Supplementary Material

No Institute Given

The appendix includes the following sections:

1. **Simplified objective** (Appendix A): discusses the proposed objective parameterization.
2. **Computational efficiency** (Appendix B): discusses complexity and training speed of ADELLO.
3. **Confidence calibration** (Appendix C): provides additional definitions and further discussion about calibration performance.
4. **Beyond natural images** (Appendix D): presents experiments conducted on additional image domains, including medical and remote sensing datasets.
5. **Additional algorithmic details** (Appendix E): includes pseudo-code of the proposed algorithm.
6. **Additional training details** (Appendix F): includes hyperparameter configurations for each dataset.

A Simplified objective

In the main paper, we utilize equally weighted losses for ADELLO. Alternatively, a more complex formulation can be expressed as:

$$\mathcal{L} = \mathcal{L}_s^{\text{FlexDA}} + \lambda_u \mathcal{L}_u^{\text{FlexDA}} + \lambda_{uC} \mathcal{L}_{uC}^{\text{FlexDA}}\,, \tag{1}$$

where λ_u and λ_{uC} are loss weights assigned to standard consistency and complementary consistency losses within the FlexDA framework, respectively. For simplicity and following the accepted $\lambda_u = 1$ norm [6, 7, 9, 12], we also set $\lambda_{uC} = 1$. Table 10 supports this choice across several datasets, presenting steady performance around the default setting, with a decline noted for extreme values.

Table 10: Ablation of complementary consistency loss weight λ_{uC}. We report test accuracy using CIFAR100-LT50 and STL10-LT20 datasets.

λ_{uC}	0	0.001	0.01	0.1	0.5	**1**	2	10
CIFAR100-LT50	48.6±0.7	48.4±0.7	48.6±1.0	48.8±0.6	49.0±0.5	**49.2**±0.5	48.5±0.5	44.5±0.5
STL10-LT20	67.1±1.6	67.3±1.4	67.4±1.1	69.3±1.0	74.0±0.6	**74.6**±0.4	72.4±1.3	71.4±0.3

B Computational efficiency

ADELLO improves FixMatch by aligning pseudo-labels with the (unknown) class distribution of unlabeled data. This is achieved by tracking the exponential moving average of pseudo-labels, which is then used to adjust cross-entropy losses to correct for long-tailed biases. Additionally, it employs a masked distillation loss. Importantly, ADELLO accomplishes these enhancements without increased complexity. It does so by avoiding additional computational steps such as extra forward passes, the use of auxiliary classifiers, or the need for data re-sampling, thus maintaining a straightforward implementation. Training times show its efficiency: **ADELLO** at **5h18m** closely aligns with **FixMatch** at **5h15m** and ABC at 5h21m, and surpasses CReST+ at 6h22m, CoSSL at 7h29m, DARP at 7h43m, and **DASO** at **19h32m** for CIFAR100-LT50 on a single Nvidia V100-32GB GPU.

C Confidence calibration

Calibration definitions. At its core, model calibration evaluates how closely a model's predicted confidence aligns with the actual likelihood of correctness [1]. For example, if a model predicts a certain class with 95% confidence, in an ideal scenario, that prediction should be accurate 95% of the time. A practical calibration requirement is *argmax calibration* [4]. For a model P, outputting normalized probabilities, this criterion requires that for the class with the highest predicted confidence, denoted as $\hat{Y} = \arg\max P(X)$ with confidence $\hat{P}(X) = \max P(X)$, said confidence should match the actual probability of that class being correct, across all levels of confidence:

$$\mathbb{P}(\hat{Y} = Y | \hat{P}(X) = p) \stackrel{!}{=} p, \qquad \forall p \in [0, 1]. \tag{2}$$

In practice, we empirically evaluate the congruence between predicted confidence and actual accuracy over a test dataset $\mathcal{D}_{\text{test}} = \{x_i, y_i\}_{i=1}^{N_{\text{test}}}$. This involves grouping model predictions into M bins based on confidence levels and analyzing the accuracy and confidence within each bin. For a given bin B_m, its accuracy, $acc(B_m)$, is the proportion of correct predictions, and its confidence, $conf(B_m)$, is the average predicted confidence. The Expected Calibration Error (ECE) quantifies the overall discrepancy between accuracy and confidence across all bins:

$$ECE = \sum_{m=1}^{M} \frac{|B_m|}{n} |acc(B_m) - conf(B_m)|. \tag{3}$$

Similarly, the Maximum Calibration Error (MCE) identifies the largest such discrepancy, indicating the worst-case deviation between confidence and accuracy:

$$MCE = \max_{m \in \{1,..,M\}} |acc(B_m) - conf(B_m)|. \tag{4}$$

More results on model calibration. In Section 5.3, we show how our approach not only improves generalization capabilities but also significantly enhances model

calibration in various LTSSL contexts. Additionally, in Tables 11 and 12, we present the calibration performance of various models, focusing specifically on the expected calibration error and the maximum calibration error, respectively. Our approach is consistently the top performer for reducing ECE, as shown in Table 11. Analogously, ADELLO achieves the leading position for MCE reduction, as presented in Table 12. This aspect is especially crucial in mission-critical applications, where reducing the maximum errors in model predictions is imperative.

Table 11: Expected Calibration Error (ECE) across different datasets. Best scores **bold**, second-best underlined.

	CIFAR10-LT	STL10-LT	CIFAR100-LT				Friedman	Final
$\gamma_l \rightarrow$	100	20	20	50	50	50	Rank	Rank
$\gamma_u \rightarrow$	100	N/A	20	50	1	0.02		
$N_1 \rightarrow$	500	150	50	150	150	150		
$M_1 \rightarrow$	4000	N/A	400	300	300	6		
FixMatch [9]	23.9±1.9	37.8±4.5	39.9±0.4	37.4±0.4	34.6±0.5	37.7±0.8	9.0	9
+DARP [6]	19.2±1.2	31.6±2.9	32.2±0.5	33.3±0.1	33.1±0.7	35.7±0.7	6.8	8
+CReST+ [12]	15.4±0.3	30.0±2.7	34.2±0.7	31.1±0.1	29.0±0.7	31.9±0.8	5.1	5
+ABC [8]	13.5±1.0	24.6±2.3	31.6±0.2	**24.5**±0.5	22.8±0.7	27.2±1.5	2.7	2
+DebiasPL [10]	17.0±4.2	24.2±1.1	35.1±1.1	33.9±0.3	30.4±0.7	31.3±1.3	5.8	7
+CoSSL [3]	12.1±0.5	22.7±1.3	34.6±0.5	31.2±0.3	29.7±0.9	34.4±0.7	4.8	4
+UDAL [7]	12.9±0.4	25.7±2.3	33.5±0.3	31.1±0.1	29.0±0.7	31.9±0.8	4.4	3
+ADELLO (ours)	**10.4**±0.3	**6.9**±0.3	**28.8**±0.3	26.1±0.9	**21.0**±0.9	**26.2**±0.5	**1.2**	**1**
SoftMatch [2]	15.7±0.8	20.0±0.5	36.7±0.3	34.2±0.5	26.2±0.5	31.4±0.7	5.2	6

D Beyond natural images

Following the CIFAR10-LT protocol, we constructed long-tailed versions of TissueMNIST [13], with 28×28 greyscale **microscopy medical images** across 8 classes, and EuroSAT [5], featuring 32×32 RGB **satellite images** in 10 classes. We use 1/3 of labeled data and all hyper-parameters are set following CIFAR10-LT experiments. Tab. 13 shows that our approach can effectively tackle class imbalance and label shift across various image domains.

E Additional algorithmic details

In Algorithm 1, we provide pseudo-code for ADELLO, utilizing FixMatch as the base SSL algorithm.

F Additional training details

In Table 14, we provide a comprehensive list of the hyperparameter settings utilized for each dataset. For supervised baselines, the base learning rate starts at 0.1 with

Table 12: Maximum Calibration Error (MCE) across different datasets. Best scores **bold**, second-best underlined.

	CIFAR10-LT	STL10-LT	CIFAR100-LT				Friedman	Final
$\gamma_l \rightarrow$	100	20	20	50	50	50	Rank	Rank
$\gamma_u \rightarrow$	100	N/A	20	50	1	0.02		
$N_1 \rightarrow$	500	150	50	150	150	150		
$M_1 \rightarrow$	4000	N/A	400	300	300	6		
FixMatch [9]	47.5±5.0	55.1±4.9	61.3±1.8	57.3±1.1	55.3±0.8	55.5±2.4	9.0	9
+DARP [6]	46.1±5.4	52.4±5.1	58.0±1.6	53.0±1.7	50.9±1.1	55.1±0.9	7.5	8
+CReST+ [12]	37.7±5.8	48.9±5.6	**51.0**±1.9	51.6±0.8	49.3±1.6	49.8±1.2	3.2	2
+ABC [8]	42.1±4.6	49.2±5.1	56.4±1.2	**41.4**±1.0	40.9±0.4	43.1±2.1	3.5	3
+DebiasPL [10]	40.0±8.8	45.2±5.1	56.0±1.6	53.7±1.1	50.4±1.1	51.9±0.8	5.2	6
+CoSSL [3]	42.6±4.6	50.7±4.9	58.8±1.6	50.9±0.5	49.5±0.3	51.7±1.6	6.2	7
+UDAL [7]	41.0±5.8	50.1±5.1	56.5±1.3	51.6±0.8	49.3±1.6	49.8±1.2	4.9	5
+ADELLO (ours)	39.5±6.4	**25.9**±1.0	52.8±1.6	46.2±0.6	**37.9**±1.6	**42.0**±0.5	**1.7**	**1**
SoftMatch [2]	**36.8**±5.1	41.7±6.0	56.2±2.2	53.9±1.1	45.5±2.3	50.8±1.1	3.8	4

Table 13: Test balanced accuracy (%) on TissueMNIST-LT and EuroSAT-LT. Comparison of single-classifier approaches.

	TissueMNIST-LT			EuroSAT-LT
$\gamma_l = 100$ / $\gamma_u \rightarrow$	100	≈ 1	0.01	100
FixMatch [9]	44.6±0.2	45.0±0.2	44.7±0.3	89.9±0.6
+DARP [6]	44.5±0.2	44.5±0.1	43.9±0.5	90.2±0.8
+DebiasPL [10]	45.2±0.5	46.0±0.2	45.6±0.1	91.8±0.4
+UDAL [7]	50.9±0.3	51.5±0.3	51.4±0.1	93.5±0.3
+ADELLO (ours)	**52.3**±0.3	**54.3**±0.3	**54.4**±0.3	**94.1**±0.7

Algorithm 1 ADELLO with FixMatch as SSL algorithm

Input: Labeled dataset D_L=(X_L, Y_L), Unlabeled dataset D_U=(X_U, $\cdot$), Model f
Parameters: Batch size B, Batch-ratio μ, Number of classes K, Max iterations t_{total}, Confidence threshold τ, Min debiasing factor α_{min}, Schedule speed factor d, EMA momentum β, Warmup iterations t_{warmup}
▷ σ for softmax, ω and Ω for weak and strong data augmentation functions
Initialize: $P_{\text{bal}} \leftarrow (\frac{1}{K}, ..., \frac{1}{K})$, $\hat{Q} \leftarrow P_{\text{bal}}$, $T \leftarrow 1$
for $t = 1$ to t_{total} **do** ▷ Main training loop
  $\alpha_t \leftarrow 1.0 - (1.0 - \alpha_{\text{min}}) \cdot \left(\frac{t}{t_{\text{total}}}\right)^d$ ▷ Update FlexDA target prior
  $\hat{\mathcal{Q}}_{\alpha_t} \leftarrow \text{normalize}(\hat{\mathcal{Q}}^{\alpha_t})$
  if $t = t_{\text{warmup}}$ **then**
    $T \leftarrow \text{KL}(P_{\text{bal}}||\hat{Q})$ ▷ Infer temperature T after warmup
  end if
  Sample mini-batches B_L from D_L and B_U from D_U
  $\mathcal{M}(B_u) = \mathbf{1}[\max(\sigma(f(\omega(B_u))), \text{axis} = -1) \geq \tau]$ ▷ High-confidence mask
  $\mathcal{M}^C(B_u) = 1 - \mathcal{M}(B_u)$ ▷ Complement mask
  $\hat{y} = \text{argmax}(\sigma(f(\omega(B_u))), \text{axis} = -1)$ ▷ Predict Hard PLs
  $\tilde{y} = \sigma(\frac{1}{T} f(\omega(B_u)))$ ▷ Predict Soft PLs
  $\mathcal{L}_s^{\text{FlexDA}} = \frac{1}{B} \sum_{b=1}^{B} \mathcal{H}(y_b, \sigma(f(\omega(x_b)) + \log \frac{\mathcal{P}_L}{\hat{\mathcal{Q}}_{\alpha_t}}))$ ▷ Supervised loss
  $\mathcal{L}_u^{\text{FlexDA}} = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathcal{M}(u_b) \cdot \mathcal{H}(\hat{y}_b, \sigma(f(\Omega(u_b)) + \log \frac{\hat{Q}}{\hat{\mathcal{Q}}_{\alpha_t}}))$ ▷ Consistency loss
  $\mathcal{L}_{uC}^{\text{FlexDA}} = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathcal{M}^C(u_b) \cdot \mathcal{H}(\tilde{y}_b, \sigma(\frac{1}{T}(f(\Omega(u_b)) + \log \frac{\hat{Q}}{\hat{\mathcal{Q}}_{\alpha_t}})))$ ▷ CCR loss
  $\mathcal{L} = \mathcal{L}_s^{\text{FlexDA}} + \mathcal{L}_u^{\text{FlexDA}} + \mathbf{1}[t \geq t_{\text{warmup}}] \cdot \mathcal{L}_{uC}^{\text{FlexDA}}$ ▷ ADELLO objective
  Update f to minimize $\mathcal{L}$
  $\hat{Q} \leftarrow \beta \cdot \hat{Q} + (1 - \beta) \cdot \text{mean}(\sigma(f(\omega(B_U))), \text{axis} = 0)$ ▷ Update $\hat{Q}$ w/EMA of PLs
end for
Output: Model f

a linear warmup. Unless stated otherwise, we reproduce all methods using unified codebases based on [11][1] for CIFAR10, CIFAR100, and STL10, and based on [3][2] for ImageNet127.

Table 14: Hyperparameter settings for different datasets.

Hyperparameter	CIFAR10-LT	CIFAR100-LT	STL10-LT	ImageNet127
Backbone	Wide-ResNet-28-2	Wide-ResNet-28-2	Wide-ResNet-28-2	ResNet-50
Base SSL algorithm	FixMatch	FixMatch	FixMatch	FixMatch
Confidence Threshold	0.95	0.95	0.95	0.95
Optimizer	SGD+Nesterov	SGD+Nesterov	SGD+Nesterov	Adam
Nesterov Momentum	0.9	0.9	0.9	-
Weight Decay	5e-4	5e-4	5e-4	-
Base Learning Rate	0.03	0.03	0.03	0.002
Epochs	256	256	256	500
Steps per Epoch	1024	1024	1024	500
Batch Size (labeled)	64	64	64	64
Batch Size (unlabeled)	128	128	128	64x2 views
FlexDA α_{min}	0.1	0.1	0.1	0.1
FlexDA d	2	2	2	2
FlexDA EMA β	0.999	0.999	0.999	0.999
Temperature T	inferred	inferred	inferred	inferred
Warm-up t_{warmup}	50k	50k	0	0
λ_u	1	1	1	1
λ_{uC}	1	1	1	1

[1] https://github.com/microsoft/Semi-supervised-learning (MIT license)
[2] https://github.com/YUE-FAN/CoSSL (MIT license)

References

1. Brocker, J.: Reliability, sufficiency, and the decomposition of proper scores (2008), https://api.semanticscholar.org/CorpusID:15880012 2
2. Chen, H., Tao, R., Fan, Y., Wang, Y., Savvides, M., Wang, J., Raj, B., Xie, X., Schiele, B.: Softmatch: Addressing the quantity-quality tradeoff in semi-supervised learning. In: Eleventh International Conference on Learning Representations. OpenReview. net (2023) 3, 4
3. Fan, Y., Dai, D., Schiele, B.: Cossl: Co-learning of representation and classifier for imbalanced semi-supervised learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022) 3, 4, 6
4. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: International conference on machine learning. pp. 1321–1330. PMLR (2017) 2
5. Helber, P., Bischke, B., Dengel, A., Borth, D.: Introducing eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. In: IEEE IGARSS (2018) 3
6. Kim, J., Hur, Y., Park, S., Yang, E., Hwang, S., Shin, J.: Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning. In: Advances in neural information processing systems (2020) 1, 3, 4
7. Lazarow, J., Sohn, K., Lee, C.Y., Li, C.L., Zhang, Z., Pfister, T.: Unifying distribution alignment as a loss for imbalanced semi-supervised learning. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 5644–5653 (2023) 1, 3, 4
8. Lee, H., Shin, S., Kim, H.: Abc: Auxiliary balanced classifier for class-imbalanced semi-supervised learning. Advances in Neural Information Processing Systems **34**, 7082–7094 (2021) 3, 4
9. Sohn, K., Berthelot, D., Li, C.L., Zhang, Z., Carlini, N., Cubuk, E.D., Kurakin, A., Zhang, H., Raffel, C.: Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In: Advances in Neural Information Processing Systems (2020) 1, 3, 4
10. Wang, X., Wu, Z., Lian, L., Yu, S.X.: Debiased learning from naturally imbalanced pseudo-labels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14647–14657 (2022) 3, 4
11. Wang, Y., Chen, H., Fan, Y., Sun, W., Tao, R., Hou, W., Wang, R., Yang, L., Zhou, Z., Guo, L.Z., Qi, H., Wu, Z., Li, Y.F., Nakamura, S., Ye, W., Savvides, M., Raj, B., Shinozaki, T., Schiele, B., Wang, J., Xie, X., Zhang, Y.: Usb: A unified semi-supervised learning benchmark for classification (2022) 6
12. Wei, C., Sohn, K., Mellina, C., Yuille, A., Yang, F.: Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021) 1, 3, 4
13. Yang, J., Shi, R., Ni, B.: Medmnist classification decathlon: A lightweight automl benchmark for medical image analysis. In: IEEE ISBI (2021) 3