A. Gaussian Mixture Models

Having a set of LLC scores $\{\mathrm{LLC}_i\}_{i=0}^{N}$, we assume $\mathrm{p}(\mathrm{LLC}) = \sum_{k=1}^{2} \pi_k \mathcal{N}(\mathrm{LLC} \mid \mu_k, \sigma_k)$, where $\mathcal{N}$ is a normal distribution with mean π and variance σ and π is the weight of each Gaussian component. As a latent variable model (latent variables π), one can find the parameters of the probability distribution function using the Expectation maximization (EM) algorithm by maximizing maximum likelihood $\ln \mathrm{p}(\{\mathrm{LLC}\}_{i=0}^{N} \mid \pi, \mu, \sigma) = \sum_{i=1}^{N} \ln \sum_{k=1}^{2} \pi_k \mathcal{N}(\mathrm{LLC}_i \mid \mu_k, \sigma_k)$. After fitting, one can obtain the clean probability of i-th sample by $\frac{\pi_1 \mathcal{N}(\mathrm{LLC}_i \mid \mu_1, \sigma_1)}{\mathrm{p}(\mathrm{LLC}_i \mid \pi, \mu, \sigma)}$ (assuming $\mu_1 > \mu_2$).

B. Comprasion on ANIMAIL-10N

For comparison on ANIMAL-10N, the results of SELFIE [38], PLC [53], NCT [6] are reported in Table 5.

SELFIE	PLC	NCT	Ours
81.8	83.4	84.1	**88.2**

Table 5. Comparison with other methods on ANIMAl-10N. The results of other methods are from [6].

C. More imbalance ratio

Waterbirds is a synthetic dataset, which allows us to control the imbalance ratio. Still, with 4,795 training examples consisting of waterbird and landbird images, 90% of waterbirds have water backgrounds, and the remaining 10% of waterbirds have land backgrounds (instead of the commonly used 95% vs. 5%). Similarly, 90% of landbirds are placed against land backgrounds, and the remaining 10% of landbirds are placed against water backgrounds. We report more experiments on this training set in Table 6. The results have a similar trend as our experiments on the 95% vs. 5% setting.

Dataset	Waterbirds (10% tail subpopulations)							
Noise rate	25%		30%		35%		40%	
Method/Group	Avg.	Worst	Avg.	Worst	Avg.	Worst	Avg.	Worst
ERM	85.59	52.18	85.92	50.46	77.06	54.98	Not converged	
DivideMix*	**90.66**	63.70	88.63	64.95	73.52	46.11	87.25	69.00
Ours	90.09	**73.52**	**90.20**	**76.47**	**82.53**	**61.37**	**89.82**	**72.11**

Table 6. Comparison over the corrupted Waterbirds dataset under 90% head subpopulation samples vs. 10% tail subpopulation samples. *We use the same label confidence estimation method as DivideMix. For a fair comparison, we replace its other components and training schemas with ours.

D. Data augmentation

We perform an ablation study for the augmentation techniques in Table 7. We also further verify our method's superiority under the same data augmentation on CIFAR in Table 8.

Noise rate	25%		30%		35%		40%	
Method/Group	Avg.	Worst	Avg.	Worst	Avg.	Worst	Avg.	Worst
Random crop and random horizontal flip	82.49	54.81	75.31	50.43	77.15	53.33	74.40	38.51
Cutout	87.82	67.40	78.40	56.89	82.74	64.44	75.90	48.88
RandAugment (ours)	**88.40**	**70.56**	**86.09**	**69.31**	**84.93**	**71.88**	**79.77**	**59.97**

Table 7. Comparison over the strong augmentation techniques on Waterbirds dataset

Dataset		CIFAR-10					CIFAR-100			
Noise type		Sym.				Asym.	Sym.			
Method/Noise ratio		20%	50%	80%	90%	40%	20%	50%	80%	90%
DivideMix-AutoAugment*	Best	**96.3**	95.4	93.8	91.9	94.6	**79.5**	**77.2**	66.4	41.2
	Last	**96.2**	95.1	93.6	91.8	**94.3**	**79.2**	**77.0**	**66.1**	40.9
DivideMix-RandAugment*	Best	96.1	-	-	89.6	-	78.1	-	-	36.8
	Last	96.0	-	-	89.4	-	77.8	-	-	36.7
Ours	Best	**96.1**	**95.9**	**95.9**	**94.8**	**94.7**	**79.2**	**75.5**	**67.7**	**51.0**
	Last	**96.0**	**95.8**	**95.8**	**94.6**	**94.1**	**78.1**	**74.5**	**65.3**	**49.2**

Table 8. Comparison over the augmentation techniques on CIFAR. Results are from [29], which adds recent data augmentation techniques on DivideMix.

Out RandAugment follow the version in FixMatch [36] https://github.com/google-research/fixmatch. Table 9 provides the augmentation pool of the FixMatch-version RandAugment[36].

Operation	Range	Operation	Range
AutoContrast	[0, 1]	Rotate	[-30, 30]
Brightness	[0.05, 0.95]	Sharpness	[0.05, 0.95]
Color	[0.05, 0.95]	ShearX	[-0.3, 0.3]
Contrast	[0.05, 0.95]	ShearY	[-0.3, 0.3]
Equalize	[0, 1]	Solarize	[0, 256]
Identity	[0, 1]	TranslateX	[-0.3, 0.3]
Posterize	[4, 8]	TranslateY	[-0.3, 0.3]

Table 9. List of operations for strong transformations of the modified RandAugment. Three transformations are randomly chosen and performed with stochastic magnitude.

E. More noisy/clean classification results

We provide more noisy/clean classification results in Fig. 5. Our method gains much better noisy/clean classification for tail subpopulations.

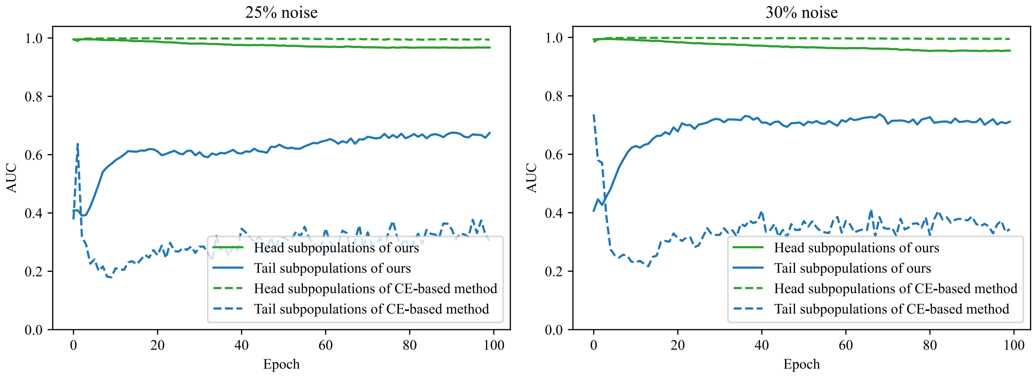

Figure 5. Clean/noisy classification AUC in every epoch of different subpopulations' data on the corrupted Waterbirds dataset under 25% and 30% noise.

F. Hyper-parameters

F.1. corrupted datasets with explicit imbalanced subpopulations

Our method mainly has two Hyper-parameters: The k in knn for LLC and τ for the refurbish-DRO. For other hyper-parameters in network training and GMM, we don't further tune them and follow previous methods.

For both datasets: The backbone model is Resnet18. The model is trained for 100 rounds after 5 epochs of warm-up. In every round, we train the network using SGD with a learning rate of 0.0005, a momentum of 0.9, a weight decay of 0.0005, and a batch size of 128. The learning rate is reduced by a factor of 10 in the last 20 rounds. For Waterbirds, the GMM is fitted with a maximal iteration of 10, a convergence threshold of 0.001, and a non-negative regularization of 0.0005. The k in knn for LLC is 50. Under 25%,30% noise rate, the τ for refurbish-DRO is 70%. Under 35%,40% noise rate, the τ for refurbish-DRO is *0%. For CelebA, the GMM is fitted with a maximal iteration of 10, a convergence threshold of 0.001, and a non-negative regularization of 0.001. The k in knn for LLC is 25. Under 15%,20% noise rate. the τ for refurbish-DRO is 70%, Under 35%,40% noise rate. the τ for refurbish-DRO is 80%. The performance is averaged over 3 trials to avoid randomness.

F.2. Standard LNL datasets

For CIFAR: the model is trained for 500 rounds after 15 epochs of warm-up. In every round, we train the network using SGD with a learning rate of 0.03, a momentum of 0.9, a weight decay of 0.0005, and a batch size of 448. The learning rate is reduced by a factor of 10 in the last 100 rounds. The GMM is fitted with a maximal iteration of 10, a convergence threshold of 0.01, and a non-negative regularization of 0.0005. The τ for refurbish-DRO is 100%, meaning that we don't explicitly enforce model focus on tail-subpopulation. For CIFAR-10, the k in knn for LLC is 1000. For CIFAR-100, the k in knn for LLC is 400.

For Mini-WebVision: the backbone model is the inception-resnet v2 [40]. The model is trained for 300 rounds after 1 epoch of warm-up. In every round, we train the network using SGD with a learning rate of 0.01, a momentum of 0.9, a weight decay of 0.0005, and a batch size of 160. The learning rate is reduced by a factor of 10 in the last 100 rounds. For GMM, the convergence threshold is 0.01, the non-negative regularization is 0.001, and other hyper-parameters follow the default settings of scikit-learn. The k in knn for LLC is 10. The τ for refurbish-DRO is 100%.

For ANIMAL-10N, the model is trained for 500 rounds after 15 epochs of warm-up. In every round, we train the network using SGD with a learning rate of 0.01, a momentum of 0.9, a weight decay of 0.0005, and a batch size of 128. The learning rate is reduced by a factor of 10 in the last 100 rounds. The model is the VGG-19 [35]. The k in knn for LLC is 5000. The τ for refurbish-DRO is 90%. For GMM, the convergence threshold is 0.01, and the non-negative regularization is 0.0005. Other hyper-parameters follow the default settings of scikit-learn.