WHEN AND WHY DOES TEST-TIME ADAPTATION IMPROVE LABEL DIFFERENTIALLY PRIVATE MODELS?

Anonymous authors
Paper under double-blind review

ABSTRACT

Label differential privacy (LabelDP) protects sensitive training labels but often degrades accuracy. We study when test-time adaptation (TTA) can recover part of this lost utility. Since TTA uses only the released model and unlabeled inputs, it incurs no additional training-label privacy cost. We first show that symmetric randomized-response (RR) noise contracts the population label posterior toward uniform without changing the Bayes decision, biasing training toward a uniform-label objective with a privacy-dependent excess clean risk. We then characterize empirically when adaptation improves a label-private model. On clean CIFAR-10, TENT lowers the error of every LabelDP model, with gains growing from 0.87 to 3.62 points as privacy strengthens, while it increases the error of a matched non-private model. These improvements come with lower calibration error and vanish when the source model is near chance. Under common corruptions, the robustness benefits of TTA transfer to LabelDP-trained models, mostly through re-estimating batch-normalization (BN) statistics: across four TTA methods, entropy-based updates help only when combined with target-batch BN statistics. We evaluate on CIFAR-10/100, their corruption benchmarks, and ImageNet64 subsets, and identify small batches and label-correlated streams as failure conditions. With sufficiently large test batches that are not strongly label-skewed, TTA is a practical way to keep label-private models useful with or without distribution shift.

1 INTRODUCTION

Modern learning systems are increasingly trained in settings where the target labels encode sensitive information. In medical imaging, the image itself may be shareable while the diagnosis is private; in surveys and user analytics, covariates may be public or collected with consent while labels such as income, preference, or affiliation are sensitive; and in recommendation systems, interaction histories may be observable while the inferred rating or preference should be protected. These settings motivate *label differential privacy* (LabelDP) (Ghazi et al., 2021), a relaxation of example-level differential privacy in which neighboring datasets differ only in the label of one example. When only labels are private, LabelDP can offer better utility than mechanisms that privatize both inputs and labels.

The standard way to obtain LabelDP is to randomize labels before training, for example using randomized response and its prior-aware variants (Warner, 1965; Ghazi et al., 2021). However, this utility advantage does not eliminate the privacy–accuracy tradeoff: stronger label privacy requires more aggressive label perturbation, which weakens supervision and often produces models that are less accurate, less robust, and poorly calibrated. This raises a natural question that existing LabelDP training algorithms do not address: *after a label-private model has been trained, can we recover some of the lost utility at inference time without accessing any additional private labels?*

We answer this question through the lens of *test-time adaptation* (TTA). TTA methods adapt a deployed model using only unlabeled target inputs, often to improve robustness under distribution shift (Wang et al., 2021; Niu et al., 2022). At first glance, TTA and LabelDP address different problems: LabelDP protects sensitive labels during training, whereas TTA corrects distribution mismatch during inference. Our key observation is that they are highly compatible. Since TTA uses only unlabeled target inputs and applies after the private training mechanism, it is a post-processing operation with respect to the privatized labels. Therefore, adaptation cannot weaken the original LabelDP guarantee. At the same time, randomized response preserves which class is most likely in

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

the population but shrinks the margins, so label randomization pushes models toward high-entropy predictions, the quantity that entropy-based TTA acts on.

This paper studies the resulting *post-private adaptation* pipeline: train once under LabelDP, then adapt at inference using unlabeled test batches. We focus on two lightweight adaptation mechanisms: batch normalization (BN) adaptation, which updates test-time normalization statistics, and TENT, which minimizes prediction entropy by updating only the BN affine parameters (Wang et al., 2021). This design is intentionally minimal. It requires no source data at deployment, no retraining, no access to clean labels, and no additional privacy accounting for labels.

TTA improves both clean and corrupted performance of LabelDP models. On clean CIFAR-10, TENT reduces test error by 0.87–3.62 percentage points across all privacy budgets, with larger gains under stronger privacy, while it increases the error of a matched non-private control. Under common corruptions, TENT reduces mean corruption error by 17.39–26.49 points after LabelDP training, and the non-private control benefits at least as much. To interpret these trends, we analyze the two stages separately: symmetric RR biases private training toward a uniform-label objective with an excess clean risk that grows with privacy, and TTA descends an entropy objective on unlabeled inputs. Empirically, adaptation stops helping once privacy noise leaves the source model near chance.

Our contributions are as follows:

- We study a LabelDP-then-TTA pipeline and characterize when it improves accuracy across privacy budget, clean versus corrupted data, source model quality, batch size, and stream composition, without consuming additional label-privacy budget.
- We provide a two-stage analysis showing that symmetric RR contracts the population posterior toward uniform without changing the Bayes decision and biases training toward a uniform-label objective, with a clean-risk excess and BN-statistic sensitivity that grow with privacy.
- Empirically, we show that clean-data gains grow with the strength of label privacy and reverse on a matched non-private model. The improvements are accompanied by higher confidence and lower calibration error, and a prediction-flip analysis quantifies the resulting decision changes. A comparison of four TTA methods, including MEMO with and without target-batch statistics, shows that corruption gains come mainly from batch normalization.

2 BACKGROUND

Differential privacy and LabelDP. Differential privacy (DP) formalizes the requirement that the output distribution of a randomized algorithm be stable to the addition, removal, or replacement of a single individual record (Dwork et al., 2006; Abadi et al., 2016). A randomized algorithm $\mathcal{M}$ is (ε, δ)-DP if, for any adjacent datasets D, D' and measurable event S,

$$\Pr[\mathcal{M}(D) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(D') \in S] + \delta. \tag{1}$$

The parameter ε controls the privacy strength, while δ allows a small failure probability. In supervised learning, example-level DP treats two datasets as adjacent if they differ in an entire input–label pair. Label differential privacy (LabelDP) instead considers the setting where inputs are public or non-sensitive and only labels are private: two datasets are adjacent if they differ in the label of one example, while all inputs are fixed (Ghazi et al., 2021). This weaker adjacency relation is appropriate in applications where labels encode the sensitive attribute and can yield substantially greater utility than example-level DP.

Randomized response for label privacy. A canonical mechanism for LabelDP is randomized response (RR) (Warner, 1965). For a K-class problem with true label $y \in [K]$, the privatized label $\tilde{y}$ is sampled as

$$\Pr[\tilde{y} = j \mid y] = \begin{cases} \dfrac{e^{\varepsilon}}{e^{\varepsilon} + K - 1}, & j = y, \\ \dfrac{1}{e^{\varepsilon} + K - 1}, & j \neq y. \end{cases} \tag{2}$$

Thus, a smaller ε increases the probability of replacing the true label with an incorrect one.

Prior-aware LabelDP methods such as *RRWithPrior* and *RRTop-k* improve the utility of this mechanism by restricting randomization to a likely subset of labels (Ghazi et al., 2021). These methods consider the set Y_k of k labels with the largest prior probabilities. If the true label y belongs to Y_k, then we use the standard RR in eq. 2. Otherwise, we uniformly sample a label from Y_k. *Label Privacy Multi-Stage Training* (LP-MST) introduces a multi-stage training framework that refines priors progressively from a uniform prior. At each stage, the priors are estimated from the model predictions from the previous stage, and aid in selecting the ideal Y_k for RRWithPrior. LP-MST partitions the training set independently of the labels and privatizes each label once, while later stages reuse earlier privatized labels; like RR, it is ϵ-LabelDP.

Test-time adaptation. TTA adapts a trained model using unlabeled target data during inference. Unlike domain adaptation methods that require source data or labeled target examples, fully test-time methods operate only on the deployed model and incoming test batches. We center our study on two lightweight TTA methods. Batch normalization (BN) adaptation recomputes normalization statistics on target batches, thereby correcting shifts between training and test feature distributions. Test-Time Entropy Minimization (TENT) adapts the affine parameters of batch-normalization layers by minimizing prediction entropy on unlabeled target inputs (Wang et al., 2021). Given a model f_θ with predictive distribution $p_\theta(y \mid x)$, TENT minimizes $-\frac{1}{|B|}\sum_{x\in B}\sum_{k=1}^{K} p_\theta(k \mid x)\log p_\theta(k \mid x)$, where B is an unlabeled test batch; all other weights stay frozen. We also provide additional experiments for SAR (Niu et al., 2023), which adds sharpness-aware, reliability-filtered entropy minimization on top of target-batch statistics, and MEMO (Zhang et al., 2022), which minimizes marginal entropy over test-input augmentations; we evaluate MEMO both with source and with test-batch normalization statistics. These methods let us separate the effect of updating normalization statistics from the effect of entropy minimization.

3 Related Work

Label differential privacy. Differential privacy (DP) protects against the influence of individual training records (Dwork et al., 2006; Abadi et al., 2016). When only labels are sensitive, label differential privacy (LabelDP) offers a less restrictive alternative by fixing inputs and protecting changes to training labels (Ghazi et al., 2021). Randomized response and prior-aware variants remain the standard mechanisms for LabelDP, while recent work has explored alternatives based on semi-supervised learning, projection-based training, vector-valued privatization, and auditing label leakage (Malek et al., 2021; Tang et al., 2022; Ghazi et al., 2024; Zhao et al., 2025; Busa-Fekete et al., 2024). Closely related, Das et al. (2025) retrain on model-predicted hard labels at no additional privacy cost. Our work instead studies deployment-time adaptation on unlabeled target inputs, which can respond to target covariate shift without retraining on the source data. We include an exploratory empirical comparison in Appendix C.5. Rather than proposing a new LabelDP mechanism, we study whether a model trained under LabelDP can recover utility at deployment time.

Test-time adaptation. Test-time adaptation (TTA) updates a deployed model using unlabeled target data to handle distribution shift (Wang et al., 2021). Subsequent methods improve robustness in single-sample, continual, non-i.i.d., and dynamic test streams through augmentation, sample filtering, memory, robust normalization, or stability-aware objectives (Zhang et al., 2022; Niu et al., 2022; Wang et al., 2022; Gong et al., 2022; Yuan et al., 2023; Niu et al., 2023). We center our paper on BN adaptation and TENT because they are lightweight and source-free.

Private training with adaptive deployment. Most private learning methods treat inference as fixed, while most TTA methods study robustness without privacy constraints. LabelDP creates a clean interface between these areas: the private information is the training label, while TTA uses only unlabeled test inputs, so adaptation is post-processing (Remark 1). Li et al. (2026) address a different setting, privatizing the adaptation updates when the test inputs themselves are sensitive. We assume test inputs are not sensitive; settings where both are sensitive would require combining the two mechanisms. We study this interface empirically and theoretically, showing when deployment-time adaptation recovers utility lost to label noise.

4 METHOD

We study a two-stage pipeline that combines LabelDP during training with TTA at inference: (I) **LabelDP training**: supervised training with RR, RRWithPrior, or LP-MST to protect the sensitive labels, and (II) **Test-Time Adaptation**: adapting the released LabelDP model to unlabeled target data, which preserves privacy (Remark 1). Algorithm 1 in Appendix A summarizes the pipeline.

Consider K-class classification with data $(x, y) \sim \mathcal{D}$ on $\mathcal{X} \times [K]$. A network f_θ produces logits, and the predicted class probabilities are given by the softmax output $p_\theta(x) = \mathrm{softmax}(f_\theta(x))$. We decompose its parameters as $\theta = (\phi, \psi)$, where $\psi = (\gamma, \beta)$ are the BN affine parameters TENT updates and ϕ is frozen at test time. Symmetric RR replaces each label y with $\tilde{y}$ such that $\Pr[\tilde{y} = y] = \alpha := \frac{e^\varepsilon}{e^\varepsilon + K - 1}$. Define the *signal strength*

$$\rho(\varepsilon) \;:=\; \frac{e^\varepsilon - 1}{e^\varepsilon + K - 1} \;\in\; [0, 1], \tag{3}$$

so that $\rho \to 0$ as $\varepsilon \to 0$ (pure noise) and $\rho \to 1$ as $\varepsilon \to \infty$ (no noise). We write $R(\theta) := \mathbb{E}[-\log p_\theta^{(y)}(x)]$ for the clean cross-entropy risk, $\tilde{R}(\theta)$ for the population noisy-label risk under symmetric RR, and $R_{\mathrm{unif}}(\theta) := \mathbb{E}_x[\frac{1}{K}\sum_j - \log p_\theta^{(j)}(x)]$ for the uniform-label risk.

The pipeline is sequential. Private training produces the source model, and deployment then updates only the BN parameters on incoming unlabeled batches B_t:

$$\hat{\theta} = (\hat{\phi}, \psi_0) = \mathrm{Train}_\varepsilon(D), \qquad \psi_{t+1} = \mathrm{Update}(\psi_t, B_t; \hat{\phi}). \tag{4}$$

Training does not anticipate adaptation, so we analyze the two stages separately: Section 4.1 characterizes how symmetric RR biases the population training objective, and Section 4.2 analyzes the entropy objective that TENT descends. The results are population-level and local. They identify mechanisms and regimes that we then test empirically (Section 4.3); they are not guarantees for the finite-sample, Adam-based implementation. Proofs, assumptions, and notation are in Appendix F.

4.1 PRIVATE TRAINING: HOW SYMMETRIC RR BIASES THE OBJECTIVE

Proposition 1 (Posterior contraction under symmetric RR)**.** *Let $\pi(x)$ be the clean class posterior, $\pi_\varepsilon^{(j)}(x) := Pr[\tilde{y} = j \mid x]$ the posterior of the privatized label, and u the uniform distribution over K classes. Symmetric randomized response gives*

$$\pi_\varepsilon(x) = \rho(\varepsilon)\pi(x) + \big(1 - \rho(\varepsilon)\big)u. \tag{5}$$

For $\rho(\varepsilon) > 0$, $\pi_\varepsilon(x)$ preserves a unique Bayes argmax and contracts every class margin by $\rho(\varepsilon)$.

This identity is exact and needs no smoothness or PL assumption: symmetric RR contracts the population posterior margins without changing the Bayes decision rule. In finite trained networks, label privatization can also affect estimation, optimization, and the learned representation. Because TTA changes shared normalization statistics and parameters, it can alter class rankings, not only confidence; Sections 5 and 6 characterize when these changes improve accuracy and how they relate to calibration. Our experiments use prior-aware LP-MST, for which Proposition 1 is not exact.

Lemma 2 (Gradient Bias Decomposition)**.** *For every θ,*

$$\nabla\tilde{R}(\theta) \;=\; \rho(\varepsilon)\,\nabla R(\theta) \;+\; \big(1 - \rho(\varepsilon)\big)\,\nabla R_{\mathrm{unif}}(\theta). \tag{6}$$

Lemma 2 states that symmetric-RR training optimizes a mixture of the clean risk and a uniform-label risk; equivalently, it is cross-entropy against the smoothed target $\rho e_y + (1-\rho)u$, as in label smoothing. A smaller privacy budget means a larger uniform-label component, which pushes the population optimum toward less confident predictions, consistent with Figure 1.

Assumption 1 (Clean PL condition and bounded gradient bias)**.** *The clean risk R satisfies the Polyak–Lojasiewicz (PL) inequality with constant $\mu_R > 0$: $R(\theta) - R^* \le \frac{1}{2\mu_R}\|\nabla R(\theta)\|^2$, where $R^* := \inf_\vartheta R(\vartheta)$. Moreover, $\Delta_g := \sup_\theta \|\nabla R_{\mathrm{unif}}(\theta) - \nabla R(\theta)\| < \infty$.*

Theorem 3 (Clean-risk excess from noisy-risk stationarity)**.** *Under Assumption 1, for an exact stationary point $\tilde{\theta}^*$ of the noisy risk $\tilde{R}$ (i.e., $\|\nabla \tilde{R}(\tilde{\theta}^*)\| = 0$),*

$$R(\tilde{\theta}^*) - R(\theta^*) \;\le\; \frac{(1-\rho(\varepsilon))^2}{2\mu_R}\,\Delta_g^2, \tag{7}$$

where θ^* is a clean-risk minimizer.

Theorem 3 bounds the excess clean risk of the unadapted model at a population stationary point by a term scaling as $(1-\rho(\varepsilon))^2$. The bound saturates as $\varepsilon \to 0$ because $1-\rho(\varepsilon) \le 1$, rather than diverging.

To relate this excess risk to BN statistics, we additionally assume local quadratic growth, which links excess risk to the distance from the set of clean minimizers. Let $\Theta_* := \arg\min_\theta R(\theta)$ and $\mathrm{dist}(\theta, \Theta_*) := \inf_{\vartheta \in \Theta_*} \|\theta - \vartheta\|$. For BN layer l, define the population pre-BN mean $m_l(\theta) := \mathbb{E}[h_l(X;\theta)]$. Equivalently, with class prior $\pi_k = \Pr(Y=k)$ and class-conditional mean $\mu_l^{(k)}(\theta) := \mathbb{E}[h_l(X;\theta) \mid Y=k]$, $m_l(\theta) = \sum_{k=1}^K \pi_k \mu_l^{(k)}(\theta)$.

Assumption 2 (Quadratic growth and BN Lipschitzness)**.** *On the region of interest, R satisfies quadratic growth with constant $\nu_R > 0$:*

$$R(\theta) - R^* \ge \frac{\nu_R}{2}\,\mathrm{dist}(\theta, \Theta_*)^2.$$

Moreover, for each BN layer l, m_l is locally Lipschitz along the segment connecting the LabelDP-trained parameter to a nearest point in Θ_*: there is $G_l < \infty$ such that

$$\sup_{s\in[0,1]} \Big\| Jm_l(\bar{\theta} + s(\tilde{\theta}^* - \bar{\theta})) \Big\|_{\mathrm{op}} \le G_l,$$

where $\bar{\theta} \in \Theta_*$ is a nearest clean minimizer to the noisy-risk stationary point $\tilde{\theta}^*$ and Jm_l is the Jacobian of m_l with respect to θ.

Proposition 4 (BN Statistics Perturbation)**.** *Under Assumption 1 and Assumption 2, the shift in the BN running mean is bounded by:*

$$\big\| m_l(\tilde{\theta}^*) - m_l(\bar{\theta}) \big\| \;\le\; \frac{G_l(1-\rho(\varepsilon))}{\sqrt{\mu_R \nu_R}}\,\Delta_g \tag{8}$$

Proposition 4 is a limited sensitivity result: it bounds how far the population BN mean of the private model can lie from that of a nearest clean minimizer, with a factor $1-\rho(\varepsilon)$ that grows as privacy strengthens. This differs from the mismatch $\hat{m}_l - \mathbb{E}_{X\sim\mathcal{D}_{\mathrm{test}}}[h_l(X;\hat{\theta})]$ between stored and target statistics that test-time normalization addresses; we evaluate the latter empirically.

4.2 Adaptation: The Entropy Objective

Given the source model with frozen parameters $\hat{\phi}$, TENT adapts only the BN affine parameters $\psi = (\gamma, \beta)$ by minimizing the prediction entropy on unlabeled test data. Let $\mathcal{D}_{\mathrm{test}}$ denote the test distribution over $\mathcal{X}$. We analyze the pointwise objective

$$H(\psi) := \mathbb{E}_{x\sim\mathcal{D}_{\mathrm{test}}}\left[-\sum_{c=1}^K p^{(c)}_{\hat{\phi},\psi}(x) \log p^{(c)}_{\hat{\phi},\psi}(x) \right], \tag{9}$$

where $p_{\hat{\phi},\psi}(x) = \mathrm{softmax}(f_{\hat{\phi},\psi}(x))$; with batch statistics, the implemented predictions also depend on the other inputs in the batch.

Remark 1 (Privacy preservation). Let $M(X,Y)$ be an ε-LabelDP training mechanism with public training covariates X, and let U be adaptation inputs that carry no information from the protected labels Y. Any adaptation rule A that uses only $M(X,Y)$, X, and U, including BN adaptation, TENT, SAR, and MEMO, is post-processing, so $A(M(X,Y), X, U)$ remains ε-LabelDP for any batch size, batch composition, or stream order. It provides no privacy guarantee for U.

Let ψ_0 be the BN parameters from LabelDP training, $\psi^\dagger := \arg\min_\psi H(\psi)$, and $\Delta H_0(\varepsilon) := H(\psi_0) - H(\psi^\dagger)$ be the initial entropy gap.

Theorem 5 (TENT Objective Convergence)**.** *Assume H is L_H-smooth and satisfies the PL inequality with constant $\mu_H > 0$ on a region containing the iterates. After T steps of gradient descent on H with step size $\eta = 1/L_H$, the prediction entropy converges geometrically:*

$$H(\psi_T) - H(\psi^\dagger) \;\le\; \left(1 - \frac{\mu_H}{L_H}\right)^T \Delta H_0(\varepsilon) \;\le\; \frac{L_H}{L_H + \mu_H T} \Delta H_0(\varepsilon). \tag{10}$$

Theorem 5 concerns gradient descent on a fixed objective, whereas our experiments take one Adam step per batch on a changing stream, so we use it to describe the entropy dynamics rather than as a convergence guarantee for the implementation. Low entropy also does not guarantee low classification error, so linking adaptation to accuracy requires an explicit *error–entropy transfer* assumption.

Corollary 6 (Error Bound under Error–Entropy Transfer)**.** *Under the assumptions of Theorem 5, suppose there exists an error–entropy transfer constant $\kappa > 0$ such that the relevant iterates satisfy $\mathrm{err}(\psi) - \mathrm{err}(\psi^\dagger) \le \kappa(H(\psi) - H(\psi^\dagger))$. Then*

$$\mathrm{err}(\psi_T) - \mathrm{err}(\psi^\dagger) \;\le\; \kappa \left(1 - \frac{\mu_H}{L_H}\right)^T \Delta H_0(\varepsilon) \;\le\; \kappa \frac{L_H}{L_H + \mu_H T} \Delta H_0(\varepsilon). \tag{11}$$

The transfer condition is an assumption, not a consequence of calibration, and the bound is relative to $\psi^\dagger$, which need not be a good classifier; it does not by itself establish $\mathrm{err}(\psi_T) < \mathrm{err}(\psi_0)$.

Remark 2 (Local alignment). A before-versus-after statement is available locally. If R is L_R-smooth in ψ, a step along a direction d satisfies $R(\psi - \eta d) - R(\psi) \le -\eta \langle \nabla_\psi R(\psi), d\rangle + \frac{L_R \eta^2}{2} \|d\|^2$. For $d = \nabla_\psi H$, positive alignment $\langle \nabla_\psi R, \nabla_\psi H\rangle > 0$ ensures that a sufficiently small entropy step lowers clean cross-entropy, whereas negative alignment permits confidently-wrong updates. We do not measure these gradients; the higher error of the non-private control after TENT (Section 5.1) is consistent with, but does not demonstrate, negative alignment.

4.3 Combining the Stages and Empirical Regimes

Sections 4.1 and 4.2 establish the following. Symmetric RR contracts population margins without changing the Bayes rule (Proposition 1) and biases the population objective toward a uniform-label risk (Lemma 2). Under local assumptions, this induces a clean-risk excess and a BN-mean sensitivity that scale with $1 - \rho(\varepsilon)$ (Theorem 3, Proposition 4), and entropy descent on the BN affine parameters converges (Theorem 5). Adding the bounds of Theorems 3 and 5 gives a combined surrogate bound (Theorem 7, Appendix F.9). It sums the clean-risk excess of the *unadapted* model and the entropy residual after adaptation; it is not the adapted predictor's clean risk and does not guarantee improvement over the source.

The more useful output of the analysis is the signal strength $\rho(\epsilon)$, which governs how strongly privatization biases training and organizes our experiments into three regimes (Sections 5.1 and 5.4). (a) At large ϵ, $\rho(\epsilon) \approx 1$ and TTA yields small clean-data gains. (b) At moderate ϵ, the source retains useful representations and TTA yields its largest gains. (c) At very small ϵ, the source is near chance and adaptation does not help. Because $\rho(\epsilon)$ decreases with K, larger label spaces reach regime (c) at larger ϵ. The boundaries between regimes are set empirically, since the bounds do not determine how ΔH_0 or the transfer constants vary with ϵ.

The analysis also points to a mechanism for the clean-data gains: adaptation changes the decisions of low-confidence, near-boundary predictions of the private model. Section 6 examines this through calibration and prediction flips. That evidence is consistent with the mechanism but not causal, and it does not separate privacy-specific effects from those of softened supervision more generally (label smoothing, mixup).

5 Experiments

Experimental setup. We evaluate the post-private adaptation pipeline on CIFAR-10 and CIFAR-100 (Krizhevsky & Hinton, 2009), and use CIFAR-10-C/CIFAR-100-C (Hendrycks & Dietterich, 2019) to test robustness to common corruptions. Unless stated otherwise, the source model is a pre-activation ResNet-18 (He et al., 2016a;b) with batch-normalization layers, trained under LabelDP

Method	Label-private $\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = 4$	$\varepsilon = 8$	Non-private $\varepsilon = \infty$
CIFAR-10 (clean)					
No TTA	45.76	16.10	8.59	7.32	**8.44**
+ BN	**42.00** −3.76	14.96 −1.14	8.54 −0.05	7.59 +0.27	10.46 +2.02
+ TENT	42.14 −3.62	**13.47** −2.63	**7.51** −1.08	**6.45** −0.87	9.23 +0.79
CIFAR-100 (clean)					
No TTA	98.41	93.71	73.23	34.31	**33.65**
+ BN	**98.37** −0.04	92.75 −0.96	69.11 −4.12	31.17 −3.14	35.13 +1.48
+ TENT	98.43 +0.02	**92.18** −1.53	**64.90** −8.33	**29.34** −4.97	33.68 +0.03
CIFAR-10-C (severity 5)					
No TTA	68.32	50.30	44.97	44.17	55.21
+ BN	**50.43** −17.89	28.93 −21.37	23.79 −21.18	23.28 −20.89	27.78 −27.43
+ TENT	50.93 −17.39	**23.93** −26.37	**18.58** −26.39	**17.68** −26.49	**24.89** −30.32
CIFAR-100-C (severity 5)					
No TTA	98.65	95.01	88.16	73.62	77.77
+ BN	**98.36** −0.29	93.61 −1.40	78.38 −9.78	51.29 −22.33	55.17 −22.60
+ TENT	98.40 −0.25	**93.03** −1.98	**75.21** −12.95	**46.54** −27.08	**52.71** −25.06

Table 1: Test error (%) of LP-MST ResNet-18 models with and without TTA; small numbers give the change from No TTA. Corrupted: mean of 15 corruptions at severity 5. $\varepsilon = \infty$: matched non-private control (single run). Five-seed standard deviations for CIFAR-10 LabelDP models are in Table 10.

Method	Label-private $\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = 4$	$\varepsilon = 8$
CIFAR-10 (clean)				
No TTA	45.76	16.10	8.59	7.32
+ MEMO	46.31	16.19	8.54	7.41
+ BN	**42.00**	14.96	8.54	7.59
+ MEMO+BN	42.03	14.37	8.29	6.92
+ SAR	42.19	14.90	8.50	7.57
+ TENT	42.14	**13.47**	**7.51**	**6.45**
CIFAR-10-C (severity 5)				
No TTA	68.32	50.30	44.97	44.17
+ MEMO	69.10	51.81	46.06	45.17
+ BN	50.43	28.93	23.79	23.28
+ MEMO+BN	**50.11**	27.09	22.46	21.42
+ SAR	50.45	28.76	23.56	23.04
+ TENT	50.93	**23.93**	**18.58**	**17.68**

Table 2: CIFAR-10 test error (%) of TTA methods on the models of Table 1. MEMO keeps source BN statistics; MEMO+BN uses test-batch statistics.

with two-stage LP-MST (Ghazi et al., 2021) (Section 2). We follow the training recipe of Ghazi et al. (2021): mini-batch SGD for 200 epochs with a piecewise-constant learning-rate schedule, cross-entropy loss, and mixup regularization (Zhang et al., 2018; Carratino et al., 2022) for noisy labels. The LP-MST split allocates more data to the first stage when ϵ is small.

At test time, we compare the unadapted LabelDP model with two source-free online adaptation rules, and add SAR and MEMO variants in Section 5.3. BN adaptation updates only test-batch normalization statistics. TENT minimizes prediction entropy on unlabeled target batches while updating only the BN affine parameters; all other weights are frozen. We use online adaptation with batch size 200 and one Adam (Kingma & Ba, 2015) step per batch with learning rate 10^{-3}, following the official TENT implementation; adaptation carries over across batches and is reset for each corruption type. Clean results use the standard CIFAR test sets, while corrupted results report mean error across the 15 corruption types at severity 5 unless otherwise noted. We report changes in error as absolute percentage points and compute ECE with 15 bins (Guo et al., 2017).

5.1 TTA IMPROVES CLEAN ACCURACY OF LABELDP-TRAINED MODELS

Table 1 shows that TTA lowers the clean test error of LabelDP-trained models, with improvements that grow as label privacy becomes stronger. TENT reduces error at every finite privacy budget, from 0.87 points at $\epsilon = 8$ to 3.62 points at $\epsilon = 1$, while BN helps mainly under strong privacy (3.76 points at $\epsilon = 1$). In contrast, both methods increase the error of a matched non-private control ($\epsilon = \infty$) trained with the same architecture, optimizer, and schedule, by 2.02 points with BN and 0.79 points with TENT. Within this protocol, the clean-data benefit is therefore **associated with label privatization**, rather than a generic effect of TTA on this architecture. On CIFAR-100, TENT reduces error by 8.33 points at $\epsilon = 4$ and 4.97 points at $\epsilon = 8$, while the non-private control is essentially unchanged (+0.03 points; Table 1). The smaller CIFAR-100 gains at $\epsilon \leq 2$ correspond to regime (c) of Section 4.3, where the source is near chance (Section 5.4). We analyze how TTA changes prediction confidence and calibration in Section 6.

5.2 TTA IMPROVES CORRUPTION ROBUSTNESS OF LABELDP-TRAINED MODELS

TTA also improves the robustness of LabelDP-trained models to corrupted data. On CIFAR-10-C at severity 5 (Table 1), BN reduces error by 17.89–21.37 points across all privacy budgets, and TENT reduces it by 17.39–26.49 points, outperforming BN for $\epsilon \geq 2$. On CIFAR-100-C, the gains are large once the source model is usable, with TENT reducing error by 12.95 points at $\epsilon = 4$ and 27.08 points

at $\epsilon = 8$. Unlike the clean setting, these gains do not depend on label privacy: on CIFAR-10-C, the non-private model benefits at least as much, with BN and TENT reducing its error by 27.43 and 30.32 points. Substantial corruption-robustness gains thus persist after LabelDP training in the evaluated batched settings, at no additional label-privacy cost. Per-corruption results (Appendix D) show that TENT lowers error at every severity and privacy level, more so at higher severity.

5.3 When Does TTA Recover Utility?

To separate the effect of normalization from that of entropy minimization, we compare TTA methods on the same LabelDP checkpoints (Table 2). For MEMO, toggling the BN statistics gives a two-factor comparison. With source statistics, MEMO's entropy update stays close to the unadapted model on clean data and is slightly worse under corruption; with test-batch statistics, the same update improves on BN adaptation alone by 0.3–1.9 points on CIFAR-10-C. All rules that use target-batch statistics cut CIFAR-10-C error by 17.4–26.5 points, and SAR stays within 0.25 points of BN in every setting. Most of the corruption gain therefore comes from re-estimating BN statistics, and entropy updates add a method-dependent refinement on top: TENT adds 5.0–5.6 points on CIFAR-10-C and 1.0–1.5 points on clean data at $\epsilon \geq 2$, but nothing at $\epsilon = 1$, where MEMO+BN is the best rule under corruption. Entropy minimization thus refines the adapted model when the source is reasonably accurate, but it cannot compensate for a weak source.

The improvements rely on test batches that give reliable statistics, a known issue in TTA (Niu et al., 2023; Gong et al., 2022) that LabelDP models inherit (Table 3). The evaluated online BN and TENT configurations fail at $B = 1$, with near-chance error. Corruption gains (averaged over severities 1–5 in this sweep) appear from $B = 8$ for $\epsilon \geq 2$, clean gains for TENT require $B = 64$, and at $\epsilon = 1$ TENT beats the source on clean data only at our main batch size of 200. Stronger privacy also makes TENT more sensitive to small batches. At $B = 8$ on clean data, it is 23.2 points worse than BN at $\epsilon = 1$ but within 2.3 points at $\epsilon \geq 2$. Stream composition matters as well (Table 4). At $\epsilon = 8$ and $B = 64$, TENT improves on the source for homogeneous, gradual, i.i.d., and domain-correlated streams. When labels are concentrated within batches, however, the batch estimate of the BN mean m_l is taken under a skewed class prior (Section 4.1), and TENT and BN fail badly (66.07% and 62.96% vs. 23.86% source error). Online predictions also depend on arrival order: on mixed i.i.d. streams, 13.9% of samples change correctness across three orders under TENT (Table 5). When test batches are small or their label composition is unknown, source inference is safer (Appendix C).

5.4 Source Model Quality and Scale

TTA does not remove the privacy-utility tradeoff. It can only recover utility that the source model still encodes, so its benefit depends on source quality (Zhao et al., 2023). When the source is near chance, TTA offers little improvement, as with the ResNet-18 on CIFAR-100 at $\epsilon \leq 2$ (93.71% and 98.41% error). Under the symmetric-RR model, $\rho(\epsilon)$ shrinks with the number of classes ($\rho \approx 0.06$ for $K = 100, \epsilon = 2$ versus $\rho \approx 0.15$ for $K = 10, \epsilon = 1$), so larger label spaces reach this regime at larger ϵ. A stronger source model restores the benefit. With a CLIP (Radford et al., 2021) prior for LP-MST on CIFAR-100, BN and TENT reduce error at every privacy budget, with TENT recovering up to 7.40 points on clean data and 21.36 points on CIFAR-100-C (Table 7). To test scale, we train an LP-MST linear probe on a frozen self-supervised DINO (Caron et al., 2021) ResNet-50 for 10- and 100-class ImageNet64 (Chrabaszcz et al., 2017) subsets (Table 6). Because the backbone is frozen, this setting tests scale and corruption adaptation rather than privacy-induced BN drift. At $\epsilon \geq 2$, normalization removes 12.75–16.81 points of corruption error for $K = 10$ and 3.59–12.21 points for $K = 100$, while at $K = 100$ and $\epsilon = 1$ the source is near chance and no method helps. As in Section 5.3, the gain comes from normalization rather than the entropy step.

6 Analysis

6.1 Confidence and Calibration under TTA

To examine how TTA changes predictions, we measure prediction confidence, the softmax probability of the predicted class, and how well it matches correctness. At $\epsilon = 1$, the source LabelDP model has a broad, low-confidence distribution centered around 0.4–0.6 (Figure 1, left), consistent with the

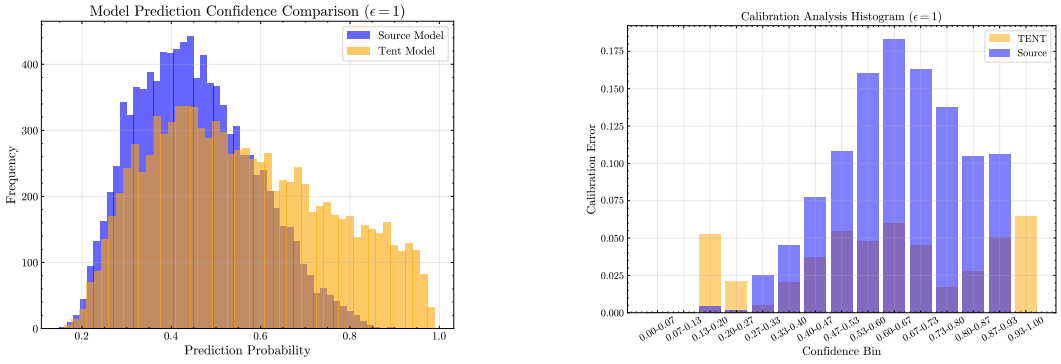


Figure 1: Confidence (left) and per-bin calibration error $|\mathrm{acc} - \mathrm{conf}|$ (right) of the $\epsilon = 1$ LabelDP model on clean CIFAR-10, before (source) and after TENT, which lowers ECE from 0.089 to 0.037.

uniform-label bias of Lemma 2. Using only unlabeled test inputs, TENT shifts predictions toward higher confidence, especially into the 0.6–0.9 range. This added confidence does not come at the cost of calibration: TENT lowers the calibration error in most bins, particularly across 0.4–0.9, which are also the bins where confidence rises most (Figure 1, right). ECE falls from 0.089 to 0.037. The same pattern holds for $\epsilon \in \{2, 4, 8\}$ (Figures 3 and 4 in Appendix E). The source model becomes more confident as the privacy budget grows, yet TENT still shifts predictions right and lowers ECE to 0.020–0.046. Source ECE, however, is not monotone in ϵ (0.089 at $\epsilon = 1$ versus 0.24–0.27 at $\epsilon \geq 2$), so calibration error alone does not predict the size of the accuracy gain. Sharpening alone also cannot explain the gains: temperature scaling raises confidence without changing any top-1 prediction. Accuracy can improve only because adaptation, through shared BN statistics and affine parameters, also changes class rankings, which the prediction flips below quantify.

6.2 Prediction Flips

Net accuracy hides how many individual predictions TTA changes. For each clean CIFAR-10 test image, we record whether adaptation turns a wrong prediction correct (W→C) or a correct one wrong (C→W) (Table 8 and Figure 2, Appendix C.4). The balance tracks label privacy: for TENT, the ratio of W→C to C→W falls from 2.0 at $\epsilon = 1$ (739 vs. 377) to 1.5 at $\epsilon = 8$ and below one on the matched non-private control (254 vs. 333), matching its accuracy loss in Table 1. Adaptation also changes 17.5–18.7% of predictions at $\epsilon = 1$, which matters for consistent per-sample decisions.

7 Conclusion

LabelDP training and test-time adaptation work well together: because adaptation uses only the released model and unlabeled inputs, it improves utility without weakening the training-label privacy guarantee. On clean CIFAR-10, TTA reduces error by up to 3.76 points with BN adaptation and 3.62 points with TENT. These gains grow with the strength of label privacy, reverse on the matched non-private control, and accompany higher confidence and lower calibration error. Under corruption, TENT reduces CIFAR-10-C error by up to 26.49 points, mostly through re-estimating BN statistics, a benefit shared by non-private models. In the evaluated settings, adaptation is thus a useful deployment tool for label-private models when the source retains useful accuracy and test batches are large and not strongly label-skewed. Promising next steps include LabelDP-aware adaptation that uses the known $\rho(\epsilon)$, adapt-then-freeze deployment for single queries, controls that separate label privacy from softened supervision, other LabelDP mechanisms, and private TTA (Li et al., 2026) for sensitive test inputs (scope in Appendix B).

AI USE STATEMENT

We used generative AI tools as writing, reviewing, and coding assistants. We used them to improve grammar and word choice; to restructure the manuscript and revise how theoretical and empirical claims are stated, including their assumptions, scope, and limitations; to review the manuscript for internal consistency and to suggest controls and analyses; to help understand technical concepts and locate related work, which we verified against the original sources; and to write and debug code. We did not use generative AI tools to generate synthetic data. Dataset cleaning, translation, and qualitative data analysis are not applicable to this work. The authors checked every AI suggestion before including it. We manually checked every cited reference against its official record, and all AI-assisted code was reviewed and tested by the authors before use. We take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies how test-time adaptation can improve the utility of models trained under label differential privacy. All experiments use publicly available benchmark datasets, and no new data was collected from human subjects. The intended positive impact is to make privacy-preserving learning more practical when labels encode sensitive information, such as medical diagnoses, user preferences, or demographic attributes, since adaptation on unlabeled inputs can improve accuracy and robustness without spending additional label-privacy budget. Improved utility can also increase deployment of private models in high-stakes settings, which carries risk if a model is used without adequate validation or under distribution shifts unlike those evaluated here. Entropy-minimization adaptation can produce overconfident predictions or degrade under difficult, non-i.i.d., or adversarial streams, so deployment should include safeguards against uncontrolled adaptation, and application-specific evaluation. Our privacy claim is limited to the LabelDP setting, where only training labels are private and adaptation uses unlabeled inputs. If test inputs are sensitive, adaptation still preserves the training-label guarantee but provides no privacy for the test inputs, which would require separate accounting. This work should not be read as eliminating the privacy-utility tradeoff, but as showing that deployment-time adaptation can recover some lost utility while preserving the original label-privacy guarantee under the stated assumptions.

REPRODUCIBILITY STATEMENT

We describe the training and adaptation setup in Section 5, including the backbones, the LP-MST configuration, the optimizer and learning-rate schedule, the adaptation batch size, step count, and learning rate, and the ECE binning. All theoretical claims are stated with their assumptions in Section 4, and complete proofs, the full set of assumptions, and a notation table are given in Appendix F. The datasets are CIFAR-10, CIFAR-100, their common-corruption variants CIFAR-10-C and CIFAR-100-C, and ImageNet64 class subsets, with the data-processing and corruption protocols described in Section 5 and the appendix. Anonymous code is available at the link in Appendix A, and the compute used is listed in Appendix G. Appendix D.1 reports five-seed standard deviations for the CIFAR-10 LabelDP clean and corrupted results. The batch-size sweep and ImageNet64 results average three seeds, the stream experiments use three arrival orders, and the remaining experiments, including the non-private controls, are single runs.

REFERENCES

Martin Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pp. 308–318, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450341394. doi: 10.1145/2976749.2978318.

Róbert István Busa-Fekete, Travis Dick, Claudio Gentile, Andrés Muñoz Medina, Adam Smith, and Marika Swanberg. Auditing privacy mechanisms via label inference attacks. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

Luigi Carratino, Moustapha Cissé, Rodolphe Jenatton, and Jean-Philippe Vert. On mixup regularization. *Journal of machine learning research*, 23(325):1–31, 2022.

Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819*, 2017.

Rudrajit Das, Inderjit S. Dhillon, Alessandro Epasto, Adel Javanmard, Jieming Mao, Vahab Mirrokni, Sujay Sanghavi, and Peilin Zhong. Retraining with predicted hard labels provably increases model accuracy. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/das25b.html.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, pp. 265–284. Springer, 2006.

Badih Ghazi, Noah Golowich, Ravi Kumar, Pasin Manurangsi, and Chiyuan Zhang. Deep learning with label differential privacy. In *Advances in Neural Information Processing Systems*, volume 34, pp. 27131–27145, 2021.

Badih Ghazi, Yangsibo Huang, Pritish Kamath, Ravi Kumar, Pasin Manurangsi, and Chiyuan Zhang. LabelDP-Pro: Learning with label differential privacy via projections. In *International Conference on Learning Representations*, 2024.

Taesik Gong, Jongheon Jeong, Taewoo Kim, Yewon Kim, Jinwoo Shin, and Sung-Ju Lee. Robust continual test-time adaptation against temporal correlation. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pp. 630–645, 2016b.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://openreview.net/forum?id=HJz6tiCqYm.

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 795–811. Springer, 2016.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, Toronto, ON, Canada, 2009.

Zefeng Li, Qiaoyue Tang, Mathias Lécuyer, and Evan Shelhamer. Private and stable test-time adaptation with differential privacy. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Mani Malek, Ilya Mironov, Karthik Prasad, Igor Shilov, and Florian Tramèr. Antipodes of label differential privacy: PATE and ALIBI. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*, pp. 16888–16905. PMLR, 2022.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. In *International Conference on Learning Representations*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Xinyu Tang, Milad Nasr, Saeed Mahloujifar, Virat Shejwalkar, Liwei Song, Amir Houmansadr, and Prateek Mittal. Machine learning with differentially private labels: Mechanisms and frameworks. *Proceedings on Privacy Enhancing Technologies*, 2022(4):332–350, 2022.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021.

Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7201–7211, 2022.

Stanley L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.

Longhui Yuan, Binhui Xie, and Shuang Li. Robust test-time adaptation in dynamic scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15922–15932, 2023.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

Marvin Zhang, Sergey Levine, and Chelsea Finn. MEMO: Test time robustness via adaptation and augmentation. In *Advances in Neural Information Processing Systems*, volume 35, pp. 38629–38642, 2022.

Hao Zhao, Yuejiang Liu, Alexandre Alahi, and Tao Lin. On pitfalls of test-time adaptation. In *International Conference on Machine Learning*, 2023.

Puning Zhao, Jiafei Wu, Zhe Liu, Li Shen, Zhikun Zhang, Rongfei Fan, Le Sun, and Qingming Li. Enhancing learning with label differential privacy by vector approximation. In *International Conference on Learning Representations*, 2025.

Algorithm 1 LabelDP Training with Test-Time Adaptation

Require: Training data $\mathcal{D}$, privacy budget ε, test batches $\{B_t\}_{t=1}^{T}$, TTA method $m \in \{\mathrm{BN}, \mathrm{TENT}\}$
Ensure: Adapted predictor f_θ
1: Train f_θ with RR, RRWithPrior, or LP-MST under ε-LabelDP.
2: **for** each unlabeled test batch B_t **do**
3: Forward B_t, normalizing each BN layer with the statistics of B_t.
4: Predict $\hat{y} = \arg\max_k f_\theta(x)_k$ for each $x \in B_t$ from this forward pass.
5: **if** $m = \mathrm{TENT}$ **then**
6: Update only the BN affine parameters with one entropy step on the same forward pass (effective from B_{t+1}): $\theta_{\mathrm{BN}} \leftarrow \theta_{\mathrm{BN}} - \eta \nabla_{\theta_{\mathrm{BN}}} \frac{1}{|B_t|} \sum_{x \in B_t} H(f_\theta(x))$.
7: **end if**
8: **end for**
9: **return** f_θ

A Code Availability

Anonymous code is available at: `https://anonymous.4open.science/r/dptta-CFA3`

B Scope and Limitations

Threat model. Our privacy claim concerns the training labels. Adaptation uses only the released model, public training covariates, and adaptation inputs that carry no information from the protected labels, so it is post-processing of the ϵ-LabelDP mechanism for any batch size, batch composition, or stream order (Remark 1). It provides no guarantee for the test inputs themselves. When they are sensitive, adaptation needs its own privacy accounting, and combining our pipeline with private TTA (Li et al., 2026) is a natural extension.

Theory. Proposition 1 and Lemma 2 are exact for symmetric RR, while our experiments use prior-aware LP-MST. The analysis is population-level: stationarity of the expected noisy risk does not imply small gradients on a finite privatized training set. The entropy objective is pointwise, whereas batch-normalized predictions depend on batches, and Theorem 5 analyzes gradient descent rather than one Adam step per batch on a changing stream. The PL, quadratic-growth, BN-regularity, and error–entropy transfer conditions are local and are not verified for deep networks. Proposition 4 bounds a population statistic of the private model relative to a nearest clean minimizer, which differs from the target mismatch that test-batch normalization corrects, and Theorem 7 does not guarantee improvement over the source model.

Controls. The expected symmetric-RR objective coincides with label smoothing at coefficient $1 - \rho$ (Lemma 2), and our training uses mixup. We did not run label-smoothing or mixup-ablated non-private controls, so we cannot separate privacy-specific effects from those of softened supervision; our wording therefore claims an association with label privatization rather than a privacy-specific mechanism. The matched non-private controls are single runs, and we compare point estimates rather than paired per-seed gains. Our from-scratch LP-MST sources are also weaker than the best results reported by Ghazi et al. (2021) at some budgets, so the size of the clean-data gains on stronger sources remains to be measured. Finally, we did not run a temperature-only sharpening baseline or a fixed-statistics TENT ablation; for MEMO, we compare source and test-batch statistics directly (Appendix C.1).

Operating regime. Like other BN-based online TTA methods, our pipeline needs moderately large test batches whose label composition is not strongly skewed. The evaluated online configurations fail at $B = 1$, and label-correlated streams can make them worse than source inference (Section 5.3); buffering alone does not guarantee representative batches. Online adaptation also makes predictions depend on batchmates and arrival order (Table 5). Because we take one update per batch, the batch-size sweep also changes the number of updates over the stream, so it does not isolate BN estimation error. For single-query serving, a privacy-preserving alternative is to adapt once on a fixed unlabeled pool, such as the public training covariates, freeze the model, and serve in evaluation mode; we have

not measured the utility of this variant. We also did not tune hyperparameters for each batch size, so stronger small-batch variants may exist.

Experimental scope. Our utility evidence comes from LP-MST configurations (trained from scratch, with a CLIP prior, and as a linear probe on a DINO backbone) and BN-based image classifiers. The ImageNet64 experiments freeze the backbone, so they test scale and corruption adaptation rather than privacy-induced BN drift. Evaluating other LabelDP mechanisms, architectures without BN, and other modalities is future work.

C Additional Experimental Analyses

C.1 Adaptation components

We evaluate source-free rules using the same LP-MST ResNet-18 checkpoints, target batches, and online ordering as the main CIFAR experiments. BN-Norm ("BN" in the main text) recomputes target-batch statistics with no gradient update, and TENT adds entropy minimization of the BN affine parameters. SAR uses target-batch statistics with sample-filtered entropy minimization and SAM; we evaluate it with the ResNet-18's batch normalization, although the original work also studies batch-agnostic normalization (Niu et al., 2023). MEMO minimizes the marginal entropy over test-input augmentations. The original MEMO adapts each test point independently and, by default, mixes source and single-point BN statistics with prior strength $N = 16$ (Zhang et al., 2022). We report two batched variants: MEMO with source BN statistics, which isolates the entropy update, and MEMO+BN, which normalizes with test-batch statistics ($B = 200$, source prior strength 16) and takes Adam steps with learning rate 10^{-2}. Neither is the published single-point configuration, so we do not use them to characterize MEMO in general.

MEMO screening. Our sweep script reproduces the MEMO, BN-Norm, and source numbers in Table 2. We screened about 30 configurations on the first 500 or 1,000 images of each severity-5 corruption. For MEMO with source statistics, no setting improved on the source, including learning rates from 10^{-4} to $2 \cdot 10^{-2}$ with 5 steps and 32 augmented views, AugMix severity 1 or 6, and the original CIFAR recipe (SGD over all parameters, learning rate $5 \cdot 10^{-3}$). Error approached the source as the learning rate went to zero and grew with larger learning rates or more steps. Adapting one image at a time with the original BN prior does improve on the source, but the gain comes entirely from the BN statistics: with the same prior and no gradient step, error stays within 0.25 points at every ϵ. The best prior strength ($N = 4$) lowers corruption error at $\epsilon = 1$ from 68.3% to 61.9% on the screening subset but raises clean error from 45.8% to 49.1%. For MEMO+BN, learning rates from $5 \cdot 10^{-3}$ to $2 \cdot 10^{-2}$ work well, and $5 \cdot 10^{-2}$ is worse.

Results. On clean data, the methods differ little once the source has reasonable accuracy. Under corruption, MEMO with source statistics is slightly worse than the unadapted source, whereas every rule that uses target-batch statistics improves sharply. Adding an entropy update to target-batch statistics then helps: MEMO+BN improves on BN-Norm on all 15 corruptions at $\epsilon \geq 2$ and on 13 of 15 at $\epsilon = 1$ (0.3–1.9 points on average), and it is the best rule at $\epsilon = 1$. TENT adds 5.0–5.6 points beyond BN-Norm at $\epsilon \geq 2$, 3.2–3.9 points more than MEMO+BN, but is 0.50 points worse than BN-Norm at $\epsilon = 1$. Target-batch normalization therefore accounts for most of the corruption effect, and entropy optimization adds a method- and regime-dependent refinement.

C.2 Batch size and stream composition

For the batch sweep, we hold the target distribution fixed and repeat each setting over three seeds. Each cell in Table 3 is clean / CIFAR-10-C error, and the final column is the source model evaluated in inference mode. Unlike the severity-5 metric of Table 1, CIFAR-10-C error in this sweep averages the 15 corruptions over severities 1–5, which is why its source values are lower. The clean source values differ slightly from Table 1 because this is a separate three-seed sweep. Because each batch receives one update, smaller batches also mean more updates over the stream, so the sweep does not isolate BN estimation error; it also does not include $B = 200$. BN-Norm and TENT both compute statistics from the current online batch. We did not tune method-specific hyperparameters at each

Table 3: Batch-size sweep: clean / CIFAR-10-C error (%), three seeds. CIFAR-10-C error averages 15 corruptions over severities 1–5.

ϵ	Method	$B = 1$	$B = 8$	$B = 16$	$B = 64$	Source
1	BN-Norm	89.90/89.93	47.20/52.03	44.12/49.25	41.98/47.47	45.83/60.00
1	TENT	90.00/90.00	70.38/72.77	59.89/65.78	46.61/53.77	45.83/60.00
2	BN-Norm	90.00/90.00	22.25/31.10	18.28/27.36	15.68/24.65	15.78/36.09
2	TENT	90.00/90.00	24.56/32.78	16.49/23.37	13.28/19.28	15.78/36.09
4	BN-Norm	87.75/88.36	16.21/25.92	12.00/21.76	9.25/18.69	8.80/28.37
4	TENT	89.99/89.99	16.81/24.76	11.73/18.15	7.93/14.00	8.80/28.37
8	BN-Norm	86.62/87.48	14.86/24.86	10.54/20.46	7.97/17.43	7.33/26.34
8	TENT	89.98/89.99	14.77/22.79	10.28/16.46	6.82/12.87	7.33/26.34

Table 4: Test error (%) for stream constructions ($\epsilon = 8$, $B = 64$).

Stream	Source	BN-Norm	ema_norm	TENT
homogeneous	23.86	17.14	24.09	13.90
mixed-gradual	23.86	16.33	24.02	12.54
mixed-i.i.d.	23.86	22.19	21.56	20.19
mixed-domain-correlated	23.86	21.31	21.94	18.17
mixed-joint-correlated	23.86	26.15	21.84	24.51
mixed-label-correlated	23.86	62.96	25.46	66.07

batch size or measure offline-then-frozen utility. Thus, the table characterizes the submitted online protocol rather than a best possible small-batch variant.

The stream experiment fixes $\epsilon = 8$ and $B = 64$, and scores the same 10,000 images under three arrival orders. The stream includes 15 corruption types at five severities and 20% clean data; only composition and order change. The homogeneous row has conventional single-corruption batches. Mixed rows combine domains or labels within batches, while the label-correlated row deliberately concentrates labels. ema_norm predicts with accumulated past statistics and updates those statistics only after each batch. This stabilizes one failure mode but makes predictions history-dependent.

Table 5 measures instability across arrival orders, which differs from the source-versus-adapted flips in Appendix C.4. Under mixed-i.i.d. streams, 13.9% of samples change correctness across TENT arrival orders, and 31.0% do under the label-correlated stream. ema_norm is more stable in several mixed or correlated settings, but its nonzero disagreement confirms that it remains history-dependent.

TENT remains better than source on mixed-i.i.d. and domain-correlated streams, but loses much of the homogeneous improvement. It fails severely on the label-correlated stream (66.07% versus 23.86% source error). BN-Norm also fails there, while ema_norm avoids the collapse but is still worse than source. These results motivate the conservative deployment recommendation in the main text: when stream representativeness is unknown, ordinary source inference is a safer default.

C.3 IMAGENET64 AND PRETRAINED-SOURCE RESULTS

ImageNet64 results average eight corruption types and five severities over three seeds. We draw one 10-class and one 100-class subset at random, fix both subsets across all runs, and report error over their corrupted test images. The backbone is a self-supervised DINO ResNet-50 and only its linear probe is trained with LP-MST. Because the pretraining does not use the protected downstream class labels, it does not spend label privacy under our adjacency relation; nevertheless, the frozen representation can contain useful semantic structure. The experiment therefore tests scale and corruption adaptation, rather than privacy-induced BN drift in the frozen backbone.

For $K = 10$, normalization removes 12.75–16.81 points at $\epsilon \geq 2$ (4.55 at $\epsilon = 1$); for $K = 100$, it removes 3.59–12.21 points at $\epsilon \geq 2$, once the source has useful accuracy. At $K = 100, \epsilon = 1$, the source is close to chance and no method helps. TENT does not improve on Norm in this setting and is sometimes worse, so the ImageNet64 gains come from normalization.

For a complementary source-quality check, we use a CLIP prior within the same CIFAR-100 LP-MST protocol. These are single-run evaluations and should be interpreted as a configuration check rather

Table 5: Arrival-order sensitivity ($\epsilon = 8$, $B = 64$): percentage of samples whose correctness changes across three arrival orders / mean pairwise prediction disagreement (%). Source inference is 0 / 0 by construction.

Stream	BN-Norm	ema_norm	TENT
homogeneous	12.4 / 7.4	13.0 / 8.5	12.3 / 6.9
mixed-gradual	8.8 / 5.1	18.2 / 10.9	8.9 / 5.0
mixed-i.i.d.	11.9 / 7.7	3.8 / 2.6	13.9 / 8.8
mixed-domain-correlated	16.5 / 10.2	5.9 / 3.9	16.2 / 10.1
mixed-joint-correlated	18.8 / 11.9	5.9 / 3.8	21.5 / 13.4
mixed-label-correlated	32.6 / 25.5	11.9 / 7.3	31.0 / 26.5

Table 6: ImageNet64 corruption error (%), mean over three seeds. Norm denotes BN adaptation.

	$K = 10$			$K = 100$		
ϵ	Source	Norm	TENT	Source	Norm	TENT
1	71.17	66.62	67.37	96.34	95.93	97.05
2	61.87	49.12	49.46	84.30	80.71	83.66
4	59.76	42.95	43.36	77.25	66.06	67.49
8	59.54	42.79	43.41	76.71	64.50	65.88
∞	58.28	37.56	37.52	76.48	63.69	63.46

than an estimate of seed variability. Table 7 reports the complete clean and corrupted-data results. The stronger prior makes both normalization and TENT useful at every reported privacy budget, with the largest TENT improvement of 7.40 points on clean data and 21.36 points on CIFAR-100-C.

C.4 Prediction flips

For every one of the 10,000 clean CIFAR-10 images, we compare source and adapted predictions. W→C denotes an initially wrong prediction corrected by adaptation, C→W a previously correct prediction made wrong, and W→W′ a change between two incorrect classes. "Changed" is the fraction of all labels that change. Reporting all three outcomes makes prediction churn visible rather than reporting only net accuracy.

At $\epsilon = 1$, both BN-Norm and TENT make a net improvement because W→C exceeds C→W, but they still modify roughly one in six predictions. The net benefit shrinks as privacy weakens. For the matched non-private model, C→W exceeds W→C for both rules, matching the negative accuracy change. These counts are descriptive: they do not identify a unique model defect or causal mechanism.

C.5 Exploratory comparison to hard-label retraining

We reimplemented RT and consensus-filtered RT from Das et al. (2025), as no official implementation was available. RT retrains from model-predicted hard labels on the source distribution; Cons-RT retains only consensus-filtered predictions. Both variants retrain from scratch for 60 epochs, inheriting the base backbone, optimizer, and augmentation while rebuilding the schedule for that horizon. TTA hyperparameters are identical for base and retrained checkpoints. This is an exploratory matched setup, not a claim about all implementation choices or operating regimes for hard-label retraining.

In this implementation, base plus TENT is the best clean result in every reported privacy block, and the retrained checkpoints do not benefit from subsequent clean TENT. Because we could not validate the original implementation or identify its favorable regimes, these numbers should not be read as evidence that hard-label retraining is generally ineffective. The relevant setting difference remains that RT performs source-distribution retraining, whereas our pipeline adapts at deployment on unlabeled target inputs.

Table 7: CIFAR-100 results using a CLIP prior (test error, %). All entries are single evaluations. Δ is relative to LabelDP; lower is better.

		Clean CIFAR-100			
ϵ	LabelDP	LabelDP + BN	LabelDP + TENT	Δ(BN)	Δ(TENT)
1	36.90	34.14	**33.14**	-2.76	-3.76
2	41.98	37.38	**35.41**	-4.60	-6.57
4	40.20	34.80	**32.80**	-5.40	-7.40
8	25.48	24.20	**23.64**	-1.28	-1.84
∞		External non-private source: 23.62			
		CIFAR-100-C, severity 5			
ϵ	LabelDP	LabelDP + BN	LabelDP + TENT	Δ(BN)	Δ(TENT)
1	56.07	42.60	**40.80**	-13.47	-15.27
2	60.55	46.02	**42.84**	-14.53	-17.71
4	61.86	43.79	**40.50**	-18.07	-21.36
8	49.39	34.06	**31.60**	-15.33	-17.79

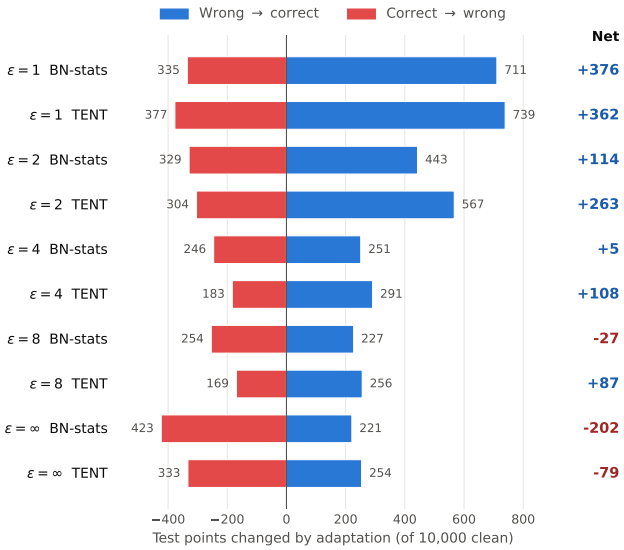


Figure 2: Predictions fixed (W→C) and broken (C→W) by TTA on clean CIFAR-10; "BN-stats" is BN adaptation. The net column is the change in the number of correct predictions out of 10,000, matching Table 1.

D Additional Results

D.1 Seed Variability

To assess statistical variability for the main empirical claims, we repeat the CIFAR-10 clean and corrupted evaluations over five seeds. Seeds vary model initialization, minibatch order, and randomized-response label perturbations. Table 10 reports the point estimates used in Table 1 with standard deviations across the five seeds, and Table 8 counts prediction transitions for the evaluation behind these point estimates. The batch-size sweep and ImageNet64 experiments use three seeds, the stream experiments use three arrival orders, and the remaining experiments, including the non-private controls, are single runs.

D.2 Corruption Results by Severity

This section provides the corruption error tables for all severity levels and corruption types, comparing LabelDP models with and without TENT at varying ϵ.

Table 8: Per-sample prediction flips on the clean CIFAR-10 test set. Source accuracy is the unadapted accuracy (%); the $\epsilon = \infty$ rows are the matched non-private control. W→C wrong-to-correct, C→W correct-to-wrong, W→W′ wrong-to-different-wrong. Counts are for the evaluation behind the Table 1 point estimates.

ϵ	Method	Src acc	Δacc	Changed %	W→C	C→W	W→W′
1	BN-Norm	54.24	+3.76	17.51	711	335	705
1	TENT	54.24	+3.62	18.73	739	377	757
2	BN-Norm	83.90	+1.14	10.04	443	329	232
2	TENT	83.90	+2.63	10.89	567	304	218
4	BN-Norm	91.41	+0.05	5.80	251	246	83
4	TENT	91.41	+1.08	5.61	291	183	87
8	BN-Norm	92.68	−0.27	5.58	227	254	77
8	TENT	92.68	+0.87	5.04	256	169	79
∞	BN-Norm	91.56	−2.02	7.63	221	423	119
∞	TENT	91.56	−0.79	6.87	254	333	100

Table 9: Clean CIFAR-10 test error (%) for exploratory hard-label retraining.

ϵ	Model	None	+TENT
1	Base	45.76	42.14
1	RT	45.79	47.09
1	Cons-RT	44.21	46.19
4	Base	8.59	7.51
4	RT	11.94	13.26
4	Cons-RT	16.08	17.37
8	Base	7.32	6.45
8	RT	10.97	11.71
∞	Base	8.44	9.23
∞	RT	10.56	11.71

	Clean error (%)				**Corrupted error** (%)			
Method	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = 4$	$\varepsilon = 8$	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = 4$	$\varepsilon = 8$
LabelDP	$45.76_{\pm 0.79}$	$16.10_{\pm 0.74}$	$8.59_{\pm 0.64}$	$7.32_{\pm 0.20}$	$68.32_{\pm 1.37}$	$50.30_{\pm 1.21}$	$44.97_{\pm 0.65}$	$44.17_{\pm 1.17}$
LabelDP + BN	$42.00_{\pm 1.00}$	$14.96_{\pm 0.42}$	$8.54_{\pm 0.22}$	$7.59_{\pm 0.21}$	$50.43_{\pm 0.97}$	$28.93_{\pm 0.66}$	$23.79_{\pm 0.30}$	$23.28_{\pm 0.31}$
LabelDP + TENT	$42.14_{\pm 1.13}$	$13.47_{\pm 0.26}$	$7.51_{\pm 0.03}$	$6.45_{\pm 0.14}$	$50.93_{\pm 1.12}$	$23.93_{\pm 0.36}$	$18.58_{\pm 0.06}$	$17.68_{\pm 0.15}$

Table 10: CIFAR-10 clean and corrupted test error (%). Values match the main-text point estimates; error bars show standard deviation over five seeds.

Model (ε)	**gauss**	**shot**	**imp**	**defoc**	**glass**	**motion**	**zoom**	**snow**	**frost**	**fog**	**bright**	**contrast**	**elastic**	**pixel**	**jpeg**	**mCE**
LabelDP ($\epsilon = 8$)	80.16	74.80	86.34	40.50	41.73	34.87	37.30	24.17	23.90	35.75	13.64	46.35	29.27	60.18	33.50	44.17
LabelDP+TENT	26.81	23.83	28.48	10.86	27.46	12.60	11.47	14.21	14.38	12.60	8.38	10.89	19.67	18.12	25.53	**17.68**
LabelDP ($\epsilon = 4$)	79.53	74.69	73.67	42.83	45.01	36.72	38.76	26.07	27.03	42.23	14.77	51.94	30.25	56.47	34.51	44.97
LabelDP+TENT	27.32	25.18	29.09	11.37	28.57	12.89	12.51	15.56	15.66	13.80	10.14	11.87	20.61	17.87	26.25	**18.58**
LabelDP ($\epsilon = 2$)	68.28	63.65	68.95	48.80	50.96	47.68	49.64	39.59	36.93	61.87	30.25	54.52	36.57	57.39	39.39	50.30
LabelDP+TENT	31.01	28.38	33.31	17.32	33.28	19.38	16.80	23.20	22.44	20.84	16.24	17.80	25.55	23.28	28.87	**23.93**
LabelDP ($\epsilon = 1$)	74.98	72.68	72.64	74.54	68.28	71.12	73.87	66.83	73.80	59.89	64.16	74.59	57.39	64.73	55.26	68.32
LabelDP+TENT	54.36	53.26	55.00	43.73	57.20	46.87	45.99	55.95	54.80	49.59	47.60	49.30	50.58	50.19	49.57	**50.93**

Table 11: Corruption error (%) performance (**severity 5**) across models. TENT lowers the error for every corruption type and privacy budget. mCE represents the mean corruption error.

Model (ε)	gauss	shot	imp	defoc	glass	motion	zoom	snow	frost	fog	bright	contrast	elastic	pixel	jpeg	mCE
LabelDP ($\epsilon = 8$)	74.70	63.01	72.42	20.88	45.40	27.64	28.11	19.02	19.22	14.65	9.76	19.31	24.25	41.57	28.26	33.88
LabelDP+TENT	23.67	20.69	22.46	7.99	27.72	10.30	9.62	13.18	12.86	8.65	7.55	8.70	13.41	12.51	21.67	**14.73**
LabelDP ($\epsilon = 4$)	75.09	62.84	59.62	22.82	47.21	29.81	29.85	21.25	21.35	18.01	11.06	22.77	25.85	37.95	29.23	34.31
LabelDP+TENT	24.43	21.01	23.71	9.18	29.12	11.97	10.99	13.87	13.41	9.65	8.59	9.59	14.60	13.68	22.45	**15.75**
LabelDP ($\epsilon = 2$)	63.60	54.76	61.28	34.78	53.31	41.42	41.46	32.72	32.57	33.27	22.75	33.01	36.21	40.79	35.33	41.15
LabelDP+TENT	28.45	26.40	27.86	15.31	33.62	17.82	17.02	21.16	21.17	15.97	14.61	15.57	20.72	19.47	26.37	**21.44**
LabelDP ($\epsilon = 1$)	73.39	70.04	68.47	66.57	70.22	69.12	69.75	61.73	70.42	50.60	56.80	60.34	60.86	59.28	52.64	64.02
LabelDP+TENT	52.87	51.17	52.43	42.75	57.01	45.93	44.67	51.67	52.20	45.97	44.99	46.19	47.47	47.04	48.00	**48.69**

Table 12: Corruption error (%) performance (**severity 4**) across models.

Model (ε)	gauss	shot	imp	defoc	glass	motion	zoom	snow	frost	fog	bright	contrast	elastic	pixel	jpeg	mCE
LabelDP ($\epsilon = 8$)	67.69	53.31	52.32	13.55	30.69	27.64	22.14	15.54	19.28	9.80	8.77	12.78	17.33	20.12	25.55	26.43
LabelDP+TENT	21.76	18.36	15.64	6.92	18.90	10.05	8.84	10.98	12.54	7.58	7.10	7.99	9.84	9.92	20.52	**12.46**
LabelDP ($\epsilon = 4$)	68.49	52.70	41.12	15.16	33.20	30.04	24.10	17.89	21.05	12.32	10.08	14.85	19.29	21.02	26.61	27.20
LabelDP+TENT	22.45	19.11	17.06	7.90	19.85	11.25	10.13	11.93	13.97	8.75	8.31	9.02	11.16	11.03	21.62	**13.57**
LabelDP ($\epsilon = 2$)	58.53	47.31	49.30	25.34	42.82	41.16	35.12	28.62	33.57	22.62	20.17	25.77	28.82	25.08	31.78	34.40
LabelDP+TENT	26.70	23.85	22.50	13.81	25.27	17.57	16.27	19.37	20.81	14.43	14.02	14.84	17.38	16.54	24.91	**19.22**
LabelDP ($\epsilon = 1$)	72.02	68.49	63.07	58.63	62.87	68.97	66.13	57.62	71.29	46.90	53.34	53.96	59.26	49.74	51.17	60.23
LabelDP+TENT	50.95	50.08	48.67	42.14	52.38	45.53	44.22	49.34	53.41	44.47	43.85	44.96	45.84	45.15	47.16	**47.21**

Table 13: Corruption error (%) performance (**severity 3**) across models.

Model (ε)	gauss	shot	imp	defoc	glass	motion	zoom	snow	frost	fog	bright	contrast	elastic	pixel	jpeg	mCE
LabelDP ($\epsilon = 8$)	47.89	28.78	42.34	8.52	32.59	19.46	16.85	16.95	14.89	8.29	8.22	9.95	13.21	14.61	22.54	20.34
LabelDP+TENT	17.32	12.86	12.84	6.55	19.17	8.84	8.07	11.45	10.69	6.99	6.80	7.50	9.41	9.07	18.83	**11.09**
LabelDP ($\epsilon = 4$)	48.33	27.90	32.58	10.42	35.69	21.82	18.66	19.57	16.36	10.04	8.98	11.49	14.87	15.72	23.92	21.09
LabelDP+TENT	18.12	13.85	14.16	7.69	19.94	10.04	9.34	12.10	11.66	8.37	8.00	8.53	10.48	10.13	19.73	**12.14**
LabelDP ($\epsilon = 2$)	44.45	30.83	41.74	18.10	45.26	33.15	29.45	30.47	27.57	17.62	18.09	20.64	23.72	20.85	29.58	28.77
LabelDP+TENT	22.53	19.15	19.52	13.51	25.58	16.39	15.39	19.16	18.92	13.96	13.66	14.42	17.10	15.96	23.66	**17.93**
LabelDP ($\epsilon = 1$)	67.84	62.37	59.29	49.01	65.44	65.09	61.45	59.12	64.93	43.94	50.22	49.55	54.20	47.42	49.64	56.63
LabelDP+TENT	48.56	46.01	46.47	42.06	53.14	44.91	43.40	50.58	50.46	42.86	43.02	44.29	45.98	44.73	46.73	**46.21**

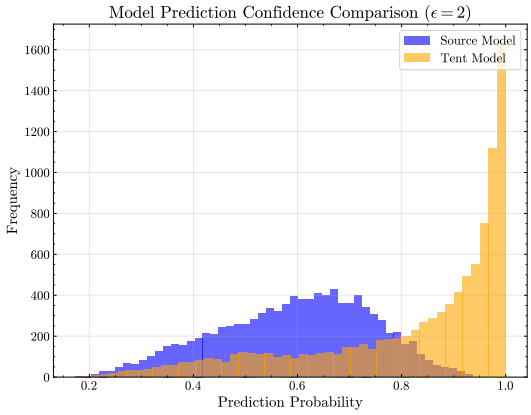
Table 14: Corruption error (%) performance (**severity 2**) across models.

Model (ε)	gauss	shot	imp	defoc	glass	motion	zoom	snow	frost	fog	bright	contrast	elastic	pixel	jpeg	mCE
LabelDP ($\epsilon = 8$)	26.31	17.93	25.88	7.17	32.67	12.12	13.68	10.54	10.13	7.24	7.56	7.47	13.05	9.53	17.08	14.56
LabelDP+TENT	12.71	10.56	10.13	6.37	18.67	7.63	8.04	8.08	8.57	6.50	6.74	6.77	9.66	7.86	14.72	**9.53**
LabelDP ($\epsilon = 4$)	27.05	18.86	20.44	8.48	36.62	14.03	15.76	12.14	12.55	8.38	8.65	8.58	14.78	11.14	18.61	15.74
LabelDP+TENT	13.47	11.49	11.04	7.54	19.76	9.21	9.04	9.40	9.95	7.79	7.67	7.89	10.99	9.42	15.75	**10.69**
LabelDP ($\epsilon = 2$)	29.91	23.75	30.27	15.94	45.68	23.72	24.48	21.14	21.81	15.54	16.71	15.66	23.61	17.18	23.95	23.29
LabelDP+TENT	18.82	16.71	16.75	13.43	25.20	15.16	15.50	15.46	16.09	13.48	13.54	13.95	17.41	14.82	19.93	**16.42**
LabelDP ($\epsilon = 1$)	62.04	57.29	53.46	45.78	67.34	57.73	56.15	51.20	56.15	43.78	47.65	44.28	53.19	44.75	47.21	52.53
LabelDP+TENT	45.82	44.06	44.58	42.24	52.24	43.35	43.81	45.36	46.63	42.28	42.58	42.50	47.00	43.42	45.07	**44.73**

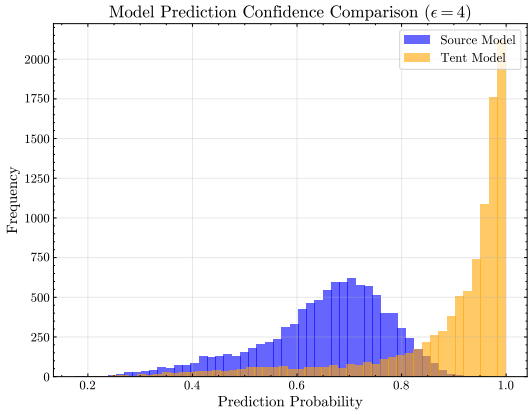
Table 15: Corruption error (%) performance (**severity 1**) across models.

E ADDITIONAL FIGURES FOR CONFIDENCE AND CALIBRATION ANALYSIS

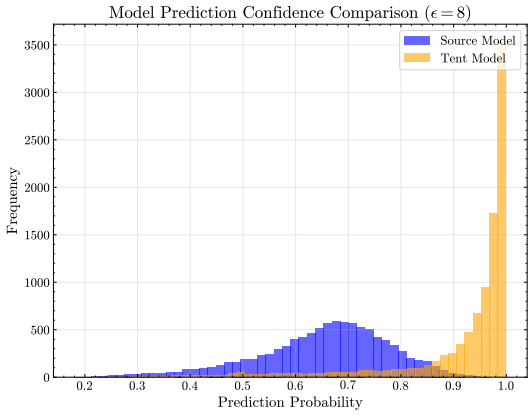
This section includes the confidence and calibration analysis for $\epsilon = [2, 4, 8]$ for CIFAR-10.



(a) $\epsilon = 2$

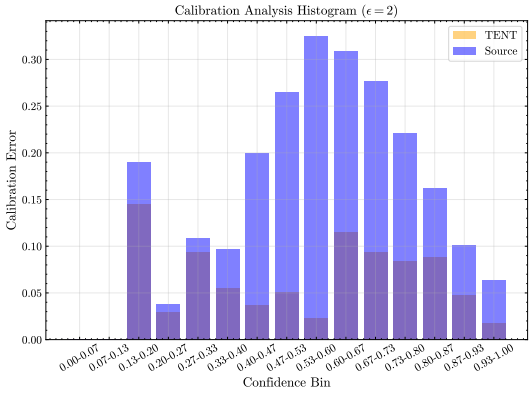


(b) $\epsilon = 4$

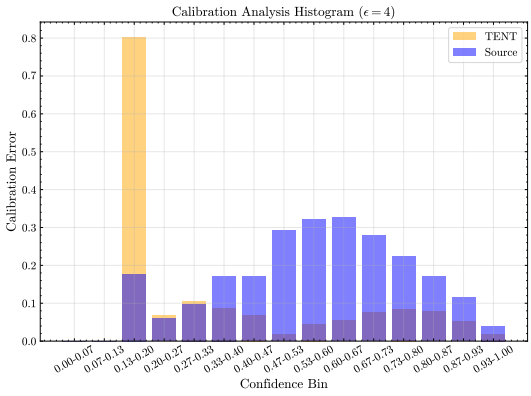


(c) $\epsilon = 8$

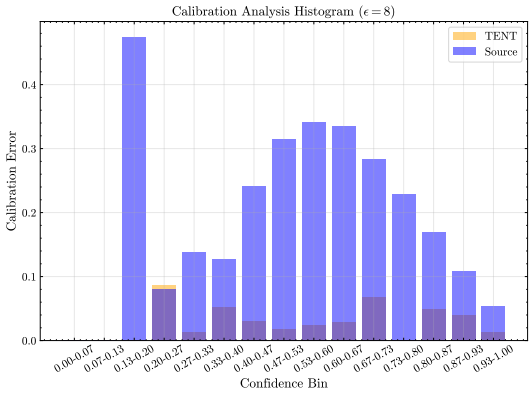
Figure 3: Confidence distributions of LabelDP-trained models at different privacy budgets. Test-time adaptation (TENT) consistently shifts the model toward higher prediction confidence across all ϵ.



(a) $\epsilon = 2$. ECE: 0.244 (source) vs. 0.046 (TENT).



(b) $\epsilon = 4$. ECE: 0.262 (source) vs. 0.037 (TENT).



(c) $\epsilon = 8$. ECE: 0.273 (source) vs. 0.020 (TENT).

Figure 4: Calibration Analysis of LabelDP-trained models at different privacy budgets. Calibration error is the absolute difference between the average accuracy of the bin and the prediction probability.

F Detailed Proofs for Section 4

F.1 Notation for the Theory Proofs

Notation	Meaning
K, $[K]$	Number of classes and the class index set $\{1, \dots, K\}$.
$(x, y) \sim \mathcal{D}$	Clean input-label pair drawn from the data distribution.
$\tilde{y}$	Label released by randomized response under ε-LabelDP.
α, β_{flip}	Retention probability and per-wrong-class flip probability in randomized response.
$\rho(\varepsilon)$	Label signal strength $\frac{e^{\varepsilon}-1}{e^{\varepsilon}+K-1}$.
$p_\theta^{(j)}(x)$	Model probability assigned to class j at input x.
$\ell(\theta; x, j)$	Cross-entropy loss for class j, equal to $-\log p_\theta^{(j)}(x)$.
$R(\theta)$, R^*	Clean risk and its infimum over parameters.
$\tilde{R}(\theta)$	Noisy-label risk optimized under randomized-response labels.
$R_{\text{unif}}(\theta)$	Uniform-label risk $\mathbb{E}_x[\frac{1}{K}\sum_{j=1}^{K} \ell(\theta; x, j)]$.
θ^*, $\tilde{\theta}^*$, $\hat{\theta}$	A clean-risk minimizer, an exact noisy-risk stationary point, and an approximate outer-training output.
Δ_g	Uniform bound on the gradient bias $\sup_\theta \|\nabla R_{\text{unif}}(\theta) - \nabla R(\theta)\|$.
μ_R, ν_R	Clean PL constant and quadratic-growth constant for R.
Θ_*, $\bar{\theta}$	Set of clean-risk minimizers and a nearest clean minimizer to the current parameter.
$h_l(x; \theta)$, $m_l(\theta)$	Pre-BN activation at layer l and its population mean $\mathbb{E}[h_l(X; \theta)]$.
$\mu_l^{(k)}(\theta)$, π_k	Class-conditional pre-BN mean and class prior $\Pr(y = k)$.
$Jm_l(\theta)$, G_l	Jacobian of the BN mean map and its local operator-norm bound.
θ_s	Line-segment interpolation between two parameter vectors.
$\theta = (\phi, \psi)$, $\psi = (\gamma, \beta)$	Full model parameters split into frozen parameters and BN affine parameters.
$H(\psi)$, $H_{\hat{\theta}}(\psi)$	Population prediction-entropy objective for TENT, optionally with the outer parameter fixed at $\hat{\theta}$.
ψ_0, $\psi^\dagger$, $\psi^\dagger_{\hat{\theta}}$	Initial BN parameters, entropy minimizer, and fixed-$\hat{\theta}$ entropy minimizer.
L_H, μ_H	Smoothness and PL constants for the TENT entropy objective.
ΔH_0	Initial entropy gap $H(\psi_0) - H(\psi^\dagger)$, or its fixed-$\hat{\theta}$ analogue.
T, T_{in}	Number of TENT gradient steps.
δ_{out}	Approximate stationarity residual for the outer noisy-risk objective.
L_ρ, μ_ρ	Smoothness and PL constants for the noisy risk $\tilde{R}$.
η, σ^2_{sgd}	Outer SGD step size and stochastic-gradient variance bound.
$\mathcal{E}_{\text{out}}(T)$	Standard PL-SGD upper bound on the expected noisy-risk optimization gap after T outer steps.
κ, c_R, c_H	Error–entropy transfer constants used only for optional classification-error bounds.

F.2 Label Noise Model via Randomized Response

All smoothness, PL, quadratic-growth, and BN-Lipschitz conditions in this appendix are local to the region containing the iterates. We state quadratic growth as a separate local assumption (Assumption 2); under the regularity conditions of Karimi et al. (2016), PL implies related error-bound properties, but our results do not rely on that implication. The entropy PL condition concerns only BN affine parameters and remains an assumption. They support a mechanism-level account of why adaptation helps, and are not properties we verify globally for a ResNet. Our experiments are consistent with this account when source quality and batch statistics are adequate, but they do not verify the assumptions themselves.

For ε-LabelDP, each true label $y \in [K]$ is replaced by a noisy label $\tilde{y}$ drawn according to the Randomized Response mechanism:

$$\Pr[\tilde{y} = j \mid y] = \begin{cases} \frac{e^{\varepsilon}}{e^{\varepsilon}+K-1}, & \text{if } j = y, \\ \frac{1}{e^{\varepsilon}+K-1}, & \text{otherwise.} \end{cases} \tag{12}$$

Define the *retention probability* $\alpha := \frac{e^\varepsilon}{e^\varepsilon+K-1}$ and the *flip probability per class* $\beta_{\text{flip}} := \frac{1}{e^\varepsilon+K-1}$, so that $\alpha + (K-1)\beta_{\text{flip}} = 1$.

The Randomized Response can be expressed as a *label transition matrix* $Q \in \mathbb{R}^{K\times K}$:

$$Q_{jk} = \Pr[\tilde{y} = j \mid y = k] = \alpha \cdot \mathbf{1}_{j=k} + \beta_{\text{flip}} \cdot \mathbf{1}_{j\neq k} = \beta_{\text{flip}} \cdot \mathbf{1}_K + (\alpha - \beta_{\text{flip}}) \cdot I_K, \tag{13}$$

where $\mathbf{1}_K$ is the $K \times K$ all-ones matrix and I_K is the identity. The eigenvalues of Q are $\lambda_1 = 1$ (with eigenvector $\mathbf{1}$) and $\lambda_2 = \cdots = \lambda_K = \alpha - \beta_{\text{flip}} = \frac{e^\varepsilon-1}{e^\varepsilon+K-1}$. Let $\rho := \frac{e^\varepsilon-1}{e^\varepsilon+K-1}$.

F.3 Proof of Proposition 1

Recall from eq. 13 that $Q = \beta_{\text{flip}}\mathbf{1}_K + \rho\, I_K$, where $\mathbf{1}_K$ is the $K \times K$ all-ones matrix and we used $\alpha - \beta_{\text{flip}} = \rho$. We also use $K\beta_{\text{flip}} = \frac{K}{e^\varepsilon+K-1} = 1 - \rho$. Throughout, $\varepsilon > 0$, so $\rho \in (0, 1)$.

Step 1: the privatized posterior. Randomized response draws $\tilde{y}$ from y alone, so $\tilde{y}$ is independent of x given y. By the law of total probability,

$$\pi_\varepsilon^{(j)}(x) = \Pr[\tilde{y} = j \mid x] = \sum_{k=1}^{K} \Pr[\tilde{y} = j \mid y = k] \Pr[y = k \mid x] = \big(Q\pi(x)\big)_j.$$

Step 2: the mixture form. Since the entries of $\pi(x)$ sum to one, $\mathbf{1}_K\pi(x)$ is the all-ones vector, which equals Ku. Hence

$$\pi_\varepsilon(x) = Q\pi(x) = K\beta_{\text{flip}}\, u + \rho\, \pi(x) = (1-\rho)\, u + \rho\, \pi(x).$$

Step 3: ranking and margins. For any classes j, k, the uniform term cancels:

$$\pi_\varepsilon^{(j)}(x) - \pi_\varepsilon^{(k)}(x) = \rho\big(\pi^{(j)}(x) - \pi^{(k)}(x)\big).$$

Since $\rho > 0$, every pairwise difference keeps its sign, so $\pi_\varepsilon(x)$ ranks the classes exactly as $\pi(x)$ does. In particular it has the same argmax, which is unique whenever the argmax of $\pi(x)$ is unique. Since $\rho < 1$, every margin shrinks by the factor ρ.

Step 4: top-class probability. By Step 3 the top class $c(x)$ is shared, and

$$\max_j \pi_\varepsilon^{(j)}(x) = \rho\, \pi^{(c)}(x) + \frac{1-\rho}{K} \le \pi^{(c)}(x),$$

because $\pi^{(c)}(x) \ge 1/K$. Equality holds only if $\pi(x) = u$.

F.4 Proof of Lemma 2

The clean population risk is $R(\theta) := \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(\theta; x, y)]$, where $\ell(\theta; x, y) = -\log p_\theta^{(y)}(x)$ is the cross-entropy loss and $p_\theta^{(y)}(x)$ is the predicted probability for class y.

Under LabelDP, the learner observes the *noisy risk*:

$$\tilde{R}(\theta) := \mathbb{E}_{(x,y)\sim\mathcal{D}}\mathbb{E}_{\tilde{y}|y}[\ell(\theta; x, \tilde{y})] = \mathbb{E}_{x,y}\left[\alpha \cdot \ell(\theta; x, y) + \beta_{\text{flip}} \sum_{j\neq y} \ell(\theta; x, j)\right]. \tag{14}$$

Proof. Fix a sample (x, y). By the Randomized Response mechanism in Equation 13, the conditional expectation of the loss on the noisy label is:

$$\mathbb{E}_{\tilde{y}|y}[\ell(\theta;x,\tilde{y})] = \alpha\,\ell(\theta;x,y) + \beta_{\text{flip}} \sum_{j\neq y} \ell(\theta;x,j) \tag{15}$$

$$= \alpha\,\ell(\theta;x,y) + \beta_{\text{flip}} \left(\sum_{j=1}^{K} \ell(\theta;x,j) - \ell(\theta;x,y) \right) \qquad \text{(where } j = y\text{)}$$

$$= (\alpha - \beta_{\text{flip}})\,\ell(\theta;x,y) + \beta_{\text{flip}} \sum_{j=1}^{K} \ell(\theta;x,j) \tag{16}$$

$$= \rho(\varepsilon)\,\ell(\theta;x,y) + K\beta_{\text{flip}} \cdot \frac{1}{K} \sum_{j=1}^{K} \ell(\theta;x,j) \qquad (\rho = \alpha - \beta_{\text{flip}})$$

$$= \rho(\varepsilon)\,\ell(\theta;x,y) + (1-\rho(\varepsilon)) \cdot \frac{1}{K} \sum_{j=1}^{K} \ell(\theta;x,j), \tag{17}$$

Taking expectation of Equation 17 over $(x,y) \sim \mathcal{D}$ and using the definitions of R and R_{unif}:

$$\tilde{R}(\theta) = \mathbb{E}_{(x,y)}\mathbb{E}_{\tilde{y}|y}[\ell(\theta;x,\tilde{y})] \qquad \text{(def. of } \tilde{R}\text{)}$$

$$= \rho(\varepsilon)\,\mathbb{E}_{(x,y)}[\ell(\theta;x,y)] + (1-\rho(\varepsilon))\,\mathbb{E}_x \left[\frac{1}{K} \sum_{j=1}^{K} \ell(\theta;x,j) \right] \qquad \text{(linearity)}$$

$$= \rho(\varepsilon)\,R(\theta) + (1-\rho(\varepsilon))\,R_{\text{unif}}(\theta). \tag{18}$$

Finally, differentiating Equation 18:

$$\nabla_\theta \tilde{R}(\theta) \;=\; \rho(\varepsilon)\,\nabla_\theta R(\theta) + (1-\rho(\varepsilon))\,\nabla_\theta R_{\text{unif}}(\theta) \qquad \square$$

F.5 Proof of Theorem 3

Proof. **Step 1: Isolate the clean-risk gradient.** By the gradient decomposition established in Lemma 2, we can write

$$\nabla R(\theta) = \nabla \tilde{R}(\theta) - (1-\rho(\varepsilon))\{\nabla R_{\text{unif}}(\theta) - \nabla R(\theta)\}. \tag{19}$$

Let $\tilde{\theta}^*$ be an exact stationary point of the noisy risk, so $\|\nabla \tilde{R}(\tilde{\theta}^*)\| = 0$.

Step 2: Bound the clean-risk gradient. Taking norms on both sides of Equation 19 evaluated at $\tilde{\theta}^*$, and using the definition $\Delta_g := \sup_\theta \|\nabla R_{\text{unif}}(\theta) - \nabla R(\theta)\|$, we obtain

$$\|\nabla R(\tilde{\theta}^*)\| \leq \|\nabla \tilde{R}(\tilde{\theta}^*)\| + (1-\rho(\varepsilon))\|\nabla R_{\text{unif}}(\tilde{\theta}^*) - \nabla R(\tilde{\theta}^*)\| \tag{20}$$

$$\leq 0 + (1-\rho(\varepsilon))\Delta_g. \tag{21}$$

Step 3: Apply clean PL. The clean PL condition in Assumption 1 gives $R(\theta) - R^* \leq \frac{1}{2\mu_R}\|\nabla R(\theta)\|^2$ for every θ. Applying this with $\theta = \tilde{\theta}^*$ and using $R(\theta^*) = R^*$ for any clean-risk minimizer θ^* gives

$$R(\tilde{\theta}^*) - R(\theta^*) \leq \frac{1}{2\mu_R}\|\nabla R(\tilde{\theta}^*)\|^2 \leq \frac{(1-\rho(\varepsilon))^2}{2\mu_R}\Delta_g^2. \tag{22}$$

$\square$

F.6 Proof of Proposition 4: Effect on Batch Normalization Statistics

Proof. **Step 1: Convert clean-risk excess into parameter distance.** Let $\Theta_* := \arg\min_\theta R(\theta)$ and let $\bar{\theta} \in \Theta_*$ be a nearest clean minimizer to the exact noisy-risk stationary point $\tilde{\theta}^*$. Since $R(\bar{\theta}) = R^*$, Theorem 3 gives

$$R(\tilde{\theta}^*) - R^* \leq \frac{(1-\rho(\varepsilon))^2}{2\mu_R}\Delta_g^2.$$

By the quadratic-growth part of Assumption 2,

$$\frac{\nu_R}{2}\left\|\tilde{\theta}^* - \bar{\theta}\right\|^2 = \frac{\nu_R}{2}\operatorname{dist}(\tilde{\theta}^*, \Theta_*)^2 \le R(\tilde{\theta}^*) - R^*.$$

Combining the two displays yields

$$\left\|\tilde{\theta}^* - \bar{\theta}\right\| \le \frac{(1-\rho(\varepsilon))\Delta_g}{\sqrt{\mu_R \nu_R}}. \tag{23}$$

Step 2: Transfer parameter distance to the BN mean. For $\theta_s := \bar{\theta} + s(\tilde{\theta}^* - \bar{\theta})$, the fundamental theorem of calculus gives

$$m_l(\tilde{\theta}^*) - m_l(\bar{\theta}) = \int_0^1 J m_l(\theta_s)(\tilde{\theta}^* - \bar{\theta})\,\mathrm{d}s.$$

Taking norms and using the BN-Lipschitz part of Assumption 2,

$$\left\|m_l(\tilde{\theta}^*) - m_l(\bar{\theta})\right\| \le \sup_{s\in[0,1]} \|J m_l(\theta_s)\|_{\mathrm{op}} \left\|\tilde{\theta}^* - \bar{\theta}\right\| \le G_l \left\|\tilde{\theta}^* - \bar{\theta}\right\|.$$

Substituting Equation 23 proves

$$\left\|m_l(\tilde{\theta}^*) - m_l(\bar{\theta})\right\| \le \frac{G_l(1-\rho(\varepsilon))}{\sqrt{\mu_R \nu_R}}\Delta_g,$$

which is the claimed BN mean perturbation bound. □

F.7 Proof of Theorem 5

Proof. Let $\Delta_t := H(\psi_t) - H(\psi^\dagger)$. By L_H-smoothness and the update $\psi_{t+1} = \psi_t - \frac{1}{L_H}\nabla H(\psi_t)$,

$$\begin{aligned} H(\psi_{t+1}) &\le H(\psi_t) + \langle \nabla H(\psi_t), \psi_{t+1} - \psi_t\rangle + \frac{L_H}{2}\|\psi_{t+1} - \psi_t\|^2 \\ &= H(\psi_t) - \frac{1}{2L_H}\|\nabla H(\psi_t)\|^2. \end{aligned}$$

The PL inequality for H gives

$$\|\nabla H(\psi_t)\|^2 \ge 2\mu_H\big(H(\psi_t) - H(\psi^\dagger)\big) = 2\mu_H\Delta_t.$$

Combining the two displays,

$$\Delta_{t+1} \le \left(1 - \frac{\mu_H}{L_H}\right)\Delta_t.$$

Iterating from $t = 0$ to $T-1$ gives

$$H(\psi_T) - H(\psi^\dagger) \le \left(1 - \frac{\mu_H}{L_H}\right)^T \Delta H_0(\varepsilon).$$

Finally, for $a \in [0,1]$, $(1-a)^T \le e^{-aT} \le (1+aT)^{-1}$. Taking $a = \mu_H/L_H$ gives the reciprocal bound in Equation 10. □

F.8 Proof of Corollary 6

Proof. Apply the assumed error–entropy transfer inequality at the final TENT iterate:

$$\operatorname{err}(\psi_T) - \operatorname{err}(\psi^\dagger) \le \kappa\big(H(\psi_T) - H(\psi^\dagger)\big).$$

Substituting the entropy decrease from Theorem 5 gives the claimed geometric and reciprocal error bounds. □

F.9 Combined Two-Stage Surrogate Bound

The following statement adds the private-training and adaptation bounds. Its left-hand side combines the clean-risk excess of the *unadapted* model with the entropy residual after adaptation. It is not the clean risk of the adapted predictor $(\hat{\phi}, \psi_{T_{\mathrm{in}}})$, and it does not guarantee improvement over the source model.

Theorem 7 (Combined Surrogate Bound)**.** *Let $\hat{\theta}$ be the output of private training, and suppose $\|\nabla \tilde{R}(\hat{\theta})\| \le \delta_{\mathrm{out}}$. Under the assumptions of Theorems 3 and 5, for any $\lambda \ge 0$,*

$$R(\hat{\theta}) - R^* + \lambda(H_{\hat{\theta}}(\psi_{T_{\mathrm{in}}}) - H_{\hat{\theta}}(\psi^\dagger_{\hat{\theta}})) \le \frac{\big(\delta_{\mathrm{out}} + (1-\rho(\varepsilon))\Delta_g\big)^2}{2\mu_R} \;+\; \lambda\left(1 - \frac{\mu_H}{L_H}\right)^{T_{\mathrm{in}}} \Delta H_0(\widehat{\theta}). \tag{24}$$

If, in addition, the error–entropy transfer condition $\mathrm{err}(\widehat{\theta}, \psi) - \mathrm{err}_* \le c_R(R(\widehat{\theta}) - R^*) + c_H(H_{\widehat{\theta}}(\psi) - H_{\widehat{\theta}}(\psi^\dagger_{\widehat{\theta}}))$ holds for some $c_R, c_H \ge 0$, then

$$\mathrm{err}(\widehat{\theta}, \psi_{T_{\mathrm{in}}}) - \mathrm{err}_* \le c_R \frac{\big(\delta_{\mathrm{out}} + (1-\rho)\Delta_g\big)^2}{2\mu_R} + c_H\left(1 - \frac{\mu_H}{L_H}\right)^{T_{\mathrm{in}}} \Delta H_0(\widehat{\theta}). \tag{25}$$

F.10 Proof of Theorem 7

Proof. **Step 1: Outer clean-risk residual.** The gradient decomposition from Lemma 2 gives

$$\nabla R(\hat{\theta}) = \nabla \tilde{R}(\hat{\theta}) - (1-\rho(\varepsilon))\{\nabla R_{\mathrm{unif}}(\hat{\theta}) - \nabla R(\hat{\theta})\}.$$

Using $\left\|\nabla \tilde{R}(\hat{\theta})\right\| \le \delta_{\mathrm{out}}$ and the bounded gradient-bias assumption,

$$\left\|\nabla R(\hat{\theta})\right\| \le \delta_{\mathrm{out}} + (1-\rho(\varepsilon))\Delta_g.$$

The clean PL condition in Assumption 1 therefore implies

$$R(\hat{\theta}) - R^* \le \frac{\big(\delta_{\mathrm{out}} + (1-\rho(\varepsilon))\Delta_g\big)^2}{2\mu_R}. \tag{26}$$

Step 2: Inner TENT residual. Apply Theorem 5 to the fixed-backbone entropy objective $H_{\hat{\theta}}$:

$$H_{\hat{\theta}}(\psi_{T_{\mathrm{in}}}) - H_{\hat{\theta}}(\psi^\dagger_{\hat{\theta}}) \le \left(1 - \frac{\mu_H}{L_H}\right)^{T_{\mathrm{in}}} \Delta H_0(\hat{\theta}). \tag{27}$$

Step 3: Combine the surrogate bounds. Adding Equation 26 and λ times Equation 27 gives Equation 24.

Step 4: Optional classification-error bound. If the stated error–entropy transfer inequality holds, apply it at $\psi = \psi_{T_{\mathrm{in}}}$:

$$\mathrm{err}(\hat{\theta}, \psi_{T_{\mathrm{in}}}) - \mathrm{err}_* \le c_R\big(R(\hat{\theta}) - R^*\big) + c_H\big(H_{\hat{\theta}}(\psi_{T_{\mathrm{in}}}) - H_{\hat{\theta}}(\psi^\dagger_{\hat{\theta}})\big).$$

Substituting Equations 26 and 27 proves Equation 25. □

F.11 A Standard PL-SGD Plug-in for the Outer Residual

The combined bound above is stated in terms of an outer stationarity residual δ_{out}. The following standard calculation shows how such a residual can be obtained from SGD under the usual smoothness, PL, and bounded-variance assumptions. This subsection is only a plug-in for interpreting δ_{out}; the main results do not require committing to this particular optimizer model.

Corollary 8 (One standard PL-SGD plug-in for the outer residual)**.** *Suppose the noisy risk $\tilde{R}$ is L_ρ-smooth and satisfies the PL inequality with constant $0 < \mu_\rho \le L_\rho$. Suppose the stochastic gradients are unbiased and have variance at most σ^2_{sgd}:*

$$\mathbb{E}[g_t \mid \theta_t] = \nabla\tilde{R}(\theta_t), \qquad \mathbb{E}\left\|g_t - \nabla\tilde{R}(\theta_t)\right\|^2 \le \sigma^2_{\mathrm{sgd}}.$$

Run outer SGD

$$\theta_{t+1} = \theta_t - \eta g_t, \qquad 0 < \eta \le 1/L_\rho.$$

Then

$$\mathbb{E}\big[\tilde{R}(\theta_T) - \tilde{R}^*\big] \le (1-\eta\mu_\rho)^T\big(\tilde{R}(\theta_0) - \tilde{R}^*\big) + \frac{L_\rho\eta\sigma^2_{\mathrm{sgd}}}{2\mu_\rho} =: \mathcal{E}_{\mathrm{out}}(T).$$

Consequently, under Assumption 1,

$$\mathbb{E}\big[R(\theta_T) - R^*\big] \le \frac{\big(\sqrt{2L_\rho\mathcal{E}_{\mathrm{out}}(T)} + (1-\rho(\varepsilon))\Delta_g\big)^2}{2\mu_R}.$$

Proof. By L_ρ-smoothness of $\tilde{R}$ and the SGD update,

$$\begin{aligned}
\mathbb{E}[\tilde{R}(\theta_{t+1}) \mid \theta_t] &\le \tilde{R}(\theta_t) - \eta\left\|\nabla\tilde{R}(\theta_t)\right\|^2 + \frac{L_\rho\eta^2}{2}\mathbb{E}[\|g_t\|^2 \mid \theta_t] \\
&\le \tilde{R}(\theta_t) - \eta\left(1 - \frac{L_\rho\eta}{2}\right)\left\|\nabla\tilde{R}(\theta_t)\right\|^2 + \frac{L_\rho\eta^2\sigma^2_{\mathrm{sgd}}}{2} \\
&\le \tilde{R}(\theta_t) - \frac{\eta}{2}\left\|\nabla\tilde{R}(\theta_t)\right\|^2 + \frac{L_\rho\eta^2\sigma^2_{\mathrm{sgd}}}{2},
\end{aligned}$$

where the last step uses $\eta \le 1/L_\rho$. The PL inequality for $\tilde{R}$ gives

$$\mathbb{E}[\tilde{R}(\theta_{t+1}) - \tilde{R}^* \mid \theta_t] \le (1-\eta\mu_\rho)\big(\tilde{R}(\theta_t) - \tilde{R}^*\big) + \frac{L_\rho\eta^2\sigma^2_{\mathrm{sgd}}}{2}.$$

Taking total expectation and unrolling the recursion yields the displayed bound $\mathcal{E}_{\mathrm{out}}(T)$.

It remains to translate this noisy-risk optimization gap into a clean-risk statement. For any L_ρ-smooth function with a global minimizer,

$$\left\|\nabla\tilde{R}(\theta)\right\|^2 \le 2L_\rho\big(\tilde{R}(\theta) - \tilde{R}^*\big).$$

Also, the same gradient-decomposition argument used in Theorem 7 gives the pointwise bound

$$R(\theta) - R^* \le \frac{\big(\left\|\nabla\tilde{R}(\theta)\right\| + (1-\rho(\varepsilon))\Delta_g\big)^2}{2\mu_R}.$$

Applying this at θ_T, taking expectations, and using $\mathbb{E}[X] \le \sqrt{\mathbb{E}[X^2]}$ with $X = \left\|\nabla\tilde{R}(\theta_T)\right\|$, we obtain

$$\begin{aligned}
\mathbb{E}[R(\theta_T) - R^*] &\le \frac{1}{2\mu_R}\mathbb{E}\left[\left(\left\|\nabla\tilde{R}(\theta_T)\right\| + (1-\rho(\varepsilon))\Delta_g\right)^2\right] \\
&\le \frac{\left(\sqrt{\mathbb{E}\left\|\nabla\tilde{R}(\theta_T)\right\|^2} + (1-\rho(\varepsilon))\Delta_g\right)^2}{2\mu_R} \\
&\le \frac{\big(\sqrt{2L_\rho\mathcal{E}_{\mathrm{out}}(T)} + (1-\rho(\varepsilon))\Delta_g\big)^2}{2\mu_R}.
\end{aligned}$$

This proves the plug-in bound. □

G Computational Resources

Experiments were run on a single NVIDIA L40S GPU with 48 GB memory. A single CIFAR-10 or CIFAR-100 LabelDP training run completed in under 0.5 hours, BN adaptation requires only forward passes, whereas TENT requires one backward pass through the network per test batch, even though it updates only BN affine parameters. We did not measure latency or memory overhead.