A TECHNICAL APPENDICES

A.1 DERIVATION AND FUNCTION OF CRETTA

A.1.1 DERIVATION OF CRETTA

The marginal distribution of target data p_θ can be written as the product of the pretrained source model q_ϕ and an exponential residual term:

$$p_\theta(x) = \frac{1}{Z} q_\phi(x) \exp(-\frac{1}{\beta}\tilde{E}_\theta(x)),$$

where $\tilde{E}_\theta$ represents the residual energy function encoding the distribution shift. During TTA, the source model q_ϕ remains fixed, and the objective is to learn $\tilde{E}_\theta$ so as to align the source model more closely with the target distribution. By expanding the equation with respect to the energy function $\tilde{E}_\theta$, we can compute the residual energy score of an image sample x.

$$\tilde{E}_\theta(x) = -\beta \left(\log \frac{p_\theta(x)}{q_\phi(x)} + \log Z \right).$$

Next, we substitute the ground-truth residual energy function $\tilde{E}_\theta^*$ into the Bradley-Terry (BT) model (Bradley & Terry, 1952), which only depends on the difference in energy values between source and target pairs:

$$P(\tilde{E}(x_t) < \tilde{E}(x_s)) = \frac{1}{1 + \exp(-\tilde{E}_\theta(x_s) + \tilde{E}_\theta(x_t))} = \frac{1}{1 + \exp\left(\beta \log \frac{p_\theta(x_s)}{q_\phi(x_s)} - \beta \log \frac{p_\theta(x_t)}{q_\phi(x_t)}\right)},$$

where x_t and x_s denote the target and source samples, respectively. Here, for pairwise comparison, we use the negative residual energy.

Having derived the probability of the target distribution data in terms of the optimal energy function, which can further be expressed using ϕ and θ, our objective for the target model is as follows:

$$\mathcal{L}(\theta; \phi) = -\mathbb{E}_{(x_s, x_t)\sim B} \left[\log \sigma \left(\beta \log \frac{p_\theta(x_t)}{q_\phi(x_t)} - \beta \log \frac{p_\theta(x_s)}{q_\phi(x_s)} \right) \right] \tag{5}$$

In section 3, we emphasize that the key advantage of CRETTA is that it avoids the costly Stochastic Gradient Langevin Dynamics (SGLD) sampling required to compute the normalization constant as required in TEA (Yuan et al., 2024). However, the objective Equation 5 still includes the normalization constant for both target and source model.

To eliminate both normalization constants, we first redefine the target and source models using the Gibbs distribution as follows:

$$p_\theta(x) = \frac{\exp(-E_\theta(x))}{Z(\theta)}, q_\phi(x) = \frac{\exp(-E_\phi(x))}{Z(\phi)}$$

By integrating p_θ and q_ϕ into the Equation 5 and applying the logarithm, the normalization constants for both target and source model, i.e., $Z(\theta)$ and $Z(\phi)$, are canceled out as shown in below:

$$\begin{aligned}\mathcal{L}(\theta; \phi) = -\mathbb{E}_{(x_s, x_t)\sim B} \Big[\ln \sigma \Big(& \beta \big(- E_\theta(x_t) - \cancel{\ln Z(\theta)} + E_\phi(x_t) + \cancel{\ln Z(\phi)} \big) \\ & - \beta \big(- E_\theta(x_s) - \cancel{\ln Z(\theta)} + E_\phi(x_s) + \cancel{\ln Z(\phi)} \big) \Big) \Big]\end{aligned} \tag{6}$$

Therefore, the final learning objective is expressed as follows:

$$\mathcal{L}(\theta;\phi) = -\mathbb{E}_{(x_s,x_t)\sim B}\left[\ln\sigma\left(\beta\left(-E_\theta(x_t) + E_\theta(x_s) + E_\phi(x_t) - E_\phi(x_s)\right)\right)\right]$$

A.1.2 Function of CRETTA

In this section, we provide a detailed explanation of how each component of CRETTA contributes to adaptation, as well as the expected behavior during early and late stages of online adaptation.

If the target model θ successfully optimizes this objective, then residual energy function $\tilde{E}_\theta(x)$ in $p_\theta(x) = \frac{1}{Z}q_\phi(x)\exp(-\frac{1}{\beta}\tilde{E}_\theta(x))$ models the residual component of the distribution shift between the source and target domains.

At the beginning of adaptation, the model has not yet encoded the distribution shift. Therefore, the residual energy $E_\theta(x)$ is close to zero for both source samples x_s and target samples x_t. This results in: $p_\theta(x) \approx q_\phi(x)$ meaning that predictions for both source and target data remain similar to the source model outputs.

As training progresses, the residual energy function learns the discrepancy between target and source distributions. For source samples x_s, $\tilde{E}_\theta(x_s)$ remains small, leading to $p_\theta(x_s) \approx q(x_s)$, preserving source performance. For target samples x_t, the residual energy adjusts predictions reflecting the learned domain shift and improving performance on the target domain. By progressively learning the residual while maintaining alignment with the source model, CRETTA achieves better generalization.

A.1.3 Buffer Management of CRETTA

CRETTA initializes the source buffer $\mathcal{B}_s$ at model initialized, prior to adaptation, by randomly sampling source data up to a fixed buffer size, with an equal number of samples per class. During adaptation, the samples in the buffer are used sequentially in batches without any additional sampling or refresh, unlike TEA, thereby incurring no additional computational overhead.

Table 8: Comparison of classification accuracy (Acc ↑) and expected calibration error (ECE ↓) on the CIFAR10-C, CIFAR100-C, and TinyImageNet-C datasets at corruption severity level 5, the average across severity levels 1-5, and on clean data. The best adaptation results are emphasized in **BOLD**, while the second-best results are UNDERLINED.

	CIFAR-10-C					CIFAR-100-C					TinyImageNet-C				
	Clean	Corr Severity 5		Corr Severity 1-5 Avg		Clean	Corr Severity 5		Corr Severity 1-5 Avg		Clean	Corr Severity 5		Corr Severity 1-5 Avg	
Method	Acc(↑)	Acc(↑)	ECE(↓)	Acc(↑)	ECE(↓)	Acc(↑)	Acc(↑)	ECE(↓)	Acc(↑)	ECE(↓)	Acc(↑)	Acc(↑)	ECE(↓)	Acc(↑)	ECE(↓)
Source	95.08%	81.73%	10.18%	88.82%	5.45%	76.28%	53.25%	17.71%	64.11%	11.73%	59.60%	35.12%	16.17%	43.16%	13.46%
Normalization															
BN Adapt	93.59%	85.46%	4.85%	89.12%	3.15%	72.84%	60.74%	8.32%	65.83%	6.88%	56.72%	39.60%	13.66%	44.72%	12.12%
Pseudo Labeling															
PL	**94.85%**	84.85%	10.10%	90.09%	6.20 %	**75.98%**	56.33%	23.81%	65.72%	16.66%	**58.95%**	35.40%	30.95%	43.79%	23.47%
SHOT	94.38%	87.91%	5.42%	90.78%	3.86 %	75.00%	64.41%	8.93%	68.80%	7.44%	56.90%	39.84%	13.81%	44.95%	12.24%
Entropy Minimization															
TENT	94.35%	87.84%	5.49 %	90.74%	3.89 %	74.95%	64.31%	8.93%	68.73%	7.47%	56.92%	39.83%	13.82%	44.94%	12.24%
ETA	93.72%	85.46%	4.85 %	89.12%	3.15 %	73.71%	61.77%	8.54%	66.66%	7.10%	56.82%	39.67%	13.70%	44.79%	12.16%
EATA	93.72%	85.46%	4.85%	89.12%	3.15%	73.66%	61.79%	8.54%	66.65%	7.11%	56.86%	39.68%	13.70%	44.79%	12.16%
SAR	93.61%	86.54%	4.79%	89.80%	3.13%	73.73%	62.71%	8.31%	67.36%	6.91%	56.77%	39.66%	13.72%	44.77%	12.16%
AEA	94.21%	88.27%	5.09%	90.88%	3.73%	75.17%	64.40%	9.16%	68.75%	7.61%	56.97%	39.87%	13.82%	44.97%	12.25%
Energy-based Models															
TEA	94.06%	88.06%	**3.83%**	90.67%	**2.68%**	74.18%	63.66%	**7.68%**	67.93%	**6.33%**	57.17%	39.96%	13.84%	45.08%	12.24%
CRETTA (Ours)	94.43%	**88.30%**	4.15%	**91.01%**	2.88%	75.26%	**64.52%**	7.99%	**69.05%**	6.82%	58.23%	**40.30%**	**13.52%**	**45.75%**	**11.85%**

A.2 Additional Experiments and Analysis

A.2.1 Detailed Performance Comparison

Detailed Performance Comparison on Accuracy Table 8 reports accuracy on the highest severity level 5, the average across severity levels (1-5), and performance on the clean dataset (i.e., without corruption) for CIFAR10-C, CIFAR100-C, and TinyImageNet-C. This table extends the results of Table 1 by additionally reporting accuracy on the clean dataset, providing a more complete view of model performance.

While CRETTA achieves the second-best accuracy on clean data among all methods, with the PL method performing the best. However, this can lead to overfitting and significant degradation in

Table 9: Comparison of expected calibration error (ECE ↓) on TinyImageNet-C datasets across all corruptions at the average across severity level 1-5. (Values are reported to one decimal place for space efficiency.)

	Noise			Blur				Weather				Digital				
Method	**Gauss.**	**Shot**	**Impul.**	**Defoc.**	**Glass**	**Motion**	**Zoom**	**Snow**	**Frost**	**Fog**	**Brit.**	**Contr.**	**Elastic**	**Pixel**	**JPEG**	**Avg**
Source	12.5%	**11.6%**	13.1%	10.8%	13.3%	10.8%	10.5%	14.7%	14.6%	18.4%	13.9%	25.9%	11.0%	**10.1%**	10.6%	13.5%
BN Adapt	12.1%	11.9%	**12.4%**	11.0%	11.9%	11.0%	10.3%	13.1%	12.7%	14.0%	12.1%	16.7%	10.9%	10.7%	10.9%	12.1%
PL	20.9%	19.5%	26.7%	18.4%	20.4%	18.1%	17.7%	22.8%	20.6%	38.4%	19.7%	54.7%	18.3%	17.6%	18.2%	23.5%
SHOT	12.2%	12.1%	12.5%	11.1%	12.1%	11.2%	10.5%	13.2%	12.8%	14.2%	12.2%	16.9%	11.0%	10.7%	11.0%	12.2%
TENT	12.2%	12.1%	12.5%	11.1%	12.1%	11.1%	10.5%	13.2%	12.8%	14.2%	12.2%	16.9%	11.0%	10.8%	11.0%	12.2%
ETA	12.1%	12.0%	12.5%	11.1%	12.0%	11.1%	10.4%	13.1%	12.7%	14.1%	12.2%	16.7%	10.9%	10.7%	10.9%	12.2%
EATA	12.1%	12.0%	12.4%	11.1%	12.0%	11.1%	10.4%	13.2%	12.7%	14.1%	12.2%	16.8%	10.9%	10.7%	10.9%	12.2%
SAR	12.1%	12.0%	12.5%	11.1%	12.0%	11.1%	10.4%	13.2%	12.7%	14.1%	12.2%	16.8%	10.9%	10.7%	10.9%	12.2%
AEA	12.2%	12.1%	12.5%	11.2%	12.1%	11.2%	10.4%	13.3%	12.8%	14.2%	12.2%	16.9%	11.0%	10.8%	11.0%	12.3%
TEA	12.1%	12.0%	12.6%	11.2%	12.1%	11.1%	10.5%	13.2%	12.7%	14.1%	12.2%	16.9%	11.0%	10.8%	11.0%	12.2%
CRETTA	**12.0%**	11.7%	12.4%	**10.7%**	**11.9%**	**10.5%**	**10.0%**	**12.7%**	**12.1%**	**13.6%**	**11.9%**	**16.6%**	**10.7%**	10.3%	**10.6%**	**11.9%**

performance under severe corruptions. Notably, while PL exhibits substantial drops in performance under corruption, CRETTA remains robust and effective across both clean and corrupted settings, demonstrating its reliability in both distributions.

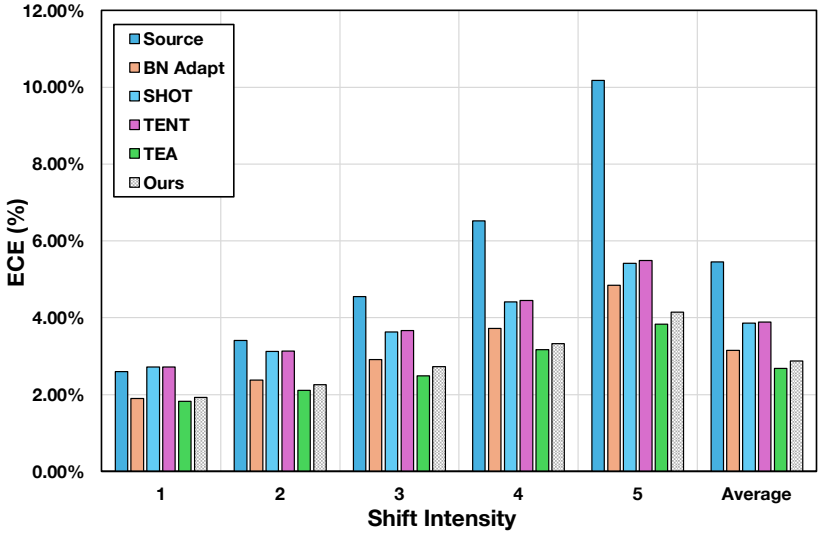

Figure 4: Comparison of Expected Calibration Error (ECE↓) on the CIFAR10-C dataset across different corruption severity levels.

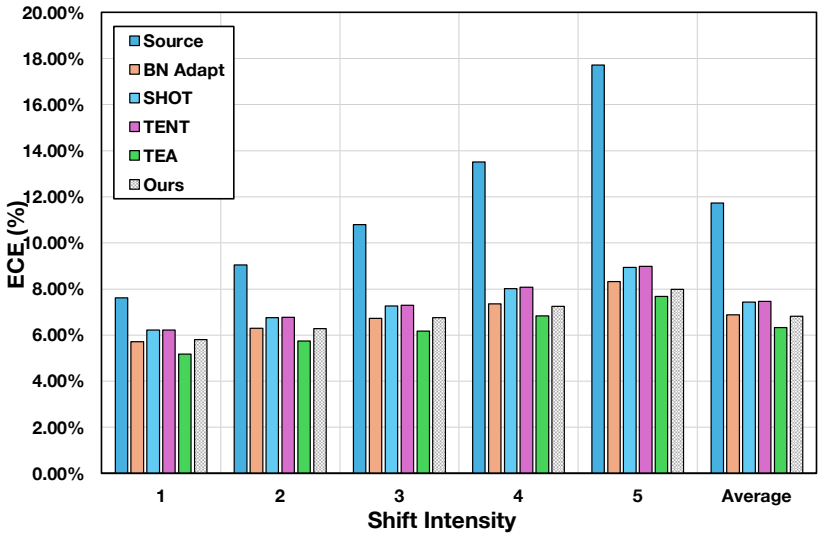

Figure 5: Comparison of Expected Calibration Error (ECE↓) on the CIFAR100-C dataset across different corruption severity levels.

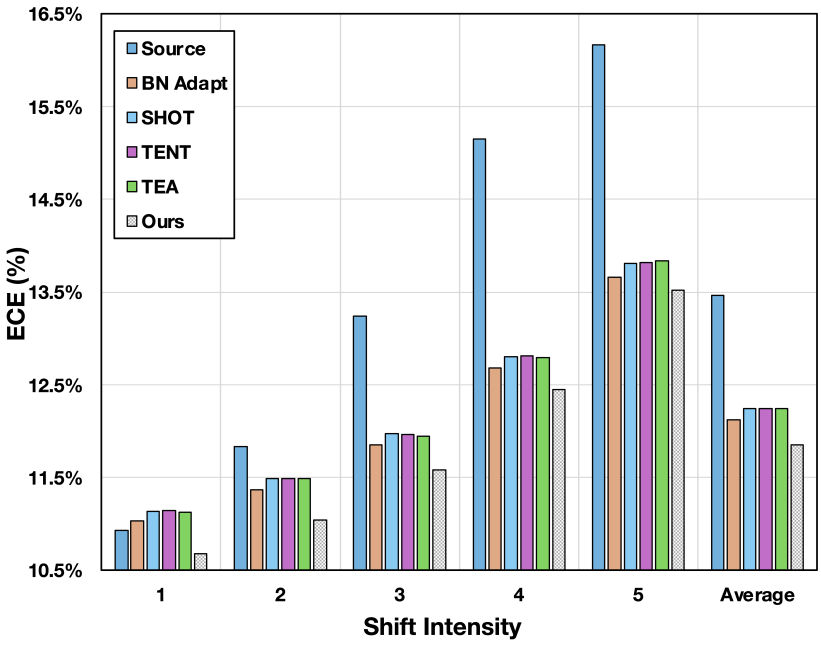

Figure 6: Comparison of Expected Calibration Error (ECE↓) on the TinyImageNet-C dataset across different corruption severity levels.

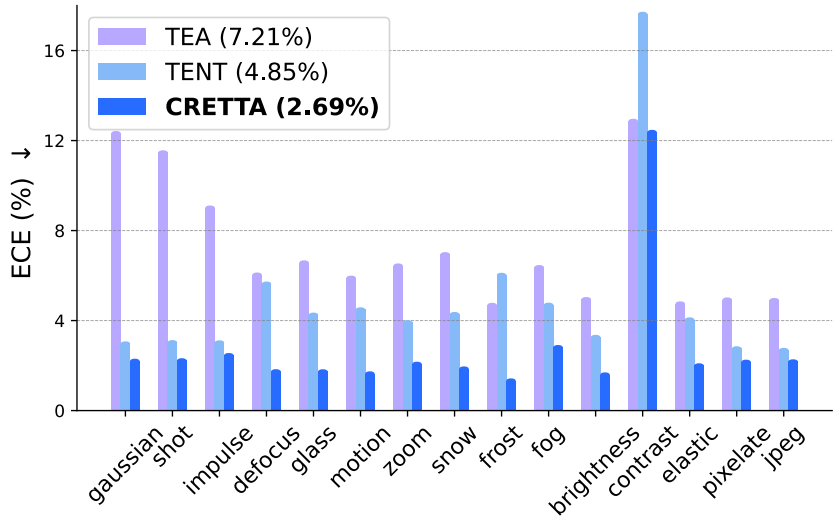

Figure 7: Comparison of Expected Comparison of Expected Calibration Error (ECE ↓) on ImageNet-C across various corruption types, with results averaged over severities 1–5..

Detailed Performance Comparison on Calibration Error In this section, we provide a detailed analysis of the Expected Calibration Error (ECE) for CIFAR10-C and CIFAR100-C. This expands upon the results shown in Table 1.

As seen in Figure 4 and Figure 5, energy-based methods such as TEA and CRETTA consistently outperform baseline approaches like TENT, which suffers from overconfidence issues. Furthermore, our method maintains computational advantage over TEA, making it more efficient while achieving comparable or superior performance.

On TinyImageNet-C dataset, shown in Figure 6, CRETTA outperforms all competing methods across all severity levels. This consistent superiority over all baseline methods demonstrates the robustness and adaptability of our approach in high-complexity datasets.

Table 10: Comparison of computational cost (GFLOPs), Memory Cost (Peak Memory Usage), and performance metrics (ECE and Acc) for baselines on the CIFAR10-C

	GFLOPs(↓)	Memory Cost(↓)	Acc(↑)	ECE(↓)
Source	131.53	443.98 MB	88.82%	5.45%
BN Adapt	131.53	452.61 MB	89.12%	3.15%
TENT	132.59	1546.05 MB	90.74%	3.89%
TEA	4335.82	3464.78 MB	90.67%	2.68%
Ours	527.40	2651.83 MB	91.01%	2.88%

Table 11: Comparison of computational cost (GFLOPs), Memory Cost (Peak Memory Usage), and performance metrics (ECE and Acc) for baselines on the CIFAR100-C

	GFLOPs(↓)	Memory Cost(↓)	Acc(↑)	ECE(↓)
Source	131.53	443.03 MB	64.11%	11.73%
BN Adapt	131.53	452.70 MB	65.83%	6.88%
TENT	132.59	1546.21 MB	68.73%	7.47%
TEA	4335.82	3465.00 MB	67.93%	6.33%
Ours	527.40	2651.85 MB	69.05%	6.82%

Detailed Performance Comparison on Computational Efficiency To further demonstrate the computational advantages of our proposed method, we present a comprehensive comparison of com-

Table 12: Comparison of classification accuracy (Acc ↑) and expected calibration error (ECE ↓) on the ImageNet-C dataset.

	Severity L5		Severity Avg	
Method	**Acc ↑**	**ECE ↓**	**Acc ↑**	**ECE ↓**
TENT	**37.39%**	7.75%	**43.78%**	4.85%
TEA	31.60%	8.39%	38.72%	7.21%
CRETTA	37.05%	**4.43%**	43.54%	**2.69%**

putational cost (GFLOPs), peak GPU memory usage (MB) with performance metrics (Accuracy and ECE) across CIFAR10-C, CIFAR100-C, and TinyImageNet-C, as summarized in Table 10,Table 11. Compared to TEA, which incurs substantial computational overhead due to SGLD-based sampling, CRETTA reduces GFLOPs by more than sevenfold across datasets. Furthermore, despite incorporating a source buffer, CRETTA maintains a modest peak GPU memory usage, significantly lower than TEA. The peak GPU memory usage is measured as the maximum allocated GPU memory during adaptation. Consequently, CRETTA offers a practical balance between performance, computational cost, and memory efficiency, making it well-suited for deployment in real-world, resource-constrained environments.

Scalability In this section, we provide a detailed results on ImageNet-C.

As shown in Table 12, CRETTA achieves performance by a significant margin, outperforming the entropy-based method TENT and the existing energy-based method TEA in ECE.

Entropy minimizations's overconfidence and MLE-based approach's approximation error introduced when estimating its normalization constant term leads to poor calibration which is inappropriate in real-wold TTA scenarios. In contrast, CRETTA generalizes well to large-scale datasets, achieving strong predictive performance with superior calibration.

Table 13: Comparison of classification accuracy on CIFAR10(-C), CIFAR100(-C) under gradual distribution shift

	CIFAR10			CIFAR100		
Domain	**OURS**	**TEA**	**TENT**	**OURS**	**TEA**	**TENT**
Source (Q)	**93.46**	93.45	93.43	**73.97**	73.88	73.57
1	**92.88**	92.80	92.77	**71.90**	71.41	71.70
2	**92.03**	91.92	91.92	**71.57**	70.40	71.36
3	**91.63**	91.29	91.35	69.99	67.71	**70.04**
4	**90.25**	89.81	90.03	67.99	65.23	**68.28**
5 (P)	**89.47**	88.78	88.58	**65.47**	60.26	65.23

Detailed Performance Comparison Under Gradual Shift scenario In subsection 4.3, we demonstrated that our contrastive residual energy-based learning shows superior performance over CD MLE-based adaptation method TEA. This tendency was consistently observed under the gradual distribution shift setting in Table 13, and here we additionally report comparisons with TENT.

For CIFAR10-C, CRETTA maintains the best performance throughtout the shift. For CIFAR100-C, CRETTA shows clear gains under stronger shifts. At severity 5, it achieves 65.47%, notably higher than TEA(60.26%) and TENT (65.23%). While TENT is compertitive at mid-level severities, it degrades more under severe shifts. Overall, CRETTA provides robust adaptation across gradual shifts while preventing forgetting, outperforming both TEA and TENT.

Test-time Adaptation for Non-IID Settings Our previous experiments are conducted under the assumption of i.i.d. test samples which is a widely adopted setting in prior work. Nonetheless, real-world applications can also encounter non-i.i.d. samples (Gong et al., 2022; Yuan et al., 2023; Wang et al., 2022). To further examine the robustness and generalizability of our method beyond the i.i.d. assumption, we constructed a non-i.i.d. test-time adaptation scenario. Specifically, we simulated non i.i.d. data stream by leveraging a Dirichlet distribution to control the class allocation ratio within batch, denoted as δ. A higher δ value brings the distribution closer to i.i.d., whereas a lower δ value results in a more non-i.i.d. distribution, where a specific class might dominate the batch. We conducted our experiment on CIFAR100-C using the WRN-28-10 backbone.

Table 14: Test-time adaptation in dynamic scenarios using CIFAR100-C at severity 5. Our method demonstrates higher robustness compared to baselines across varying the allocation ratio δ.

Method	$\delta = 10$	$\delta = 1$	$\delta = 0.1$	$\delta = 0.01$	Avg Acc.
BN Adapt	61.44%	61.11%	59.02%	45.61%	56.79%
PL	44.03%	37.27%	39.06%	43.38%	40.93%
SHOT	63.94%	63.60%	61.20%	46.54%	58.82%
TENT	63.91%	63.56%	61.20%	46.72%	58.85%
ETA	62.31%	62.04%	59.89%	46.04%	57.57%
EATA	62.35%	62.04%	59.84%	46.04%	57.57%
SAR	61.54%	61.22%	59.12%	45.66%	56.89%
TEA	62.58%	62.29%	60.08%	46.22%	57.79%
Ours	**66.20%**	**65.95%**	**63.47%**	**48.33%**	**60.99%**

As Table 14 shows, our method consistently outperforms entropy minimization and instance selection approaches across all δ values. Specifically, CRETTA achieves the highest average accuracy of 60.99%, surpassing TENT's 58.85% by 2.14%p. Also, even at the most imbalanced setting where $\delta = 0.01$, our method achieves a competitive accuracy of 48.33%. These findings demonstrate that our method not only excels in i.i.d. scenarios but also is effective in dynamic real-world environments.

A.3 Ablation Study

A.3.1 Detailed Ablation study

Table 15: Comparison of classification accuracy(Acc) and expected calibration error(ECE) on benchmark datasets between CRETTA(Default) and CRETTA(Loss Term without Source Model) at severity level 5.

Method	CIFAR10-C		CIFAR100-C		TinyImageNet-C	
	Acc(↑)	ECE(↓)	Acc(↑)	ECE(↓)	Acc(↑)	ECE(↓)
CRETTA	**88.30**	**4.15**	**64.52**	7.99	**40.30**	**13.52**
w.o Source Model Term	88.09	4.66	60.02	**5.93**	37.46	14.55

Loss Ablation We observed that eliminating the source model consistently degraded both accuracy and calibration (ECE) in most cases across our benchmark datasets. These results collectively demonstrate that incorporating source model related terms into our contrastive residual learning is essential for stable adaptation.

Gradient Ablation The gradient coeffcient $w(x_t, x_s)$ is the key mechanism that turns relative energy into stable updates. To verify this role, we conducted an ablation study that disrupts the proposed weighting scheme by replacing $w(x_t, x_s)$ with values randomly sampled from a uniform distribution $[0, 1)$. As shown in Table 16, this replacement lead to lower accuracy and higher calibration error, confirming that gradient coefficient is critical for stable optimization and robust adaptation under noisy target data.

Table 16: Effect of Gradient Coefficient

Method	CIFAR10-C		CIFAR100-C		TinyImageNet-C	
	Acc	ECE	Acc	ECE	Acc	ECE
Ours	**88.30**	4.15	**64.52**	**7.99**	**40.30**	**13.52**
Uniform	87.47	**4.13**	61.66	8.03	38.33	15.13

Table 17: Comparison of classification accuracy(Acc) and expected calibration error(ECE) on benchmark datasets between CRETTA(Default) and CRETTA(Single Source Sample in Buffer) at severity level 5.

Method	CIFAR10-C		CIFAR100-C		TinyImageNet-C	
	Acc(↑)	ECE(↓)	Acc(↑)	ECE(↓)	Acc(↑)	ECE(↓)
CRETTA	**88.30**	**4.15**	**64.52**	**7.99**	**40.30**	**13.52**
CRETTA with single source sample	87.62	5.39	62.67	8.93	**40.30**	14.13

Extended Buffer Ablation While the specific content of the buffer has less impact on performance, as shown in Table 5, this does not imply that the source buffer itself plays a trivial role. To further verify this, we additionally conducted an experiment where the buffer consists of only a single source sample. As shown in Table 17, accuracy dropped by up to 1.7% and ECE increased by up to 1.2% across datasets.

Table 18: Effectiveness of preference pair size on CIFAR10-C, CIFAR100-C, and TinyImageNet-C.

	CIFAR10-C	CIFAR100-C	TinyImageNet-C
CRETTA wo/ CP	88.30%	64.52%	40.30%
CRETTA w/ CP	88.24%	64.69%	40.44%

Pair Size Ablation In CRETTA, we assume that the samples in a test batch represent the target distribution, while the source replay buffer represents the source distribution. The loss is computed by forming pairs between target and source samples within each batch, enabling a direct comparison between the two distributions.

To demonstrate the assumption is valid, we examined the impact of increasing the number of pair combinations using a Cartesian Product (CP) to generate all possible combinations of target and source data within each batch. For example, we use 200 pairs for each adaptation in CIFAR10-C, while the Cartesian Product results in 200×200 pairs.

Our results across three datasets summarized in Table 18 indicate that generating more pairs does not necessarily lead to performance gain. With only a few pairs, CRETTA can efficiently adapt to the target distribution.

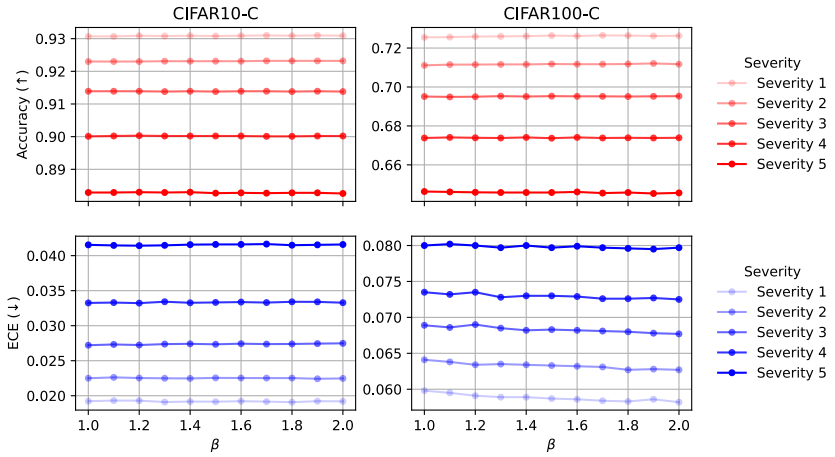

Figure 8: Ablation on varying β values on CIFAR10-C and CIFAR100-C at severity 1-5.

Hyperparameter β Ablation The hyperparameter β in Equation 3 controls the deviation from the pretrained source model, serving as a scaling parameter. To evaluate the robustness of our method, we experiment its performance across varying values of β, assessing both accuracy and expected calibration error (ECE) on CIFAR10-C, CIFAR100-C and TinyImageNet-C. As shown in Figure 8, our method consistently demonstrates stable performance across all corruption severity levels (1-5), validating its robustness.

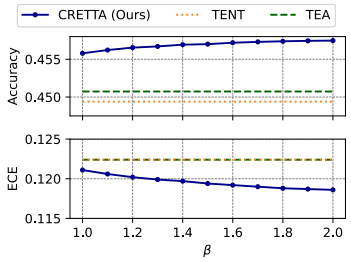

Figure 9: Ablation on varying values of β.

In addition, we further examine the effectiveness of CRETTA across varying values of the hyperparameter β on TinyImageNet-C, averaging results over severity levels 1 to 5, and compare its performance against competitive baselines (see Figure 9). These results confirm that the strong adaptation performance of CRETTA is not reliant on a specific setting of the temperature parameter β, but rather stems from our contrastive residual learning objective itself.

A.3.2 Detailed Setting of CRETTA

Table 19: Detailed hyperparameters settings for each dataset.

Dataset	LR	β	Batch Size	Transformation Type (probability)
CIFAR10-C	1e-3	1.0	200	rotate(1.0)
CIFAR100-C	2e-3	2.0	200	flip, rotate, affine, perspective, crop(0.2)
TinyImageNet-C	1e-3	2.0	1000	None

Hyperparameters This section details the hyperparameter settings for CRETTA. To optimize performance, minimal hyperparameter tuning was conducted, focusing solely on learning rate, β and type and probability of random transformations for source buffer. With only slight adjustments, CRETTA achieved significantly better performance than the current state-of-the-art (SOTA). The batch sizes were aligned with the default settings used in TENT and TEA, which are 200 for CIFAR10-C and CIFAR100-C, 1000 for TinyImageNet-C. For ImageNet-C we follow TENT default settings, using a batch size of 64 and learning rate of $2.5\text{e}{-4}$. These settings ensured consistency across experiments while highlighting the robustness and effectiveness of CRETTA. For the PACS domain-generalization task, we used a learning rate of $1\text{e}{-3}$, a batch size of 100, applying source-sample augmentation in the same way as for CIFAR100-C. All experiments were conducted using a single NVIDIA RTX A6000 GPU (48GB).

Evaluation Metrics Expected Calibration Error (ECE) (Guo et al., 2017) is a metric used to measure the calibration quality of a probabilistic model. Calibration refers to how closely the predicted probabilities of a model match the actual probabilities. ECE quantifies the discrepancy between predicted confidence and actual accuracy. ECE is calculated as shown in Equation 7:

$$\text{ECE} = \sum_{m=1}^{M} \frac{|\text{bin}_m|}{N} \cdot |\text{confidence}_m - \text{accuracy}_m| \tag{7}$$

where M is the number of bins, N is the total number of data points, bin_m is the number of predictions in m-th bin, and confidence_m and accuracy_m are the confidence and accuracy of bin m, respectively.

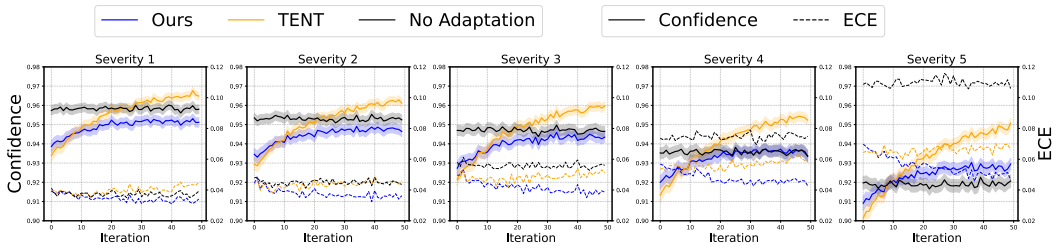

Figure 10: The overconfidence problem of entropy minimization in test-time adaptation on CIFAR10-C. TENT tends to increase a model's confidence in uncertain predictions as adaptation progresses, often leading to worse calibration due to overconfidence. In contrast, CRETTA (Ours) stabilizes the adaptation process by gradually reducing the expected calibration error.

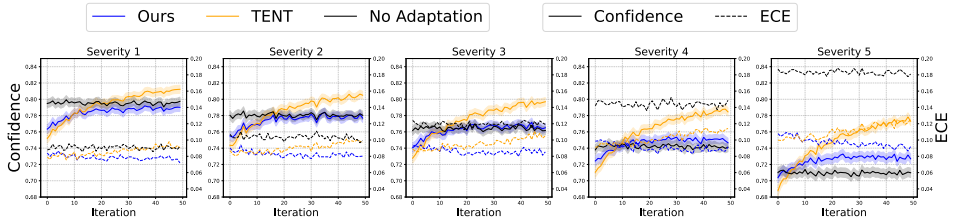

Figure 11: The overconfidence problem of entropy minimization in test-time adaptation on CIFAR100-C.

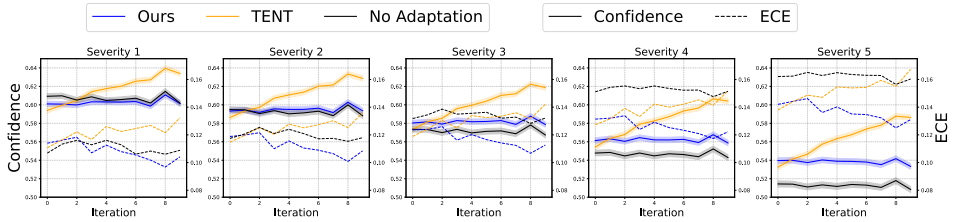

Figure 12: The overconfidence problem of entropy minimization in test-time adaptation on TinyImageNet-C.

A.3.3 Detailed Results for Overconfidence Problem of Entropy Minimization

The overconfidence issue inherent in entropy minimization has been thoroughly investigated in prior works (Liu et al., 2020; Hendrycks & Gimpel, 2016; Guo et al., 2017). Building on this, we explored that increasing a model's prediction confidence especially when the label information is unavailable can lead to bad calibration as shown in Figure 4. The trend consistently appears in other benchmark datasets including CIFAR100-C and TinyImageNet-C as illustrated in Figure 11 and Figure 12. Entropy minimization raises the model's confidence across all severity levels, with the rate of increase becoming steeper as corruption severity intensifies, thereby exacerbating error accumulation.

On the other hand, CRETTA maintains stable confidence managing uncertainty during test-time adaptation and even reduces calibration error as adaptation progresses. These results suggest that maximizing the marginal likelihood of target samples provides a safer and more effective strategy compared to relying on uncertain predicted probabilities $p_\theta(\hat{y}|x)$ in the test-time learning objective.

B Noise Contrastive Estimation

We first define a reward function $r(\cdot)$ to properly compare samples from two different sets or distributions.

$$r(x;\theta,\phi) = \log P_\theta(x) - \log P_\phi(x)$$

where P_θ is the target distribution and P_ϕ is the source distribution.

B.1 NON-RESIDUAL

If we define energy functions for each of them by utilizing gibbs distribution,

$$E_\theta(x) = -\log P_\theta(x) - \log Z(\theta)$$
$$E_\phi(x) = -\log P_\phi(x) - \log Z(\phi)$$

Then the reward function becomes

$$r(x;\theta,\phi) = -(E_\theta(x) - E_\phi(x)) + C$$

Then the loss function becomes

$$\mathcal{L}(\theta;\phi) = -\mathbb{E}_{x_t}\left[\log\sigma(r(x;\theta,\phi))\right] - \mathbb{E}_{x_s}\left[\log(1-\sigma(r(x;\theta,\phi)))\right]$$

B.2 RESIDUAL

If we define a residual energy function,

$$p_\theta(x) = \frac{1}{Z}q_\phi(x)\text{exp}(-\frac{1}{\beta}\tilde{E}_\theta(x))$$

Then the reward function becomes

$$r(x;\theta,\phi) = \log p_\theta(x) - \log q_\phi(x) = -\frac{1}{\beta}\tilde{E}_\theta(x) + c$$

Then the loss function becomes

$$\mathcal{L}(\theta;\phi) = -\mathbb{E}_{x_t}\left[\log\sigma(r(x;\theta,\phi))\right] - \mathbb{E}_{x_s}\left[\log(1-\sigma(r(x;\theta,\phi)))\right]$$

C PAIR-WISE CONTRASTIVE ESTIMATION

We first define a reward function $r(\cdot)$ to properly compare samples from two different sets or distributions.

$$r(x_t, x_s) = \tilde{r}(x_t) - \tilde{r}(x_s)$$

where $\tilde{r}$ is a reward function that assigns higher values to target samples than source samples

C.1 NON-RESIDUAL

If we define energy functions for each of them,

$$E_\theta(x) = -\log P_\theta(x) - \log Z(\theta)$$

Then the reward function becomes

$$r(x_t, x_s) = \log P_\theta(x_t) - \log P_\theta(x_s) = -(E_\theta(x_t) - E_\theta(x_s))$$

Then the loss function becomes

$$\begin{aligned}\mathcal{L}(\theta;\phi) &= -\mathbb{E}_{x_t,x_s}\left[\log\sigma(r(x_t,x_s))\right] \\ &= -\mathbb{E}_{x_t,x_s}\left[\log\sigma(-(E_\theta(x_t) - E_\theta(x_s)))\right]\end{aligned}$$

The gradient becomes

$$\begin{aligned}\nabla_\theta\mathcal{L}(\theta;\phi) &= -\mathbb{E}_{x_t,x_s}\left[\sigma(-r(x_t,x_s))\nabla_\theta r(x_t,x_s)\right] \\ &= \mathbb{E}_{x_t}\left[\sigma(E_\theta(x_t) - E_\theta(x_s))\left(\nabla_\theta E_\theta(x_t) - \nabla_\theta E_\theta(x_s)\right)\right]\end{aligned}$$

D USE OF LLMS

We used a large language model (ChatGPT) only as a general purpose assistive tool for minor editing tasks such as polishing sentences, correcting grammar and spelling and making small LaTeX table formatting adjustments. The LLM was not involved in research ideation, experimental design, data analysis, or substantive writing. All technical decisions, interpretations, and the writing of the core content were carried out entirely by the authors, who take full responsibility for the originality of the manuscript.