we also investigate the robustness of our method by varying the noise level of the corrupted training set from $\{\mathrm{L}_1, \mathrm{L}_2, \mathrm{L}_3\}$, where the average Dice coefficients are $\mathrm{Dice}_{\mathrm{L}_1} = 0.4148$, $\mathrm{Dice}_{\mathrm{L}_2} = 0.6206$, and $\mathrm{Dice}_{\mathrm{L}_3} = 0.8031$ (*i.e.*, the corrupted ratios are around 60% (L_1), 40% (L_2), and 20% (L_3)). At each noise level, we compare the baseline UNet++ which is directly trained on the noisy training data with our MLB-Seg. As shown in Table 8, we report the averaged dice coefficient over 5 repetitions for each series of experiments. The standard deviation for all experiments is within 0.5%. We could notice that while the noise level increases, performances of baseline drop from 80.03% to 59.77%, but performances of MLB-Seg only drop

Table 8: Ablation study on different noise levels

Method	Dice (%)↑
baseline - L_1	59.77
MLB-Seg - L_1	77.70
baseline - L_2	73.74
MLB-Seg - L_2	80.83
baseline - L_3	80.03
MLB-Seg - L_3	82.01

from 82.01% to 77.70% which indicates that our MLB-Seg is robust to different noisy levels and shows larger improvements under a much severer noisy situation.

B THEORETICAL ANALYSIS

B.1 EQUIVALENCE OF THE TWO LEARNING OBJECTIVES

We show that Eq. 3 is equivalent with Eq. 2 when $\forall i\ \alpha_i + \beta_i = 1$. For convenience, we denote $y_i^{\text{real}}, y_i^{\text{pseudo}}, \mathcal{F}(x_i, \theta)$ using y_i^r, y_i^p, p_i respectively.

$$\alpha_i \mathcal{L}(p_i, y_i^r) + \beta_i \mathcal{L}(p_i, y_i^p) = \sum_{l=1}^{L} \alpha_i y_{i,l}^r \log p_{i,l} \tag{15}$$

$$+ \beta_i y_{i,l}^p \log p_{i,l} = \sum_{l=1}^{L} (\alpha_i y_{i,l}^r + \beta_i y_{i,l}^p) \log p_{i,l} \tag{16}$$

Due to that $\mathcal{L}(\cdot)$ is the cross-entropy loss, we have $\sum_{l=1}^{L} y_{i,l}^r = \sum_{l=1}^{L} y_{i,l}^p = 1$. Then $\sum_{l=1}^{L} \alpha_i y_{i,l}^r + \beta_i y_{i,l}^p = \alpha_i + \beta_i$. So if $\alpha_i + \beta_i = 1$, we have

$$\sum_{l=1}^{L} (\alpha_i y_{i,l}^r + \beta_i y_{i,l}^p) \log p_{i,l} = \mathcal{L}(p_i, \alpha_i y_i^r + \beta_i y_i^p) \tag{17}$$

$$= \mathcal{L}(p_i, (1 - \beta_i) y_i^r + \beta_i y_i^p) \tag{18}$$

B.2 GRADIENT USED FOR UPDATING θ

We derivative the update rule for $\boldsymbol{\alpha}, \boldsymbol{\beta}$ in Eq. 10.

$$\alpha_{t,i} = -\eta \frac{\partial}{\partial \alpha_i} (\sum_{j=1}^{m} f_j^v(\hat{\theta}_{t+1})) \Big|_{\alpha_i = 0} \tag{19}$$

$$= -\eta \sum_{j=1}^{m} \nabla f_j^v(\hat{\theta}_{t+1})^T \frac{\partial \hat{\theta}_{t+1}}{\partial \alpha_i} \Big|_{\alpha_i = 0} \tag{20}$$

$$= -\eta \sum_{j=1}^{m} \nabla f_j^v(\hat{\theta}_{t+1})^T \tag{21}$$

$$\frac{\partial (\theta_t - \lambda \nabla (\sum_k \alpha_k\ f_k(\theta) + \beta_k\ g_k(\theta)) \Big|_{\theta = \theta_t})}{\partial \alpha_i} \Big|_{\alpha_i = 0} \tag{22}$$

$$= \eta \lambda \sum_{j=1}^{m} \nabla f_j^v(\theta_t)^T \nabla f_i(\theta_t) \tag{23}$$

$$\beta_{t,i} = -\eta \frac{\partial}{\partial \beta_i} (\sum_{j=1}^{m} f_j^v(\hat{\theta}_{t+1}))\Big|_{\beta_i=0} \tag{24}$$

$$= -\eta \sum_{j=1}^{m} \nabla f_j^v(\hat{\theta}_{t+1})^T \frac{\partial \hat{\theta}_{t+1}}{\partial \beta_i}\Big|_{\beta_i=0} \tag{25}$$

$$= -\eta \sum_{j=1}^{m} \nabla f_j^v(\hat{\theta}_{t+1})^T \tag{26}$$

$$\frac{\partial(\theta_t - \lambda \nabla(\sum_k \alpha_k\, g_k(\theta) + \beta_k\, g_k(\theta))\Big|_{\theta=\theta_t})}{\partial \beta_i}\Big|_{\beta_i=0} \tag{27}$$

$$= \eta\lambda \sum_{j=1}^{m} \nabla f_j^v(\theta_t)^T \nabla g_i(\theta_t) \tag{28}$$

Then θ_{t+1} can be calculated by Eq. 10 using the updated $\alpha_{t,i}, \beta_{t,i}$.

B.3 Convergence

This section provides the proof for covergence (Theorem 1)

Theorem. *Suppose that the training loss function f, g have σ-bounded gradients and the validation loss f^v is Lipschitz smooth with constant L. With a small enough learning rate λ, the validation loss monotonically decreases for any training batch B, namely,*

$$G(\theta_{t+1}) \leq G(\theta_t), \tag{29}$$

where θ_{t+1} is obtained using Eq. 10 and G is the validation loss

$$G(\theta) = \frac{1}{M} \sum_{i=1}^{M} f_i^v(\theta), \tag{30}$$

Furthermore, Eq. 29 holds for all possible training batches only when the gradient of validation loss function becomes 0 at some step t, namely, $G(\theta_{t+1}) = G(\theta_t)\ \forall B \Leftrightarrow \nabla G(\theta_t) = 0$

Proof. At each training step t, we pick a mini-batch B from the union of training and validation data with $|B| = n$. From section B we can derivative θ_{t+1} as follows:

$$\theta_{t+1} = \theta_t - \lambda \sum_{i=1}^{n} (\alpha_{t,i} \nabla f_i(\theta_t) + \beta_{t,i} \nabla g_i(\theta_t)) \tag{31}$$

$$= \theta_t - \eta\lambda^2 M \sum_{i=1}^{n} (\nabla G^T \nabla f_i \nabla f_i + \nabla G^T \nabla g_i \nabla g_i) \tag{32}$$

We omit θ_t after every function for briefness and set m in section B equals to M. Since $G(\theta)$ is Lipschitz-smooth, we have

$$G(\theta_{t+1}) \leq G(\theta_t) + \nabla G^T \Delta\theta + \frac{L}{2} ||\Delta\theta||^2. \tag{33}$$

Then we show $\nabla G^T \Delta\theta + \frac{L}{2} ||\Delta\theta||^2 \leq 0$ with a small enough λ. Specifically,

$$\nabla G^T \Delta\theta = -\eta\lambda^2 M \sum_i (\nabla G^T \nabla f_i)^2 + (\nabla G^T \nabla g_i)^2. \tag{34}$$

Then since f_i, g_i have σ-bounded gradients, we have

$$\frac{L}{2}||\Delta\theta||^2 \leq \frac{L\eta^2\lambda^4M^2}{2}\sum_i(\nabla G^T\nabla f_i)^2||\nabla f_i||^2 \tag{35}$$

$$+ (\nabla G^T\nabla g_i)^2||\nabla g_i||^2 \tag{36}$$

$$\leq \frac{L\eta^2\lambda^4M^2\sigma^2}{2}\sum_i(\nabla G^T\nabla f_i)^2 + (\nabla G^T\nabla g_i)^2 \tag{37}$$

Then if $\lambda^2 < \frac{2}{\eta\sigma^2ML}$,

$$\nabla G^T\Delta\theta + \frac{L}{2}||\Delta\theta||^2 \leq (\frac{L\eta^2\lambda^4M^2\sigma^2}{2} - \eta\lambda^2M) \tag{38}$$

$$\sum_i(\nabla G^T\nabla f_i)^2 + (\nabla G^T\nabla g_i)^2 \leq 0. \tag{39}$$

Finally we prove $G(\theta_{t+1}) = G(\theta_t)\ \forall B \Leftrightarrow \nabla G(\theta_t) = 0$: If $\nabla G(\theta_t) = 0$, from section B we have $\alpha_{t,i} = \beta_{t,i} = 0$, then $\theta_{t+1} = \theta_t$ and thus $G(\theta_{t+1}) = G(\theta_t)\ \forall B$. Otherwise, if $\nabla G(\theta_t) \neq 0$, we have

$$0 < ||\nabla G||^2 = \nabla G^T\nabla G = \frac{1}{M}\sum_{i=1}^{M}\nabla G^T\nabla f_i^v, \tag{40}$$

which means there exists a k such that $\nabla G^T\nabla f_k^v > 0$. So for the mini-batch B_k that contains this example, we have

$$G(\theta_{t+1}) - G(\theta_t) \leq \nabla G^T\Delta\theta + \frac{L}{2}||\Delta\theta||^2 \tag{41}$$

$$\leq (\frac{L\eta^2\lambda^4M^2\sigma^2}{2} - \eta\lambda^2M) \tag{42}$$

$$\sum_{i\in B}(\nabla G^T\nabla f_i)^2 + (\nabla G^T\nabla g_i)^2 \tag{43}$$

$$\leq (\frac{L\eta^2\lambda^4M^2\sigma^2}{2} - \eta\lambda^2M)\nabla G^T\nabla f_k^v \tag{44}$$

$$< 0. \tag{45}$$