PROGRESSIVE TRUST CALIBRATION FOR NOISY CONTINUAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning under noisy supervision in continual environments presents a fundamental challenge that existing methods are ill-equipped to address. Label corruption and distribution shift co-evolve across tasks, so that one-shot static noise-removal rules cannot keep pace with the evolving stream. In this work, we revisit this problem through a fresh lens and argue that sample reliability should not be treated as a fixed binary attribute, but rather as a dynamic property that evolves alongside the model's growing semantic competence. Motivated by this insight, we propose Progressive Trust Calibration (PTC), a unified framework that reframes noisy-label continual learning as a process of sample trust evolution. PTC operates through three synergistic stages. First, Relational Trust Initialization anchors reliability estimation in the intrinsic geometry of a frozen pretrained representation space, measuring neighborhood label consistency and local feature compactness to produce stable, prediction-independent trust scores that are immune to early-stage classifier bias. Second, Prototype-guided Trust Refinement elevates this local assessment to a global semantic level by constructing class prototypes from verified reliable samples and identifying violations of category-level structure, capturing noisy instances that evade local detection. Third, Expert-driven Trust Evolution closes the loop by treating the noise pool not as a discard bin but as a reservoir of recoverable supervision in which a task-specific expert continuously reassesses suspicious samples as reliable evidence accumulates, progressively relabeling informative examples that were initially flagged as unreliable. Empirical results show that our approach outperforms all compared methods.

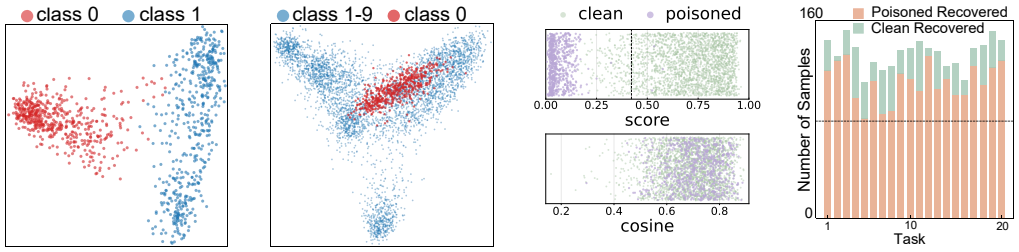
1 INTRODUCTION

Deep neural networks have achieved remarkable success in a wide range of applications, largely relying on large-scale datasets with reliable annotations Krizhevsky et al. (2012). However, in real-world scenarios, annotations are often corrupted due to human errors, automatic labeling systems, or imperfect data collection. Learning with noisy labels has therefore become a critical problem, where the central challenge is to identify unreliable supervision while preserving informative samples Fang et al. (2026).

Most existing noisy-label learning methods assume a static learning environment Fang et al. (2026); Karim et al. (2022), where the entire dataset is available and the noise distribution remains unchanged throughout training. Under this assumption, noisy samples can be detected by exploiting global statistics, prediction dynamics, or loss patterns accumulated over multiple epochs Li et al. (2020). However, such assumptions no longer hold in continual learning scenarios Zenke et al. (2017). In continual learning, data arrive as a sequence of tasks, where each task introduces new distributions and provides only limited observations. The model must simultaneously adapt to new knowledge, resist catastrophic forgetting, and handle potentially corrupted annotations. As a result, label noise and distribution shift become coupled over time, making noisy-label continual learning substantially more challenging than conventional noisy-label learning Li et al. (2026).

A fundamental difficulty in noisy-label continual learning is that the model cannot directly rely on its own predictions to determine label reliability Arpit et al. (2017). During the early stage of each task, the classifier is itself vulnerable to noisy supervision and distribution shifts Han et al. (2018a). Consequently, prediction-based confidence or loss-based filtering strategies may incorrectly trust easy-to-fit noisy samples while discarding difficult but informative examples. This raises a fundamental question:

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.



(a) (b) (c) (d)

Figure 1: **Motivation of the proposed framework.** (a) In a frozen representation space, semantic classes are naturally separated with limited boundary overlap. (b) As classes accumulate, inter-class overlap grows increasingly severe. (c) On the locally purified subset, the raw cosine similarity entangles clean and poisoned samples, while the softmax-normalized trust score yields a pronounced bimodal structure. (d) Per-task composition of removed samples. A non-negligible portion of clean samples is mistakenly removed, motivating progressive label correction. More experimental details are provided in the Appendix A.

How can a continual learner progressively estimate sample reliability without being biased by corrupted supervision, while preserving useful knowledge throughout the learning process?

We answer this question by arguing that noisy-label continual learning should be viewed as a hierarchical semantic evidence evolution process, rather than a binary sample selection problem that permanently separates samples into clean and noisy subsets. Reliability is not an intrinsic and fixed attribute of a sample. Instead, it evolves with the learner's growing semantic understanding. A sample that appears unreliable under limited knowledge may become trustworthy once the model acquires stronger task-specific representations, motivating a progressive calibration of sample trust throughout continual learning.

Our motivation is supported by three observations. First, pretrained representations provide a relatively stable semantic space before continual adaptation. In such a space, samples from different classes are naturally clustered with limited boundary overlap, while corrupted annotations appear as violations of local semantic relationships (Fig. 1(a),(b)). This suggests that neighborhood structures can provide an unbiased source for initial trust estimation without relying on noisy predictions.

Second, local consistency alone is insufficient as continual learning progressively introduces more classes and increases semantic overlap. We observe that after removing locally inconsistent samples, reliable samples exhibit strong agreement with their corresponding class semantics. However, directly measuring cosine similarity to class prototypes still fails to separate clean and corrupted samples, as every sample is closer to some prototype. By normalizing similarities across all prototypes, the probability assigned to the observed label provides a more discriminative trust measure, producing a clear bimodal distribution between reliable and suspicious samples, as empirically demonstrated in Fig. 1(c).

Third, samples removed by filtering strategies are not necessarily useless. Since label corruption affects annotations rather than visual content, many discarded samples remain semantically meaningful. Moreover, continual learning prevents revisiting previous tasks, making irreversible removal particularly harmful. As shown in Fig. 1(d), a considerable portion of clean samples may be mistakenly discarded, motivating a mechanism that progressively recovers valuable supervision as stronger semantic evidence becomes available.

Motivated by these observations, we propose Progressive Trust Calibration (PTC), which formulates noisy continual learning as a hierarchical semantic evidence evolution problem. Rather than assuming that all supervision signals are equally reliable, PTC progressively transitions from intrinsic semantic geometry, to global category semantics, and finally to task-adaptive semantic competence. Trust calibration is therefore not a one-shot filtering operation, but an evidence-driven trust update process.

First, based on the results from Fig. 1(a),(b), PTC introduces **Relational Trust Initialization (RTI)** that estimates initial trust from the intrinsic geometry of a frozen pretrained representation space. By constructing semantic neighborhoods and measuring neighborhood label consistency and local feature compactness, this stage provides prediction-independent trust estimation that is robust against early classifier bias. Second, inspired by the results from Fig. 1(c), PTC introduces

Prototype-guided Trust Refinement (PGTR) at the second stage, which performs global semantic verification using class prototypes estimated from trusted samples. This stage identifies samples that violate category-level semantic structures but may escape local neighborhood detection. Third, according to the results from Fig. 1(d), PTC introduces **Expert-driven Trust Evolution (EDTE)** at the third stage, which treats suspicious samples as recoverable supervision rather than permanently discarded data. As the continual learner becomes increasingly competent, task-specific experts dynamically reassess these samples using prediction confidence and prototype consistency, progressively recovering informative examples.

Our contributions are summarized as follows: (1) We formulate noisy continual learning as a hierarchical semantic evidence evolution problem, where sample trust is treated as a latent variable continuously updated by increasingly reliable semantic evidence; (2) We introduce the Trust Evolution Principle, which states that trust estimation should always rely on the most reliable semantic evidence currently available; (3) We instantiate this principle through Progressive Trust Calibration, consisting of relational initialization, prototype-guided refinement, and expert-driven trust evolution; (4) Extensive experiments verify that evidence evolution provides robust trust estimation under simultaneously evolving label corruption and task distributions.

2 Related Work

Continual Learning aims to progressively accumulate knowledge while mitigating catastrophic forgetting, typically via parameter regularization, experience replay, or architectural expansion Tong et al. (2025); Wang et al. (2025). More recently, Parameter-Efficient Fine-Tuning (PEFT) has emerged as a promising paradigm by freezing the backbone and optimizing lightweight task-specific adapters He et al. (2025); Luo et al. (2026). However, these approaches assume that the incoming training streams are clean, which is easily violated by data poisoning attacks that deliberately corrupt annotations and introduce persistent representation drift across tasks Nguyen et al. (2025). As a result, label corruption and distribution shift co-evolve, making noisy continual learning far harder than conventional noisy-label learning.

Robust Learning. Traditional robust learning methods rely on static defenses and fail in noisy continual learning Fang et al. (2026). Prediction- or loss-based filters over-trust easy-to-fit corrupted samples while discarding difficult but informative ones, making one-shot removal with fixed thresholds ineffective Han et al. (2018b); Li et al. (2026); Karim et al. (2022). Since corrupted annotations typically violate semantic structures preserved by frozen pre-trained backbones Momeni et al. (2025), our framework derives prediction-independent trust scores, adaptively calibrates thresholds without prior noise-ratio knowledge, and progressively recovers informative samples from the suspicious pool. More discussions are provided in **Appendix** C.

3 Methodology

3.1 Problem Formulation

Formally, the continual learning process consists of a sequence of tasks $\{\mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_M\}$. For the j-th task, the observed training dataset is denoted as $\mathcal{D}_j = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_j}$, where $\mathbf{x}_i$ denotes an input sample, $y_i \in \{1, \cdots, C_j\}$ is its observed label, N_j is the number of samples in task $\mathcal{T}_j$, and C_j is the number of classes in task $\mathcal{T}_j$. Unlike the standard supervised setting, we assume that a subset of labels has been corrupted, meaning that the observed annotation y_i may not coincide with the latent semantic category of $\mathbf{x}_i$.

3.2 Overview of Progressive Trust Calibration

Existing noisy-label learning methods typically estimate sample reliability only once before or during training. While such one-shot estimation is effective in conventional offline learning with stationary data distributions, it becomes insufficient in continual learning, where semantic representations and decision boundaries evolve across tasks and a one-shot estimate cannot keep pace. Consequently, sample reliability should be regarded as a dynamic quantity rather than a fixed property.

To address this challenge, we propose **Progressive Trust Calibration (PTC)**, a unified framework that formulates noisy continual learning as a progressive trust evolution process. Rather than relying on a single criterion, PTC progressively refines sample trust using increasingly reliable semantic evidence. The key principle is that *sample trust should evolve together with semantic understanding*. Early in training, prediction confidence is unreliable because the classifier is affected by noisy supervision. Therefore, trust is first initialized using intrinsic semantic structures extracted from a

frozen pretrained encoder. As more reliable samples are identified, global class prototypes provide stronger semantic verification. Finally, after sufficient purified supervision has been accumulated, a task-specific expert becomes a reliable semantic estimator, enabling dynamic trust evolution through iterative reassessment.

Accordingly, PTC consists of three stages, as shown in Fig. 2. **Stage I** initializes sample trust by measuring local semantic consistency in a frozen feature space. **Stage II** refines the initial trust estimation through prototype-guided global semantic verification. **Stage III** further evolves sample trust using prediction confidence and dynamically updated expert prototypes, progressively recovering informative samples that were conservatively filtered during the previous stages. Overall, the three stages correspond to increasingly reliable semantic evidence, namely *local relations*, *global prototypes*, and *task-specific experts*. Their progressive integration enables robust trust estimation throughout continual learning while effectively reducing the influence of noisy annotations.

3.3 Relational Trust Initialization

The first challenge of noisy continual learning is the lack of reliable supervision during early training, as classifiers directly optimized on corrupted annotations quickly become biased toward noisy labels. To overcome this, we leverage the stable semantic representations of vision foundation models and initialize sample trust from intrinsic semantic geometry in the frozen feature space, where correctly labeled samples form locally consistent and compact neighborhoods, while noisy samples violate neighborhood agreement or feature compactness. This provides robust supervision prior to classifier adaptation.

Frozen Semantic Representation. To prevent noisy supervision from influencing the initial trust estimation, we employ a frozen visual foundation encoder $F_\theta(\cdot)$ to extract semantic representations. Since all encoder parameters remain fixed throughout Stage I, the resulting feature space is unaffected by task-specific annotation corruption. For each input sample $\mathbf{x}_i$, its feature representation is obtained as

$$\mathbf{z}_i = \frac{F_\theta(\mathbf{x}_i)}{\|F_\theta(\mathbf{x}_i)\|_2}, \tag{1}$$

where $\mathbf{z}_i \in \mathbb{R}^d$ denotes the normalized semantic embedding and d is the embedding dimension. The ℓ_2 normalization projects all features onto the unit hypersphere, allowing cosine similarity to directly measure semantic affinity. Based on the extracted embeddings, we construct a semantic neighborhood graph. For each sample, its neighborhood is defined by the k nearest samples according to cosine similarity,

$$\mathcal{N}_i = \mathrm{TopK}_k\left(\{\cos(\mathbf{z}_i, \mathbf{z}_j)\}_{j \neq i}\right), \tag{2}$$

where $\mathrm{TopK}_k(\cdot)$ returns the indices of the k most similar samples. Unlike prediction-based graphs, the constructed semantic graph depends solely on intrinsic feature geometry. The neighborhood structure itself is independent of classifier predictions, although trust estimation still incorporates observed labels.

Although semantic proximity provides valuable information, feature similarity alone is insufficient for distinguishing difficult noisy samples. For example, samples belonging to visually similar categories may still reside close to each other in the embedding space, while boundary samples naturally exhibit lower neighborhood similarity despite having correct annotations. Therefore, we estimate trust from two complementary relational perspectives, namely *semantic agreement* and *local compactness*. The former evaluates whether the observed annotation is consistent with neighboring samples, whereas the latter measures the structural coherence of the local feature manifold.

Semantic Agreement. Given the neighborhood $\mathcal{N}_i$, the semantic agreement score is defined as

$$a_i = \frac{1}{k} \sum\nolimits_{j \in \mathcal{N}_i} \mathbb{I}(y_i = y_j), \tag{3}$$

where $\mathbb{I}(\cdot)$ is the indicator function. The score a_i quantifies the proportion of neighboring samples sharing the same observed label. A larger value indicates that the current annotation is well supported by its local semantic neighborhood, whereas a smaller value suggests potential semantic inconsistency. Nevertheless, semantic agreement alone cannot distinguish samples located inside dense regions shared by multiple visually similar categories. Therefore, additional structural evidence is required.

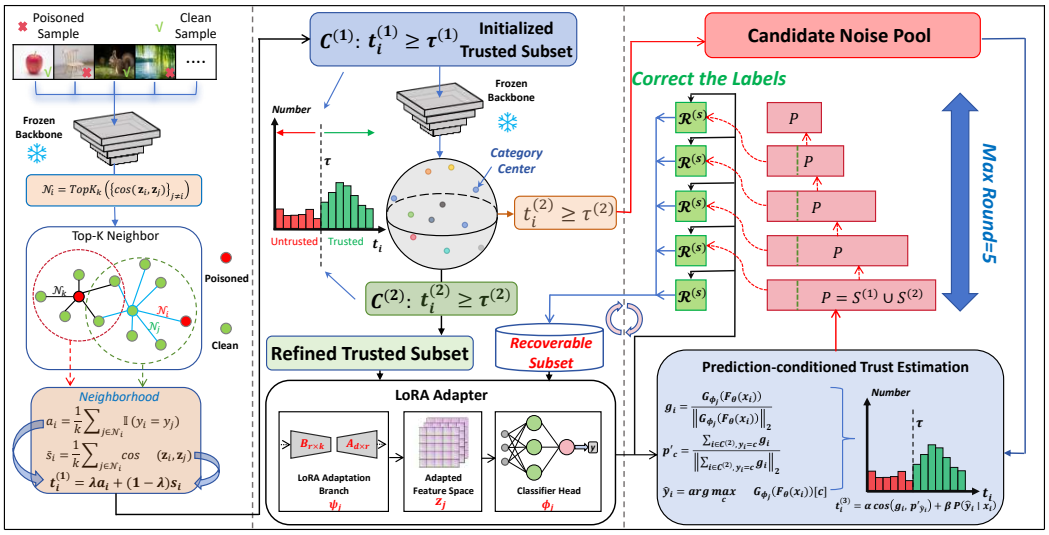


Figure 2: **Overview of Progressive Trust Calibration (PTC).** We reformulate noisy-label continual learning as progressive trust evolution. **(A) Relational Trust Initialization** estimates prediction-independent trust from a frozen representation space, fusing nearest-neighbor label agreement and task-normalized neighborhood compactness into a single trust score that an adaptive threshold splits into the trusted subset and the suspicious subset. **(B) Prototype-guided Trust Refinement** verifies the trusted subset against class prototypes via softmax-normalized prototype similarity, yielding a refined trusted subset and the candidate noise pool that merges the two suspicious subsets. **(C) Expert-driven Trust Evolution** reassesses the noise pool with a task-specific expert, whose prediction-conditioned trust score combines expert confidence with prototype agreement under the predicted class, recovering the informative samples filtered out earlier.

Neighborhood Compactness. Besides label agreement, we further evaluate the intrinsic compactness of the local semantic manifold. The average neighborhood similarity is computed as

$$\bar{s}_i = \frac{1}{k} \sum\nolimits_{j \in \mathcal{N}_i} \cos(\mathbf{z}_i, \mathbf{z}_j), \tag{4}$$

which reflects how tightly the sample is connected to its surrounding semantic region. Since similarity distributions may differ across continual tasks, we normalize the compactness score within each task,

$$s_i = \frac{\bar{s}_i - \min_j \bar{s}_j}{\max_j \bar{s}_j - \min_j \bar{s}_j}. \tag{5}$$

Compared with raw cosine similarity, the normalized score provides a task-invariant measure of local structural reliability.

Trust Fusion. The two relational views capture complementary properties of semantic reliability. Semantic agreement evaluates whether the observed annotation is consistent with neighboring labels, while neighborhood compactness measures whether the sample lies inside a reliable semantic manifold. Since neither criterion alone is sufficient, we combine both evidences into a unified trust score, expressed as:

$$t_i^{(1)} = \lambda a_i + (1 - \lambda) s_i, \tag{6}$$

where $\lambda \in [0, 1]$ balances semantic agreement and structural compactness. Importantly, $t_i^{(1)}$ should not be interpreted as prediction confidence. Instead, it measures the degree to which the current annotation agrees with intrinsic semantic relations extracted from the frozen representation space. Consequently, samples with large trust scores are more likely to possess reliable annotations, whereas low-trust samples either violate neighborhood consensus or reside in unreliable semantic regions.

Adaptive Trust Calibration. A practical difficulty in noisy continual learning is that the contamination ratio is usually unknown and varies considerably across tasks. Consequently, manually specifying the expected noise rate or using a fixed trust threshold is impractical in real-world continual

learning scenarios. Instead of assuming prior knowledge of the corruption distribution, we formulate trust partition as an unsupervised distribution separation problem. The fundamental assumption is that trustworthy and suspicious samples naturally induce two distinguishable modes in the trust distribution. Therefore, an appropriate trust boundary should maximize the statistical separation between these two populations. Given the trust scores $\{t_i^{(1)}\}_{i=1}^N$, we determine the optimal threshold by maximizing the between-class variance, expressed as:

$$\tau^{(1)} = \arg\max_{t \in \{t_i^{(1)}\}_{i=1}^N} \left\{\sigma_B^2(t)\right\}, \tag{7}$$

where $\sigma_B^2(t)$ is defined as $\sigma_B^2(t) = \omega_l(t)\omega_h(t)\left(\mu_l(t) - \mu_h(t)\right)^2$. Here, ω_l and ω_h denote the proportions of low-trust and high-trust samples, while μ_l and μ_h represent their corresponding mean trust values. The resulting threshold automatically adapts to the trust distribution of each continual task, eliminating the need for manually estimating the contamination ratio. Finally, the initialized trusted subset and suspicious subset are obtained as $\mathcal{C}^{(1)} = \{\mathbf{x}_i \mid t_i^{(1)} \geq \tau^{(1)}\}$ and $\mathcal{S}^{(1)} = \{\mathbf{x}_i \mid t_i^{(1)} < \tau^{(1)}\}$. Unlike conventional noise filtering, samples assigned to $\mathcal{S}^{(1)}$ are not permanently discarded. Instead, they are regarded as uncertain samples whose trust will be further reassessed in subsequent stages using increasingly reliable semantic evidence. Therefore, Stage I only provides a conservative initialization of sample trust, laying the foundation for the progressive trust evolution framework.

3.4 Prototype-guided Trust Refinement

Although relational trust initialization identifies samples violating local semantic consistency, local neighborhood evidence alone is insufficient for reliable trust estimation. In high-dimensional semantic spaces, mislabeled samples may remain locally consistent, while hard but correctly labeled boundary samples often appear unreliable. To address this limitation, we refine trust through global semantic verification, evaluating each sample against the prototype of its assigned category rather than its neighbors. Since class prototypes aggregate information from numerous reliable samples, they provide stable semantic references, making prototype consistency complementary global evidence for trust refinement.

Reliable Prototype Construction. Stage I provides an initial trusted subset $\mathcal{C}^{(1)}$, whose labels exhibit substantially higher reliability than the original training data. Although a few noisy samples may remain, most corrupted annotations have already been removed, so these trusted samples provide an appropriate basis for estimating class-level semantic representations. For each observed category c, we construct a prototype by averaging normalized semantic embeddings of trusted samples,

$$\mathbf{p}_c = \frac{\sum_{i \in \mathcal{C}^{(1)},\, y_i = c} \mathbf{z}_i}{\left\| \sum_{i \in \mathcal{C}^{(1)},\, y_i = c} \mathbf{z}_i \right\|_2}, \tag{8}$$

where $\mathbf{p}_c \in \mathbb{R}^d$ denotes the normalized prototype of class c. Unlike neighborhood representations, which only describe local semantic relations, the prototype summarizes the dominant semantic direction of an entire category. Consequently, it is considerably more robust to local feature fluctuations and isolated noisy samples. Nevertheless, prototype estimation becomes unstable when only a few trusted samples are available. To ensure reliable semantic statistics, prototype construction is performed only when the number of trusted samples satisfies

$$n_c = \left| \left\{ i \in \mathcal{C}^{(1)} \mid y_i = c \right\} \right| \geq n_{\min}, \tag{9}$$

where $n_{\min}$ denotes the minimum number of trusted samples required for prototype estimation. Categories with insufficient reliable observations are excluded from Stage II and will instead be handled during the subsequent expert-driven trust evolution stage.

Global Semantic Verification. Once reliable prototypes have been established, sample trust can be evaluated from a global semantic perspective. For each sample, we compute its similarity to every class prototype, expressed as $l_{i,c} = (\mathbf{z}_i^\top \mathbf{p}_c)/T$, where T is the temperature parameter controlling distribution smoothness. The prototype similarities are normalized into a semantic probability distribution,

$$q_{i,c} = \frac{\exp(l_{i,c})}{\sum_{c'} \exp(l_{i,c'})}. \tag{10}$$

Different from Stage I, which measures trust through neighborhood consistency, Stage II directly evaluates whether the observed annotation agrees with the global semantic center of its corresponding category. Accordingly, the prototype trust score is defined as $t_i^{(2)} = q_{i,y_i}$, where $t_i^{(2)}$ denotes the probability assigned to the observed label. A large trust score indicates that the sample is well aligned with the semantic prototype of its assigned category, whereas a low score suggests that the feature representation is more compatible with another semantic concept. Unlike local neighborhood relations, prototype verification is insensitive to individual neighboring samples, making it substantially more robust for distinguishing visually similar categories and boundary samples.

Progressive Trust Refinement. The trust scores obtained from prototype verification are further calibrated using the same adaptive trust calibration operator introduced in Stage I, $\tau^{(2)} = \Gamma(\{t_i^{(2)}\}_{i=1}^N)$, where $\Gamma(\{t_i^{(2)}\}_{i=1}^N) = \arg\max_{t \in \{t_i^{(2)}\}_{i=1}^N} \sigma_B^2(t)$ denotes the adaptive trust calibration operator that automatically partitions the trust distribution into reliable and suspicious groups. The refined suspicious subset is obtained as $\mathcal{S}^{(2)} = \{\mathbf{x}_i \in \mathcal{C}^{(1)} \mid t_i^{(2)} < \tau^{(2)}\}$ while the refined trusted subset becomes $\mathcal{C}^{(2)} = \mathcal{C}^{(1)} \setminus \mathcal{S}^{(2)}$. Instead of treating Stage II as an independent noise detector, we regard it as a refinement operator acting upon the trust estimation produced by Stage I. Only samples that remain semantically inconsistent under both local relational evidence and global prototype evidence are considered genuinely suspicious. Finally, the candidate noise pool is defined as $\mathcal{P} = \mathcal{S}^{(1)} \cup \mathcal{S}^{(2)}$, which contains all samples whose trust cannot be sufficiently supported by either local semantic relations or global semantic prototypes.

3.5 Expert-driven Trust Evolution

The first two stages estimate sample trust solely from fixed semantic evidence extracted from the frozen representation space. While providing reliable initialization and global semantic verification, their discriminative capability is limited by fixed representations. Consequently, samples from visually similar categories, fine-grained subclasses, or complex decision boundaries are conservatively assigned to the suspicious pool. However, these samples are not necessarily mislabeled. Their ambiguity mainly reflects insufficient task-specific knowledge. As continual learning progresses, the expert learns more discriminative decision boundaries, producing increasingly reliable predictions. This motivates a key insight: *sample trust should evolve with model competence rather than remain fixed after initialization.*

Accordingly, we introduce the final stage of Progressive Trust Calibration, where the continually adapted expert serves as an additional semantic evidence source. Instead of permanently discarding suspicious samples, their trust is periodically reassessed, enabling informative samples to be progressively recovered as the expert improves.

Prediction-conditioned Trust Estimation. Unlike the previous stages, trust estimation is no longer conditioned on the observed noisy annotation. Instead, the current expert provides a new semantic hypothesis through its prediction. For each continual task, a lightweight LoRA expert $G_{\phi_j}(\cdot)$ is instantiated upon the frozen backbone $F_\theta(\cdot)$. Only the LoRA parameters are optimized, while the backbone remains fixed, thereby preserving previously acquired knowledge and preventing catastrophic forgetting. The expert is trained using the trusted dataset $\mathcal{C}^{(2)}$ obtained after the first two trust refinement stages. The optimization objective is

$$\mathcal{L}_{\text{expert}} = -\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{C}^{(2)}}\Big[\sum\nolimits_{c=1}^{C}\big\{\mathbb{I}(y=c)\log G_{\phi_j}(F_\theta(\mathbf{x}))[c]\big\}\Big], \tag{11}$$

where $G_{\phi_j}(F_\theta(\mathbf{x}))[c]$ denotes the c-th dimension of the prediction made by the expert G_{ϕ_j} and C is the total number of categories. Compared with conventional supervised learning, the expert is optimized using progressively purified supervision, allowing it to acquire increasingly reliable category discrimination.

Dynamic Prototype Evolution. As the expert continuously improves, its latent representation gradually becomes more discriminative than the frozen semantic space. Therefore, instead of constructing prototypes from frozen features, Stage III dynamically updates class prototypes using expert representations. For each trusted sample, we have $\mathbf{g}_i = G_{\phi_j}(F_\theta(\mathbf{x}_i))/\|G_{\phi_j}(F_\theta(\mathbf{x}_i))\|_2$, where $\mathbf{g}_i$ denotes the normalized expert representation. The expert prototype of class c is then computed as

$$\mathbf{p}'_c = \frac{\sum_{i\in\mathcal{C}^{(2)},y_i=c}\mathbf{g}_i}{\Big\|\sum_{i\in\mathcal{C}^{(2)},y_i=c}\mathbf{g}_i\Big\|_2}. \tag{12}$$

Table 1: The performance of various models. "I" and "II" denote Attack I and Attack II, respectively. CIFAR-10 results are identical under both attacks, denoted by "I,II". The best result is in **bold**, and our method is highlighted with a shaded row.

Method	CIFAR-10$^{\mathrm{I,II}}$		CIFAR-100$^{\mathrm{I}}$		CUB-200$^{\mathrm{I}}$		CIFAR-100$^{\mathrm{II}}$		CUB-200$^{\mathrm{II}}$	
	Final	Avg	Final	Avg	Final	Avg	Final	Avg	Final	Avg
L2P	9.30±0.56	13.38±0.27	53.17±0.91	56.73±0.64	36.75±0.70	41.93±0.48	46.25±0.77	55.93±0.47	26.82±0.86	31.07±0.21
DER	53.21±0.25	75.30±0.57	9.84±1.54	66.06±0.23	2.91±0.55	43.55±0.36	10.34±1.32	83.79±0.50	8.97±0.59	50.07±0.22
DER++	50.87±0.74	67.89±0.18	11.54±0.65	66.66±1.05	10.04±0.49	50.01±0.34	9.72±0.63	82.26±0.80	3.23±0.45	43.64±0.49
DualPrompt	26.40±0.68	42.05±0.21	61.27±1.09	63.47±0.60	49.23±0.36	55.93±0.16	70.34±1.04	77.95±0.20	60.46±0.40	67.63±1.23
KeepLoRA	27.74±0.45	46.79±0.65	32.39±0.64	52.40±0.72	39.13±0.14	45.99±0.55	40.32±1.08	48.07±0.26	42.72±1.38	49.94±0.65
KeepLoRA+CT	75.85±0.60	84.51±0.37	72.59±0.35	80.43±0.31	60.90±1.66	65.73±0.50	72.97±0.88	81.80±0.24	63.15±0.68	70.00±0.39
CL-LoRA	80.90±0.41	87.85±0.19	80.50±0.77	89.09±0.63	62.33±0.44	77.62±0.43	53.15±0.61	66.82±0.45	61.19±0.29	77.07±0.40
SR^2-LoRA	10.01±0.06	32.37±0.08	66.69±3.09	73.49±2.81	62.16±1.39	63.16±1.23	82.03±0.35	85.80±1.99	80.87±0.98	86.23±1.43
PTC (Ours)	**97.32**±0.21	**98.46**±0.08	**87.16**±0.62	**90.66**±0.85	**78.37**±0.59	**83.58**±0.88	**87.70**±1.06	**91.31**±1.15	**85.71**±0.55	**90.57**±0.67

Unlike Stage II, these prototypes continuously evolve during optimization, thereby providing adaptive semantic references that become increasingly discriminative as learning progresses.

Prediction-conditioned Trust Evolution. For every sample $\mathbf{x}_i$ remaining in the candidate noise pool $\mathbf{x}_i \in \mathcal{P}$, the current expert predicts $\hat{y}_i = \arg\max_c G_{\phi_j}(F_\theta(\mathbf{x}_i))[c]$. Instead of evaluating trust with respect to the original noisy annotation, we measure whether the expert prediction itself is semantically supported. Specifically, the trust score is defined as

$$t_i^{(3)} = \alpha \cos(\mathbf{g}_i, \mathbf{p}'_{\hat{y}_i}) + \beta P(\hat{y}_i \mid \mathbf{x}_i), \tag{13}$$

where $P(\hat{y}_i \mid \mathbf{x}_i)$ denotes the expert confidence while $\cos(\mathbf{g}_i, \mathbf{p}'_{\hat{y}_i})$ measures prototype agreement under the predicted category. The two terms capture complementary evidence. Prediction confidence reflects local decision certainty, whereas prototype agreement evaluates global semantic consistency.

Progressive Trust Evolution. Unlike conventional relabeling methods, PTC never performs one-shot correction: sample trust evolves continuously together with expert capability. At the s-th optimization round, the updated trust scores are calibrated through the unified trust calibration operator $\tau^{(3)} = \Gamma(\{t_i^{(3)}\}_{i=1}^N)$ yielding the recoverable subset $\mathcal{R}^{(s)} = \{\mathbf{x}_i \in \mathcal{P} \mid t_i^{(3)} > \tau^{(3)}\}$. Each recovered sample is relabeled using the expert prediction, $\tilde{y}_i = \hat{y}_i$ and merged into the trusted training set, $\mathcal{C}^{(s+1)} = \mathcal{C}^{(s)} \cup \mathcal{R}^{(s)}$. The expert is subsequently re-optimized using the enlarged trusted dataset, which further improves semantic discrimination and enables additional samples to become recoverable. Consequently, sample trust, semantic prototypes, and expert capability evolve simultaneously, forming a positive feedback loop of progressive trust evolution. In addition, to avoid unnecessary re-optimization, EDTE suspends the iterative process once the trust scores become nearly identical.

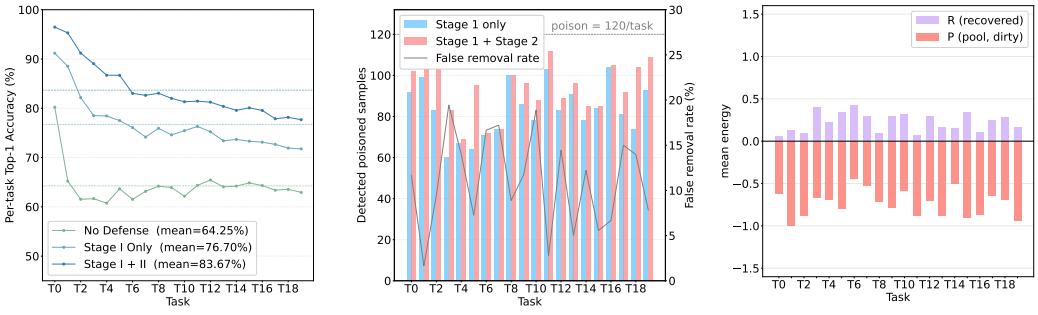
4 Experiments

Datasets and Attack Setup. We evaluate on Split CIFAR-10, Split CIFAR-100, and CUB-200 with 5, 10, and 20 sequential tasks of 2, 10, and 10 classes, respectively. Under CLDP, we inject pairwise and symmetric label-corruption attacks at a fixed 40% ratio. For Split CIFAR-10, where each task has only two classes, the two attacks are equivalent, and we report a unified result "I,II". Dataset and attack details are provided in Appendix C.2.

Baselines. We compare against standard CL methods (L2P Wang et al. (2022b), DER Yan et al. (2021), DER++ Buzzega et al. (2020), DualPrompt Wang et al. (2022a)), frozen-backbone PEFT competitors (CL-LoRA He et al. (2025), KeepLoRA Luo et al. (2026), SR^2-LoRA Wan et al. (2026)), and Co-teaching Han et al. (2018b) on KeepLoRA (KeepLoRA+CT). All share the frozen ViT-B/16 backbone; see Appendix C.1.

4.1 Main Results

Table 1 reports the final and average accuracy of PTC and all competitors on Split CIFAR-10, Split CIFAR-100, and CUB-200 under Attack I and Attack II. Across all five benchmark–attack combinations, PTC achieves the best result on both metrics: relative to the strongest competing method in each setting, it improves the final accuracy by $+16.42$, $+6.66$, $+16.04$, $+5.67$, and $+4.84$ points, and the average accuracy by $+10.61$, $+1.57$, $+5.96$, $+5.51$, and $+4.34$ points. This margin is consistent under both attacks, whereas existing methods degrade unevenly once annotations are corrupted. Such instability indicates that no single evidence source suffices in this setting, which is precisely what PTC addresses: rather than relying on predictions that are themselves corrupted, it



(a) (b) (c)

Figure 3: Effect of Progressive Trust Calibration on Split CUB-200 under the prescribed label-corruption attack. (a) Per-task Top-1 accuracy of a single representative run for No Defense, Stage I only, and Stage I+II. (b) Number of poisoned samples detected by Stage I only and by Stage I+II, together with the corresponding false removal rate of clean samples. (c) Mean energy of the candidate noise pool (P) before and the recovered subset (R) after recovery.

estimates sample reliability from intrinsic semantic relations and global category prototypes. Further analysis is provided in Appendices D.5 and D.

4.2 Ablation Study

Ablation Setup. All ablations are conducted on Split CUB-200 (20 sequential tasks of 10 classes) under the pairwise attack at a 40% corruption ratio, comparing three configurations that differ only in which part of PTC is activated: *No Defense*, *Stage I* (RTI), and *Stage I + II* (RTI + PGTR), all sharing the same frozen ViT-B/16 backbone. We additionally report a gradient-based energy in $[0, 1]$ that measures the deviation of a sample's update direction from the direction implied by its true category, with its full definition given in Appendix E.1.

Overall Effectiveness. Fig. 3(a) reports the per-task Top-1 accuracy of a single representative run: undefended training reaches a task-averaged accuracy of 64.25%, Stage I raises it to 76.70%, and Stage I + II to 83.67%, i.e., $+19.42$ points over the baseline. Stage II contributes a further $+6.97$ points over Stage I, indicating that global prototype verification captures corrupted samples that the local criterion leaves inside the trusted set. Neither stage depends on an already reliable classifier, so trust can be estimated from the first epochs of a task.

Table 2: Ablation study on CUB-200. Final and average accuracy (%) for three random seeds with Stage III disabled (Det ✗) and enabled (Det ✓, the complete PTC); Δ denotes the Stage III gain and Mean is the average over the three seeds.

	Final(%)			Average(%)		
Seed	Det ✗	Det ✓	Δ	Det ✗	Det ✓	Δ
1	77.68	79.05	**+1.37**	83.67	84.40	**+0.73**
3	76.27	78.08	**+1.81**	82.90	83.69	**+0.79**
1993	75.16	77.98	**+2.82**	80.10	82.65	**+2.55**
Mean	76.37	78.37	**+2.00**	82.22	83.58	**+1.36**

Limitation of Local Evidence. Fig. 3(b) reports the number of poisoned samples detected per task against the 120 injected, together with the false removal rate. Under a pairwise attack the two swapped classes are semantically related, so flipped samples form a compact region and are mutually nearest neighbours: the corruption is self-consistently endorsed by local geometry, which is why Stage I alone is insufficient. Class prototypes summarize the dominant semantic direction of a category rather than individual neighbours. Stage II thus recovers the recall that local evidence cannot provide, while the clean samples conservatively assigned to $S^{(1)}$ remain pooled in P and motivate Stage III.

Effect of Corrected Supervision. Stage III never trains on corrupted annotations. Fig. 3(c) shows that the recovered subset R has lower gradient-based energy than the candidate pool P, indicating more reliable supervision after relabeling. As shown in Table 2, this improves the final accuracy from 76.37% to 78.37% and the average accuracy from 82.22% to 83.58% across all random seeds.

5 Conclusion

We proposed PTC, a progressive trust calibration framework for noisy continual learning. By combining relational initialization, prototype-guided refinement, and expert-driven trust evolution, PTC progressively estimates and updates sample reliability throughout continual learning. Experimental results demonstrate consistent improvements over existing methods across multiple datasets and corruption settings.

AI USE STATEMENT

Generative AI tools were used for language editing. All technical claims, implementations, experiments, and reported numbers were verified by the authors.

REFERENCES

Devansh Arpit, Stanisław Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pp. 233–242, 2017.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020.

Jingzhong Fang, Zidong Wang, Weibo Liu, Yong Zhang, Yimeng He, and Xiaohui Liu. A comprehensive review on deep learning with noisy labels: Challenges, solutions, and frontiers. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2026.

Bo Han, Jiangchao Yao, Gang Niu, Mingyuan Zhou, Ivor W. Tsang, Ya Zhang, and Masashi Sugiyama. Masking: A new perspective of noisy supervision. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 5841–5851, 2018a. URL https://proceedings.neurips.cc/paper/2018/hash/aee92f16efd522b9326c25cc3237ac15-Abstract.html.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in Neural Information Processing Systems*, volume 31, pp. 8535–8545, 2018b.

Jiangpeng He, Zhihao Duan, and Fengqing Zhu. CL-LoRA: Continual low-rank adaptation for rehearsal-free class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 30534–30544, 2025.

Nazmul Karim, Umar Khalid, Ashkan Esmaeili, and Nazanin Rahnavard. CNLL: A semi-supervised approach for continual noisy label learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2022, New Orleans, LA, USA, June 19-20, 2022*, pp. 3877–3887. IEEE, 2022. doi: 10.1109/CVPRW56347.2022.00433. URL https://doi.org/10.1109/CVPRW56347.2022.00433.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Inf. Proc. Systems (NIPS)*, pp. 1097–1105, 2012.

Junnan Li, Richard Socher, and Steven C. H. Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations*, 2020.

Yanhong Li, Zhiqing Guo, and Liejun Wang. Cltr: Continual learning time-varying regularization for robust classification of noisy label images. *Pattern Recognition*, 171:112137, 2026.

Mao-Lin Luo, Zi-Hao Zhou, Yi-Lin Zhang, Yuanyu Wan, Tong Wei, and Min-Ling Zhang. Keeplora: Continual learning with residual gradient adaptation. *CoRR*, abs/2601.19659, 2026. doi: 10.48550/ARXIV.2601.19659. URL https://doi.org/10.48550/arXiv.2601.19659.

Saleh Momeni, Sahisnu Mazumder, and Bing Liu. Continual learning using a kernel-based method over foundation models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 19528–19536, 2025.

Trang Nguyen, Anh Tran, and Nhat Ho. Attack on prompt: Backdoor attack in prompt-based continual learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 19686–19694, 2025.

Ruilin Tong, Yuhang Liu, Javen Qinfeng Shi, and Dong Gong. Coreset selection via reducible loss in continual learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

Fengqiang Wan, Yipeng Lin, Kan Lv, and Yang Yang. SR^2-LoRA: Self-rectifying inter-layer relations in low-rank adaptation for class-incremental learning. *arXiv preprint arXiv:2605.07420*, 2026.

Huiyi Wang, Haodong Lu, Lina Yao, and Dong Gong. Self-expansion of pre-trained models with mixture of adapters for continual learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 10087–10098, 2025.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer G. Dy, and Tomas Pfister. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *ECCV*, volume 13686, pp. 631–648, 2022a.

Zifeng Wang, Zizhao Zhang, Lee, Han Zhang, Ruoxi Sun, et al. Learning to prompt for continual learning. In *CVPR*, pp. 139–149, 2022b.

Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *CVPR*, pp. 3014 – 3023, 2021.

F. Zenke, B. Poole, and S. Ganguli. Continual learning through synaptic intelligence. In *Proc. of International Conference on Machine Learning, vol. PLMR 70*, pp. 3987–3995, 2017.

A Experimental Details for the Motivation Figure

All analyses in Fig. 1 use the same frozen ViT-B/16 encoder as the main experiments, and follow the datasets, task splits and corruption protocols described in Appendix A.1. The encoder is never updated, so the feature geometry is by construction independent of annotation corruption. Ground-truth labels are used only for colouring the panels and for computing the qualitative diagnostics below; they are never accessed by PTC at any stage.

Fig. 1(a) Panel (a) shows a representative pair of semantically distinct classes drawn from the class incremental split, visualized with t-SNE using cosine distance. The two classes occupy compact regions with limited boundary overlap, and both the mean silhouette score and the ratio of inter class to intra-class mean distance are consistent with well-separated geometry. Since t-SNE is only a qualitative rendering, we verify the same separation directly in the original embedding space and report the silhouette score computed there as well. The analysis is repeated over the three random seeds used in the main tables, and the separation we report is stable across all of them.

Fig. 1(b) Panel (b) uses an identical pipeline and differs from panel (a) only in the number of classes placed in the shared space, ranging from a small subset to the full set of classes encountered across the task sequence. As classes accumulate, the silhouette score decreases steadily and the fraction of samples whose nearest neighbour belongs to a different class increases, which quantifies the growth of inter-class overlap that the panel illustrates. The same monotone trend is observed in the original embedding space, confirming that it is a property of the representation rather than an artefact of dimensionality reduction.

Fig. 1(c) The locally purified subset is the trusted set $C^{(1)}$ returned by Stage I on CUB-200 under the pairwise attack at a 40% corruption ratio, i.e. the setting in which local evidence is weakest. For each sample we compare the raw cosine similarity to the prototype of its observed label, $\cos(z_i, p_{y_i})$, with the softmax-normalized score $t_i^{(2)} = q_{i,y_i}$. Histograms use 30 equally spaced bins, and ground-truth labels are used only for colouring the two populations. The raw similarity yields two heavily

entangled modes, whereas the normalized score separates them into a pronounced bimodal distribution. We quantify this separation with the overlap coefficient and the AUC between the clean and poisoned populations, and we verify that the valley of the normalized histogram coincides with the adaptive threshold $\tau^{(2)}$ obtained by maximizing the between-class variance, which confirms that the criterion of Eq. (7) falls on the saddle of the distribution rather than being imposed on it.

Fig. 1(d) We report the per-task composition of the removed set $S^{(1)}$ on the same benchmark and under the same attack as panel (c), where 40% of the samples of each task are corrupted. A removed sample is counted as poisoned if its ground-truth label differs from the observed one, and as clean otherwise; the two legend entries correspond to these two categories, and we additionally mark the samples that Stage III subsequently reclaims and relabels. Tasks 1, 10 and 20 are shown as representatives of the early, middle and late stages of the sequence, and the complete curve over all twenty tasks is given in Fig. 3(b). Averaged over tasks and seeds, a non-negligible fraction of the removed samples is in fact clean, which indicates that permanent removal would discard a substantial amount of valid supervision and directly motivates the recovery mechanism of Stage III.

B Theoretical Analysis

In this section, we provide theoretical justifications for the three core components of Progressive Trust Calibration (PTC): the reliability of relational trust initialization under a frozen semantic encoder, the consistency of prototype-guided trust refinement, and the convergence of expert-driven trust evolution.

B.1 Reliability of Relational Trust Initialization

We first formalize the statistical separation property of the trust score $t_i^{(1)}$ defined in Stage I, and show that under mild conditions the adaptive trust calibration recovers a near-optimal partition of clean and noisy samples.

Setup. Let each sample $\mathbf{x}_i$ be either *clean* (event $\mathcal{H}_0$) or *noisy* (event $\mathcal{H}_1$). We model the trust scores as two sub-populations:

$$t_i^{(1)} \sim \begin{cases} \mathcal{D}_0 & \text{if } \mathbf{x}_i \in \mathcal{H}_0, \\ \mathcal{D}_1 & \text{if } \mathbf{x}_i \in \mathcal{H}_1, \end{cases} \tag{14}$$

with means $\mu_0 > \mu_1$ and variances $\sigma_0^2, \sigma_1^2 < \infty$. Let $\pi_0 = \Pr(\mathcal{H}_0)$ and $\pi_1 = 1 - \pi_0 = \Pr(\mathcal{H}_1)$ denote the class proportions.

Assumption B.1 (Distributional Separation) *The two sub-population distributions $\mathcal{D}_0$ and $\mathcal{D}_1$ satisfy*

$$\Delta \;=\; \mu_0 - \mu_1 \;>\; \delta > 0 \tag{15}$$

for some constant δ that does not depend on N.

Assumption B.2 (Neighborhood Label Coherence) *Let $\rho < \frac{1}{2}$ denote the label noise rate. For any clean sample $\mathbf{x}_i$ with observed label $y_i = y_i^*$, its k-nearest neighbors in the frozen feature space contain at most $\lfloor \rho k \rfloor$ samples in expectation whose observed label differs from y_i^*. For any noisy sample $\mathbf{x}_i$ with observed label $y_i \neq y_i^*$, the neighbors whose observed label equals the corrupted label y_i form a minority of the neighbourhood, i.e., at most $\lfloor \rho k \rfloor$ in expectation.*

Assumption B.2 constrains the geometry of the frozen encoder rather than our conclusions. Its first clause states that a clean sample is not surrounded by corrupted annotations, which holds whenever the pretrained feature space is semantically consistent. Its second clause states that a corrupted annotation is not shared by the majority of its semantic neighbours—a direct consequence of $\rho < \frac{1}{2}$ together with the absence of systematic collusion between the encoder geometry and the attacker.

Proposition B.3 (Consistency of Semantic Agreement) *Under Assumption B.2, the expected semantic agreement score of a clean sample satisfies*

$$\mathbb{E}[a_i \mid \mathcal{H}_0] \;\geq\; 1 - \rho, \tag{16}$$

while for a noisy sample it satisfies

$$\mathbb{E}[a_i \mid \mathcal{H}_1] \;\leq\; \rho. \tag{17}$$

For a clean sample $\mathbf{x}_i$ with true label y_i^*, its observed label equals $y_i = y_i^*$. By Assumption B.2, among its k nearest neighbors, at most $\lfloor \rho k \rfloor$ carry a corrupted label $y_j \neq y_j^*$. Consequently,

$$\mathbb{E}[a_i \mid \mathcal{H}_0] = \frac{1}{k} \sum_{j \in \mathcal{N}_i} \Pr(y_j = y_i \mid \mathcal{H}_0) \geq \frac{k - \lfloor \rho k \rfloor}{k} \geq 1 - \rho. \tag{18}$$

For a noisy sample $\mathbf{x}_i$, we have $y_i \neq y_i^*$. By the second clause of Assumption B.2, the fraction of neighbours carrying the corrupted label y_i is at most ρ in expectation, giving

$$\mathbb{E}[a_i \mid \mathcal{H}_1] \;\leq\; \rho. \tag{19}$$

When $\rho < \frac{1}{2}$, the two bounds are strictly separated, establishing $\mathbb{E}[a_i \mid \mathcal{H}_0] > \mathbb{E}[a_i \mid \mathcal{H}_1]$.

Theorem B.4 (Mean Separation of Trust Scores) *Under Assumptions B.1–B.2 and for any $\lambda \in (0, 1)$, the mean trust scores of clean and noisy samples satisfy*

$$\mathbb{E}[t_i^{(1)} \mid \mathcal{H}_0] - \mathbb{E}[t_i^{(1)} \mid \mathcal{H}_1] \;\geq\; \lambda(1 - 2\rho) - (1 - \lambda)\,\kappa, \tag{20}$$

where $\kappa = \big|\mathbb{E}[s_i \mid \mathcal{H}_0] - \mathbb{E}[s_i \mid \mathcal{H}_1]\big| \leq 1$ bounds the magnitude of the compactness gap, which may be of either sign. Consequently, whenever $\lambda > \kappa/(1 - 2\rho + \kappa)$, the two means are strictly separated.

By linearity of expectation,

$$\mathbb{E}[t_i^{(1)} \mid \mathcal{H}_0] - \mathbb{E}[t_i^{(1)} \mid \mathcal{H}_1] = \lambda\big(\mathbb{E}[a_i \mid \mathcal{H}_0] - \mathbb{E}[a_i \mid \mathcal{H}_1]\big) + (1 - \lambda)\big(\mathbb{E}[s_i \mid \mathcal{H}_0] - \mathbb{E}[s_i \mid \mathcal{H}_1]\big). \tag{21}$$

Applying Proposition B.3 to the first term and the bound $\mathbb{E}[s_i \mid \mathcal{H}_0] - \mathbb{E}[s_i \mid \mathcal{H}_1] \geq -\kappa$ to the second,

$$\geq \lambda(1 - 2\rho) - (1 - \lambda)\kappa. \tag{22}$$

The right-hand side is strictly positive exactly when $\lambda > \kappa/(1 - 2\rho + \kappa)$, since $\rho < \frac{1}{2}$ makes the denominator positive. We deliberately make no assumption about the sign of the compactness gap: neighbourhood compactness is a geometric property that may favour either population, so Theorem B.4 establishes a mean separation whose margin degrades gracefully with κ, rather than a stochastic dominance claim on the two trust distributions.

[Scope of the Separation Guarantee] Theorem B.4 is deliberately stated for *independent, label-independent* corruption, i.e. the symmetric regime of Attack II, in which the two clauses of Assumption B.2 hold simultaneously. Under the systematic pairwise corruption of Attack I the two swapped classes are strongly related, the flipped samples form a compact region and are mutually nearest neighbours, so the minority clause of Assumption B.2 may be violated and the Stage I margin in Theorem B.4 no longer applies. This is exactly the regime in which Stage I alone is insufficient (§4.2): the corruption is self-consistently endorsed by local geometry. We therefore claim the local guarantee for Attack II only, and treat Attack I as supported by the global prototype evidence of Theorem B.8 together with the empirical results in Appendix D, rather than by neighbourhood statistics.

Theorem B.5 (Optimality of Adaptive Trust Calibration) *Let $\{t_i^{(1)}\}_{i=1}^N$ be the empirical trust scores. The threshold selected by between-class variance maximization,*

$$\tau^{(1)} = \arg\max_t \, \sigma_B^2(t), \qquad \sigma_B^2(t) = \omega_l(t)\,\omega_h(t)\,\big(\mu_l(t) - \mu_h(t)\big)^2, \tag{23}$$

is equivalent to minimizing the total within-class variance $\sigma_W^2(t) = \omega_l(t)\sigma_l^2(t) + \omega_h(t)\sigma_h^2(t)$, and maximizes the Fisher discriminant ratio between the two partitions.

Observe that the total variance $\sigma_T^2 = \frac{1}{N}\sum_i (t_i^{(1)} - \bar{t})^2$ is independent of the threshold t. The standard variance decomposition gives

$$\sigma_T^2 = \sigma_W^2(t) + \sigma_B^2(t), \tag{24}$$

from which it follows immediately that

$$\arg\max_t \sigma_B^2(t) = \arg\min_t \sigma_W^2(t). \tag{25}$$

Moreover, the Fisher discriminant ratio is

$$J(t) = \frac{\sigma_B^2(t)}{\sigma_W^2(t)} = \frac{\sigma_T^2 - \sigma_W^2(t)}{\sigma_W^2(t)} = \frac{\sigma_T^2}{\sigma_W^2(t)} - 1, \tag{26}$$

which is a strictly monotone decreasing function of $\sigma_W^2(t)$. Hence maximizing $\sigma_B^2(t)$ simultaneously minimizes $\sigma_W^2(t)$ and maximizes the Fisher criterion $J(t)$, establishing the three-way equivalence.

Corollary B.6 (Consistency of Adaptive Thresholding) *Let the empirical trust scores be i.i.d. samples from the mixture $\pi_0\mathcal{D}_0 + \pi_1\mathcal{D}_1$. As $N \to \infty$, the empirical threshold $\hat{\tau}^{(1)}$ converges in probability to the population-optimal threshold τ^*,*

$$\hat{\tau}^{(1)} \xrightarrow{p} \tau^* \;=\; \arg\max_t \Pi_0(t)\,\Pi_1(t)\,\big(M_0(t) - M_1(t)\big)^2, \tag{27}$$

where $\Pi_r(t)$ and $M_r(t)$ denote the population proportion and mean of component r below/above threshold t, respectively.

The empirical weight $\hat{\omega}_l(t) = N^{-1}\sum_i \mathbf{1}[t_i^{(1)} \le t]$ converges uniformly to the population CDF $F(t)$ by the Glivenko–Cantelli theorem. Similarly, $\hat{\mu}_l(t)$ converges uniformly to its population counterpart by the law of large numbers applied uniformly over t (Donsker's theorem). Since $\sigma_B^2(t)$ is a continuous functional of these quantities, the continuous mapping theorem implies $\hat{\sigma}_B^2(t) \xrightarrow{p} \sigma_B^2(t)$ uniformly in t. Because τ^* is the unique maximizer of $\sigma_B^2(t)$ under Assumption B.1 (strict separation implies a unique maximum), the argmax is continuous at σ_B^2, and $\hat{\tau}^{(1)} \xrightarrow{p} \tau^*$ follows from the continuous mapping theorem.

B.2 Consistency of Prototype-guided Trust Refinement

We now analyze Stage II and show that the class prototypes constructed from the trusted subset $\mathcal{C}^{(1)}$ converge to the true semantic centroids, and that prototype trust scores correctly rank clean samples above noisy ones.

Assumption B.7 (Residual Noise Bound in $\mathcal{C}^{(1)}$) *After Stage I, the fraction of remaining noisy samples in the trusted subset $\mathcal{C}^{(1)}$ satisfies $\tilde{\rho} \le \epsilon$ for some $\epsilon \in [0, \frac{1}{2})$.*

Let $\mu_c^* = \mathbb{E}[\mathbf{z}_i \mid y_i^* = c]$ denote the true (normalized) class centroid in the frozen feature space, and let $\hat{\mathbf{p}}_c$ denote the empirical prototype constructed from $\mathcal{C}^{(1)}$.

Theorem B.8 (Prototype Consistency) *Under Assumption B.7, let $n_c = |\{i \in \mathcal{C}^{(1)} : y_i = c\}|$ and suppose the feature embeddings satisfy $\|\mathbf{z}_i\|_2 = 1$. Let $\cos\theta_{\max} \coloneqq \min_c \min_{i \in \mathcal{C}_c^+} \cos(\mathbf{z}_i, \mu_c^*) > 0$ denote the worst-case concentration of the trusted clean samples around their class centroid. Then*

$$\big\|\hat{\mathbf{p}}_c - \mu_c^*\big\|_2 \;\le\; \frac{2\epsilon}{\cos\theta_{\max}} + O\!\left(\frac{1}{\sqrt{n_c}}\right). \tag{28}$$

When the clean feature directions of a class are tightly concentrated, $\cos\theta_{\max} \to 1$ and the bound reduces to the familiar 2ϵ form.

Partition the trusted samples for class c into clean set $\mathcal{C}_c^+$ (of size n_c^+) and residual noisy set $\mathcal{C}_c^-$ (of size n_c^-), with $n_c = n_c^+ + n_c^-$ and $n_c^-/n_c \le \epsilon$.

The unnormalized prototype sum decomposes as

$$\mathbf{S}_c = \sum_{i \in \mathcal{C}^{(1)},\, y_i = c} \mathbf{z}_i = \underbrace{\sum_{i \in \mathcal{C}_c^+} \mathbf{z}_i}_{\mathbf{S}_c^+} + \underbrace{\sum_{i \in \mathcal{C}_c^-} \mathbf{z}_i}_{\mathbf{S}_c^-}. \tag{29}$$

Since $\|\mathbf{z}_i\|_2 = 1$ for all i, we have $\|\mathbf{S}_c^-\|_2 \le n_c^-$. By the law of large numbers, $\mathbf{S}_c^+/n_c^+ \to \mu_c^*$ with deviation $O(1/\sqrt{n_c^+})$. Writing $\tilde{\mathbf{S}}_c = \mathbf{S}_c/\|\mathbf{S}_c\|_2$, the triangle inequality gives

$$\left\|\tilde{\mathbf{S}}_c - \mu_c^*\right\|_2 \le \left\| \frac{\mathbf{S}_c}{\|\mathbf{S}_c\|_2} - \frac{\mathbf{S}_c^+}{\|\mathbf{S}_c^+\|_2} \right\|_2 + \left\| \frac{\mathbf{S}_c^+}{\|\mathbf{S}_c^+\|_2} - \mu_c^* \right\|_2. \tag{30}$$

For the first term, using the identity $\|\mathbf{u}/\|\mathbf{u}\| - \mathbf{v}/\|\mathbf{v}\|\|_2 \le 2\|\mathbf{u}-\mathbf{v}\|_2/\|\mathbf{v}\|_2$ for unit-norm targets, and $\|\mathbf{S}_c^+\|_2 \ge n_c^+ \cos\theta_{\max}$ where $\theta_{\max}$ is the maximal angular spread within the class (a class-specific constant), we obtain

$$\left\| \frac{\mathbf{S}_c}{\|\mathbf{S}_c\|_2} - \frac{\mathbf{S}_c^+}{\|\mathbf{S}_c^+\|_2} \right\|_2 \le \frac{2\|\mathbf{S}_c^-\|_2}{\|\mathbf{S}_c^+\|_2} \le \frac{2n_c^-}{n_c^+ \cos\theta_{\max}} \le \frac{2\epsilon}{(1-\epsilon)\cos\theta_{\max}} \le \frac{2\epsilon}{\cos\theta_{\max}} + O(\epsilon^2). \tag{31}$$

The concentration factor is retained rather than absorbed, because it is the quantity that makes the bound meaningful: as $\cos\theta_{\max} \to 0$ the trusted samples of a class are spread over the whole sphere, no prototype can be estimated, and the bound becomes vacuous. For the tightly clustered classes of our benchmarks $\cos\theta_{\max}$ is close to one, so the factor is numerically negligible while remaining formally present. The second term is $O(1/\sqrt{n_c^+}) = O(1/\sqrt{n_c})$ by the central limit theorem on unit sphere (cf. concentration on the hypersphere). Combining both terms yields the stated bound.

Corollary B.9 (Trust Score Ordering) *Suppose $\mu_0^* \ne \mu_1^*$ for distinct classes (i.e., the true centroids are separated) and that ϵ and $1/\sqrt{n_c}$ are sufficiently small. Then, for any clean sample $\mathbf{x}_i$ with true label c,*

$$\mathbb{E}[t_i^{(2)} \mid \mathcal{H}_0] > \mathbb{E}[t_i^{(2)} \mid \mathcal{H}_1], \tag{32}$$

i.e., clean samples receive higher prototype trust scores in expectation than noisy samples.

For a clean sample $\mathbf{x}_i$ with true label $y_i^* = c$, its feature $\mathbf{z}_i$ is drawn from the distribution centered at μ_c^*. By Theorem B.8, $\hat{\mathbf{p}}_c \approx \mu_c^*$ up to error $2\epsilon + O(1/\sqrt{n_c})$. Hence $\cos(\mathbf{z}_i, \hat{\mathbf{p}}_c) \approx \cos(\mathbf{z}_i, \mu_c^*)$, which is large for intra-class samples. For a noisy sample with observed label c but true label $c' \ne c$, its feature is centered at $\mu_{c'}^*$, so $\cos(\mathbf{z}_i, \hat{\mathbf{p}}_c) \approx \cos(\mu_{c'}^*, \mu_c^*)$, which is small when classes are semantically separated ($\|\mu_c^* - \mu_{c'}^*\|_2 > 0$). The softmax normalization in $t_i^{(2)} = q_{i,y_i}$ preserves this ordering: a larger logit $l_{i,c}$ for the true class amplifies the probability mass assigned to c relative to all other classes. Specifically, denoting the minimum inter-class centroid distance by $d_{\min} = \min_{c \ne c'} \|\mu_c^* - \mu_{c'}^*\|_2 > 0$, and using the approximation $\hat{\mathbf{p}}_c \approx \mu_c^*$:

$$\mathbb{E}[t_i^{(2)} \mid \mathcal{H}_0] - \mathbb{E}[t_i^{(2)} \mid \mathcal{H}_1] \ge \frac{e^{1/T} - e^{(1-d_{\min}^2/2)/T}}{e^{1/T} + (C-1)e^{(1-d_{\min}^2/2)/T}} > 0, \tag{33}$$

where the bound follows from the cosine identity $\cos(\angle(\mu_c^*, \mu_{c'}^*)) = 1 - d_{\min}^2/2$ for unit vectors, and C denotes the number of classes.

B.3 Convergence of Expert-driven Trust Evolution

Stage III progressively recovers informative samples from the noise pool $\mathcal{P}$ by iteratively re-assessing trust using an evolving expert. We formalize the recovery process and prove that the trusted set expands monotonically, while the relabeling error rate diminishes across rounds.

Setup. Let $\mathcal{C}^{(s)}$ denote the trusted training set at round s, and let $\rho^{(s)}$ denote the fraction of incorrectly labeled samples in $\mathcal{C}^{(s)}$.

Assumption B.10 (Expert Improvement) *Let $\mathcal{L}^{(s)}$ denote the expected classification error of the expert trained on $\mathcal{C}^{(s)}$. There exists a constant $\gamma > 0$ such that*

$$\mathcal{L}^{(s+1)} \le \mathcal{L}^{(s)} - \gamma \cdot \left|\mathcal{R}^{(s)} \cap \mathcal{H}_0\right|/N, \tag{34}$$

i.e., each additional clean sample recovered improves expert accuracy by at least γ/N.

Assumption B.11 (Trust Score Monotonicity) *As expert classification error decreases, the expected trust score of any clean sample in $\mathcal{P}$ is non-decreasing:*

$$\mathcal{L}^{(s+1)} < \mathcal{L}^{(s)} \Rightarrow \mathbb{E}[t_i^{(3,s+1)} \mid \mathbf{x}_i \in \mathcal{P}, \mathcal{H}_0] \ge \mathbb{E}[t_i^{(3,s)} \mid \mathbf{x}_i \in \mathcal{P}, \mathcal{H}_0]. \tag{35}$$

Assumption B.11 is intuitive: as the expert becomes more accurate, its predictions on clean samples become more confident and more consistent with their true semantic prototypes, both of which increase $t_i^{(3)}$.

Theorem B.12 (Monotone Expansion of Trusted Set) *Under Assumptions B.10–B.11, the trusted set expands monotonically:*

$$\mathcal{C}^{(s)} \subseteq \mathcal{C}^{(s+1)}, \qquad \forall\, s \geq 2. \tag{36}$$

By construction, the recovery rule is

$$\mathcal{C}^{(s+1)} = \mathcal{C}^{(s)} \cup \mathcal{R}^{(s)}, \qquad \mathcal{R}^{(s)} = \{\mathbf{x}_i \in \mathcal{P} \mid t_i^{(3,s)} > \tau^{(3,s)}\}. \tag{37}$$

Once a sample enters $\mathcal{C}^{(s)}$, it is never removed (no sample is expelled from the trusted set). Hence $\mathcal{C}^{(s)} \subseteq \mathcal{C}^{(s+1)}$ holds by the union operation $\mathcal{C}^{(s+1)} = \mathcal{C}^{(s)} \cup \mathcal{R}^{(s)}$.

[What Theorem B.12 does and does not say] The inclusion $\mathcal{C}^{(s)} \subseteq \mathcal{C}^{(s+1)}$ follows directly from the union form of the recovery rule and is therefore a property of the algorithm rather than of the data; it is not an independent empirical claim. We state it explicitly because it is what makes the cardinality argument of Theorem B.14 valid. The substantive content of Stage III resides in the companion results: the newly admitted set $\mathcal{R}^{(s)}$ is predominantly clean (Theorem B.13), and the trust scores of the remaining candidates are non-decreasing as the expert improves (Assumption B.11).

Theorem B.13 (Decreasing Relabeling Error Rate) *Under Assumptions B.10–B.11, suppose the prediction-conditioned trust score satisfies the discrimination condition*

$$\mathbb{E}[t_i^{(3)} \mid \mathcal{H}_0] > \mathbb{E}[t_i^{(3)} \mid \mathcal{H}_1] + \eta \tag{38}$$

for some $\eta > 0$. Define the relabeling error rate of round s as the fraction of recovered samples whose final label is incorrect,

$$\xi_{\mathcal{R}}^{(s)} \;=\; \frac{\big|\{i \in \mathcal{R}^{(s)} : \tilde{y}_i \neq y_i^*\}\big|}{|\mathcal{R}^{(s)}|}. \tag{39}$$

Writing $\varepsilon_r^{(s)} = \Pr(\tilde{y}_i \neq y_i^* \mid i \in \mathcal{R}^{(s)}, \mathcal{H}_r)$ for the conditional error on recovered samples, the relabeling error rate satisfies

$$\xi_{\mathcal{R}}^{(s)} \;=\; \sum_{r\in\{0,1\}} \frac{\pi_r \cdot \Pr(t_i^{(3)} > \tau^{(3)} \mid \mathcal{H}_r)}{\sum_{r'\in\{0,1\}} \pi_{r'} \cdot \Pr(t_i^{(3)} > \tau^{(3)} \mid \mathcal{H}_{r'})}\, \varepsilon_r^{(s)}. \tag{40}$$

Finally, $\xi_{\mathcal{R}}^{(s)}$ is controlled by the worst conditional error over the two populations, $\xi_{\mathcal{R}}^{(s)} \leq \max_{r\in\{0,1\}} \varepsilon_r^{(s)}$, so it inherits the expert's error on the purified data rather than being cancelled by the class mixture. Two regimes follow. If the expert becomes consistent on the purified data, i.e. its irreducible error $\mathcal{L}^*$ vanishes, then both conditional errors vanish and $\xi_{\mathcal{R}}^{(s)} \to 0$. If instead $\mathcal{L}^* > 0$, the conditional errors converge to the residual expert error and $\xi_{\mathcal{R}}^{(s)}$ converges to a strictly positive floor, whose size is governed by the expert's accuracy on the purified data. We therefore claim a vanishing relabeling error rate only in the realizable limit $\mathcal{L}^* \to 0$, and make no vanishing claim otherwise.

For every $i \in \mathcal{R}^{(s)}$ the final label is the expert decision, $\tilde{y}_i = \hat{y}_i$, because recovery consists precisely in re-annotating a suspicious sample with the expert's prediction. Hence $\tilde{y}_i \neq y_i^*$ holds if and only if the expert misclassifies $\mathbf{x}_i$, irrespective of whether $\mathbf{x}_i$ originates from $\mathcal{H}_0$ or $\mathcal{H}_1$. Expanding over the two populations,

$$\xi_{\mathcal{R}}^{(s)} = \sum_{r\in\{0,1\}} \Pr(\mathcal{H}_r \mid i \in \mathcal{R}^{(s)}) \cdot \varepsilon_r^{(s)}, \tag{41}$$

and Bayes' theorem gives the stated weights,

$$\Pr(\mathcal{H}_r \mid t_i^{(3)} > \tau^{(3)}) = \frac{\pi_r \cdot \Pr(t_i^{(3)} > \tau^{(3)} \mid \mathcal{H}_r)}{\sum_{r'\in\{0,1\}} \pi_{r'} \cdot \Pr(t_i^{(3)} > \tau^{(3)} \mid \mathcal{H}_{r'})}. \tag{42}$$

This formulation is essential: a noisy sample whose label is correctly restored contributes $\tilde{y}_i = y_i^*$ and therefore does not count as an error, so the noisy composition of $\mathcal{R}^{(s)}$ alone does not degrade supervision.

As s increases, by Assumption B.10 the expert error $\mathcal{L}^{(s)}$ decreases monotonically towards its irreducible value $\mathcal{L}^*$ on the purified dataset, so $\mathcal{L}^{(s)} - \mathcal{L}^*$ is a non-negative, non-increasing quantity that measures the remaining reducible error. Since $\tilde{y}_i = \hat{y}_i$ for every recovered sample, $\varepsilon_r^{(s)}$ is exactly the expert's conditional error on $\mathcal{R}^{(s)} \cap \mathcal{H}_r$, hence $\varepsilon_r^{(s)} \leq \mathcal{L}^{(s)} / \Pr(\mathcal{H}_r)$ up to the selection induced by the threshold. Both $\cos(\mathbf{g}_i, \mathbf{p}'_{\hat{y}_i})$ and $P(\hat{y}_i \mid \mathbf{x}_i)$ move towards their oracle values $\cos(\mathbf{g}_i, \mathbf{p}'_{y_i^*})$ and 1 at a rate governed by $\mathcal{L}^{(s)} - \mathcal{L}^*$, because they are both monotone functions of the expert's separation of the purified classes. In the realizable limit $\mathcal{L}^* \to 0$ the expert predicts y_i^* for every recovered sample, so $t_i^{(3,s)}$ attains its maximum value $\alpha + \beta$, $\varepsilon_0^{(s)} \to 0$, and a noisy sample that is recovered is relabeled with $\tilde{y}_i = \hat{y}_i = y_i^*$, i.e. it is *denoised*, so $\varepsilon_1^{(s)} \to 0$ as well; hence $\xi_{\mathcal{R}}^{(s)} \to 0$. When $\mathcal{L}^* > 0$, however, the conditional errors instead converge to the residual expert error, and $\xi_{\mathcal{R}}^{(s)}$ converges to a strictly positive floor that we do not claim to vanish.

Theorem B.14 (Convergence of Trust Evolution) *Under Assumptions B.10–B.11, the trust evolution process terminates in a finite number of rounds $S^* < \infty$; at termination no remaining candidate exceeds the current threshold, so $\mathcal{C}^{(S^*)}$ is a fixed point of the recovery rule. Every clean sample of $\mathcal{P}$ that the rule ever admits is admitted by round S^*, and no admitted sample is later reassigned.*

Since $|\mathcal{P}|$ is finite and $\mathcal{C}^{(s)}$ is non-decreasing (Theorem B.12), the sequence $|\mathcal{C}^{(s)}|$ is a non-decreasing integer-valued sequence bounded above by N. Therefore, it must eventually stabilize at some round S^*, i.e., $|\mathcal{C}^{(S^*+1)}| = |\mathcal{C}^{(S^*)}|$, implying $\mathcal{R}^{(S^*)} = \emptyset$.

The termination condition in PTC checks whether the trust scores are almost identical between consecutive rounds, i.e., $\|\mathbf{t}^{(3,s+1)} - \mathbf{t}^{(3,s)}\|_\infty \leq \epsilon_{\text{stop}}$. Since the trust scores are bounded in $[0, 1]$ and their updates are driven by improvements in expert accuracy, which by Assumption B.10 are bounded below by γ/N per clean recovery, the process reaches a fixed point in at most $O(N/\gamma)$ rounds.

At convergence, no additional clean samples can be recovered because either: (i) all remaining samples in $\mathcal{P}$ are genuinely noisy and their trust scores fall below $\tau^{(3)}$, or (ii) the expert has reached its optimal discriminative capability on the purified dataset and can no longer provide additional evidence to elevate the trust of remaining samples. This establishes finite termination and the stationarity of the trusted set at S^*.

[What is and is not claimed at convergence] Termination is a property of the finite candidate pool: because $\mathcal{C}^{(s)}$ grows inside a finite set and each round is driven by the expert, whose accuracy improves monotonically (Assumption B.10), only finitely many rounds can change any candidate's status. We deliberately do not claim that $\mathcal{C}^{(S^*)}$ recovers *every* clean sample of $\mathcal{P}$. Samples whose trust scores remain below $\tau^{(3)}$ stay in the pool, and $\tau^{(3)}$ is set by a distributional criterion rather than by an oracle that knows the true labels; clean samples that are indistinguishable from the corrupted population at the current expert quality are therefore not recovered, which is the conservative behaviour we want. The coverage statement we make is the one the algorithm can verify—stationarity at S^* and the absence of reassignment—while the fraction of clean samples actually recovered is reported empirically in Appendix D.

[Positive Feedback Loop] Theorems B.12–B.14 jointly characterize the positive feedback mechanism in Stage III. Each recovery round expands $\mathcal{C}^{(s)}$ with predominantly clean samples (Theorem B.13), which improves expert accuracy (Assumption B.10), which in turn increases the trust scores of remaining clean samples in $\mathcal{P}$ (Assumption B.11), enabling further recovery in the next round. This self-reinforcing cycle terminates in finite rounds (Theorem B.14), and the relabeling error rate decreases towards the expert's residual error on the purified data, vanishing in the realizable limit. This provides the foundation for the progressive trust evolution principle underlying PTC.

C Discussion on Related Works

C.1 Why poisoning-specific continual-learning defences are not directly comparable

To our knowledge, no prior work targets non-stationary label poisoning with a frozen pretrained backbone. Our setting is fixed by three constraints: label corruption is injected online, so the noise distribution changes from task to task; the learner is class-incremental and never revisits a previous task; and the backbone is a frozen pretrained ViT-B/16, so the feature space is stationary and no raw-image replay buffer is available. Two notions of stationarity should be kept apart. Because the encoder is frozen, the *evidence* from which trust is estimated does not drift with the task index; the corruption injected into each task, however, changes from task to task, and the set of classes seen so far grows, so a trust rule fitted once and reused unchanged across the stream remains ill-posed. It is this one-shot sense of *static* removal that we criticise, whereas our thresholds and per-class caps are re-derived for every task from the same fixed evidence source. Existing poisoning-specific CL defences are developed under a different combination of these axes—in particular Li et al. (2026), which recasts noisy-label learning as a continual-learning process, decomposing the network parameters into "clean" and "noisy" components whose update directions are regulated by terms with time-varying coefficients. The sequence it continualises is the iteration sequence of a single, stationary noisy training run, and the signal it exploits is the parameter trajectory of a fully trainable network; in our regime it is the non-stationarity of the stream, rather than the iteration index, that must be handled, and the representation is frozen. Porting such a method here would require ablating precisely the machinery on which it rests, so we do not report it as a numeric baseline. Instead, we span the design space by grouping the competitors according to the role they play: standard CL baselines (replay-based DER/DER++, prompt-based L2P/DualPrompt), PEFT competitors sharing our frozen-backbone regime (CL-LoRA, KeepLoRA, SR^2-LoRA), and generic noise-robust training rules obtained by integrating a standard noise-handling mechanism—of which Co-teaching Han et al. (2018b) is the representative instance—into the strongest PEFT backbone (KeepLoRA). The last group is the most direct probe of whether a generic noise-robust rule, without any trust calibration, suffices. Such rules differ from the one above in that they are portable: they replace the training objective or the sample-selection step rather than the architecture, so they can be embedded in our pipeline unchanged, sharing the same frozen ViT-B/16 backbone and the same LoRA and classifier-pool machinery, and differing from PTC only in how trust is estimated. Table 1 (Co-teaching) and Table 4 (SCE and APL) show that these rules remain clearly below PTC even with an identical backbone; the residual gap therefore isolates the contribution of progressive trust estimation from that of the backbone itself.

C.2 Experimental Setup

Datasets and task splits. We conduct experiments on three class-incremental benchmarks. Split CIFAR-10 and Split CIFAR-100 consist of $50{,}000$ training and $10{,}000$ test images of size 32×32, with 10 and 100 classes respectively. CUB-200 is a fine-grained bird benchmark containing $11{,}788$ images over 200 subcategories. We follow the standard class-incremental protocol and partition the classes into non-overlapping tasks: 5 tasks of 2 classes for Split CIFAR-10, 10 tasks of 10 classes for Split CIFAR-100, and 20 tasks of 10 classes for CUB-200. Training data of a task are observed only once and never revisited.

Corrupted continual learning (CLDP). Following the CLDP setting, we inject label corruption into the training stream of every task at a fixed ratio of 40%. Two attacks are considered. Under *Attack I* (pairwise/asymmetric noise), the labels of two predefined, visually related classes are swapped at the same ratio, producing a structured corruption whose flipped samples are mutually nearest neighbours and therefore locally self-consistent. Under *Attack II* (symmetric noise), each label is independently flipped with probability 0.4 to a class sampled uniformly from the remaining classes of the same task, so the corruption is diffuse and generally violates neighbourhood consensus. Because each Split CIFAR-10 task contains only two classes, sampling a different class within the task is equivalent to a pairwise swap, and the two attacks coincide; we therefore report a single result, denoted "I,II", for this benchmark. Test data remain clean in all settings.

Algorithm 1: The learning process of the proposed framework

Input: Task stream $\mathcal{T} = \{t_1, \ldots, t_n\}$ with datasets $\{\mathcal{D}_i\}_{i=1}^{n}$ (K classes per task), frozen ViT F, epochs E, kNN size k, trust-fusion weight λ, expert-fusion weights α, β, temperature T, per-class cap R_1, LoRA slots $\{C_m\}$, rectification schedule ($\mathrm{rect_interval}$, $\mathrm{rect_max_rounds}$, $k_{\min}$)

Output: Trained experts $\{G_\phi\}$, task-specific LoRA $\{C_m\}$, classifier pool W, trusted set $\mathcal{C}$

for $i = 1$ to n **do**
  // Stage I: Relational Trust Initialization on $\mathcal{D}_i$
  Extract ℓ_2-normalized frozen features $\mathbf{z}_j = F(\mathbf{x}_j)/\|F(\mathbf{x}_j)\|_2, \forall\, \mathbf{x}_j \in \mathcal{D}_i$
  Build feature kNN graph; per-sample trust $t_j^{(1)} = \lambda a_j + (1-\lambda) s_j$ (agreement a_j, compactness s_j) (Eqs. (1)–(4))
  Otsu threshold $\tau^{(1)}$ on $\{t_j^{(1)}\}$; clip the suspicious side by the per-class cap $\lfloor R_1 n_c \rfloor$ (Eq. (5))
  $\mathcal{C}^{(1)} = \{\mathbf{x}_j \mid t_j^{(1)} \geq \tau^{(1)}\},\ \mathcal{S}^{(1)} = \mathcal{D}_i \setminus \mathcal{C}^{(1)}$
  // Stage II: Prototype-guided Trust Refinement
  Prototypes $\mathbf{p}_c = \mathrm{norm}\big(\frac{1}{n_c} \sum_{j \in \mathcal{C}^{(1)}, y_j = c} \mathbf{z}_j\big)$; similarity $l_{j,c} = (\mathbf{z}_j^\top \mathbf{p}_c)/T$; normalized score $q_{j,c}$ (Eqs. (6)–(7))
  $t_j^{(2)} = q_{j,y_j}$; Otsu threshold $\tau^{(2)}$ on $\{t_j^{(2)}\}_{j \in \mathcal{C}^{(1)}} \Rightarrow \mathcal{S}^{(2)}, \mathcal{C}^{(2)}$
  $\mathcal{P} = \mathcal{S}^{(1)} \cup \mathcal{S}^{(2)};\ \mathcal{C} = \mathcal{C}^{(2)}$
  // Stage III: Expert-driven Trust Evolution (backbone frozen)
  Optimize G_ϕ on $\mathcal{C}$ with $\mathcal{L} = \mathcal{L}_{\mathrm{CE}} + \lambda_{\mathrm{rel}} \sum_\ell \mathrm{MSE}(g_\ell^{\mathrm{new}}, g_\ell^{\mathrm{old}})$; update LoRA slot C_i and classifier pool W
  for round s from $\lfloor E/2 \rfloor$, step $\mathrm{rect_interval}$, while $s \leq \mathrm{rect_max_rounds}$ **do**
    Recompute prototypes $\{\mathbf{p}'_c\}$ from $\mathcal{C}$; predict $\hat{y}_j = \arg\max_c G_\phi(\mathbf{x}_j)_c$ and confidence $P(\hat{y}_j \mid \mathbf{x}_j)$, $\forall\, \mathbf{x}_j \in \mathcal{P}$
    $t_j^{(3)} = \alpha \cos(\mathbf{g}_j, \mathbf{p}'_{\hat{y}_j}) + \beta\, P(\hat{y}_j \mid \mathbf{x}_j)$; Otsu $\Rightarrow$ recovered set $\mathcal{R}^{(s)}$ (Eq. (8))
    if $|\mathcal{R}^{(s)}| < k_{\min}$ **or** $\mathcal{P} = \emptyset$ **then**
      break // converged
    end if
    Relabel $\tilde{y}_j \leftarrow \hat{y}_j, \forall j \in \mathcal{R}^{(s)};\ \mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{R}^{(s)};\ \mathcal{P} \leftarrow \mathcal{P} \setminus \mathcal{R}^{(s)}$
  end for
  Align classifiers: resample per-class features from $\mathcal{N}(\mu_c, \Sigma_c)$; fine-tune W with CE+entropy (Eq. (9))
  Archive G_ϕ; append current LoRA weights to the block-diagonal classifier pool W
end for

Baselines. We compare against two categories of methods. *Standard CL Baselines* comprise representative strategies for mitigating catastrophic forgetting, including prompt-based methods (L2P Wang et al. (2022b), DualPrompt Wang et al. (2022a)) and rehearsal-based methods (DER Yan et al. (2021), DER++ Buzzega et al. (2020)). *SOTA Competitors* comprise parameter-efficient and noise-robust learning methods, including CL-LoRA He et al. (2025), KeepLoRA Luo et al. (2026), and SR^2-LoRA Wan et al. (2026). We additionally integrate Co-teaching Han et al. (2018b) into KeepLoRA, denoted KeepLoRA+CT, to obtain a robust baseline specifically designed for label poisoning. All methods adopt the same frozen ViT-B/16 backbone for fair comparison.

Metrics. We report two metrics. *Final* accuracy is the Top-1 accuracy on the test sets of all tasks seen so far, measured after the last task has been learned. *Avg* (average) accuracy is the mean of the Top-1 accuracies measured immediately after each task. The main results (Table 1) and the corruption-ratio study (Table 4) are averaged over three random seeds and reported as mean$\pm$std; the Stage III ablation (Table 2) lists the individual seed values together with their mean; the efficiency measurements (Table 5) are single runs on one NVIDIA RTX 3090.

C.3 Algorithm Pseudocode of the Proposed Framework

Algorithm 1 summarizes the complete pipeline. Each task passes through the three stages of PTC in order: Stage I initializes sample trust from the intrinsic geometry of the frozen representation, Stage II refines it against class prototypes, and Stage III iteratively recovers suspicious samples using the evolving expert. The backbone remains frozen throughout; only current-task data is accessible at each stage, and the trusted set accumulates monotonically across rectification rounds.

Table 3: Hyper-parameters of PTC. Symbols follow Algorithm 1; "—" marks settings without a dedicated symbol.

Symbol	Meaning	Value
E	training epochs per task	10
k	feature kNN size (Stage I)	20
λ	trust-fusion weight (Stage I)	0.85
R_1	per-class suspicious-sample cap (Stage I)	0.5
T	prototype temperature, softmax matching (Stage II)	0.1
—	prototype centroid / split	class mean / global
α, β	expert-fusion weights (Stage III)	1.0, 1.0
—	rectification interval / max. rounds	1 / 5
—	rectification ratio	0.5
—	rectification policy	aggressive; auto top-k (Otsu)
$k_{\min}$	min. samples to continue rectification	$\lfloor 0.1\,\lvert\mathcal{P}'\rvert \rfloor$
r	LoRA rank	8
—	LoRA modules	q, v of all attention blocks
—	LoRA initialization	zeros A, orthonormal B
—	expert/LoRA optimizer	Adam, $(\beta_1, \beta_2) = (0.9, 0.999)$, wd 0
—	classifier-head optimizer	SGD, lr 0.005, mom. 0.9, wd 5×10^{-4}
—	learning rate	10^{-5}, cosine decay per task
—	batch size	32
—	random seeds	$1, 3, 1993$

C.4 Implementation Details Omitted from the Main Text

For completeness, this subsection documents the components that appear in Algorithm 1 but are only summarized in the main text; all settings are shared across datasets.

Hyper-parameters. Table 3 collects the numerical settings used in all experiments; the symbols are those of Algorithm 1.

Stages I–II. Stage I caps the number of samples marked suspicious at $\lfloor R_1 n_c \rfloor$ per class c, where n_c is the number of samples carrying the observed label c in the current task, with R_1 fixed to the same small value for every dataset and task; the cap prevents a class from being emptied when the adaptive threshold falls inside a concentrated trust mode. Prototypes are built only from $\mathcal{C}^{(1)}$ and re-normalized after averaging, with the temperature T fixed across benchmarks; classes whose trusted set is too small to yield a stable prototype are deferred to Stage III.

Expert and classifier. The expert G_ϕ is trained on $\mathcal{C}^{(2)}$ with a cross-entropy term plus a distillation term that penalizes the squared distance between the new and the stored feature responses, weighted by λ_{rel}; only the LoRA slot of the current task is optimized, while the backbone stays frozen. Rather than a single classifier we maintain a block-diagonal pool W that concatenates the per-task classifiers, appending the weights of a finished task on archiving. To reduce the mismatch between the task-local training distribution and the cumulative test distribution over all seen classes, the alignment step resamples per-class features from $\mathcal{N}(\mu_c, \Sigma_c)$ estimated on the trusted set and fine-tunes W with a cross-entropy plus entropy objective.

Rectification schedule. Stage III is triggered from epoch $\lfloor E/2 \rfloor$ and re-run every $\mathrm{rect_interval}$ epochs, for at most $\mathrm{rect_max_rounds}$ rounds, stopping early once fewer than $k_{\min}$ samples pass the trust threshold or the candidate pool becomes empty. These settings are shared across datasets; the stopping condition is the practical counterpart of the fixed-point criterion analysed in Theorem B.14.

D More Details on Experimental Results.

D.1 Attack-dependent and reversed degradation.

Although PTC improves over the strongest competitor under both attacks (Table 1), the competing methods do not degrade in a common direction: their sensitivity to the two corruption types is not merely different but *reversed*. On Split CIFAR-100, CL-LoRA attains 80.50% under Attack I yet only 53.15% under Attack II, a drop of 27.35 points, whereas SR^2-LoRA behaves in the opposite way, reaching 82.03% under Attack II but 66.69% under Attack I; the same reversal recurs on CUB-200, where SR2-LoRA scores 62.16% under Attack I against 80.87% under Attack II. L2P leans in the same direction as CL-LoRA, dropping from 53.17% to 46.25% on Split CIFAR-100 and from 36.75% to 26.82% on CUB-200 when moving from Attack I to Attack II. A distinct pattern appears among rehearsal-based methods: DER and DER++ retain a comparatively high average accuracy yet suffer a collapse in final accuracy (9.84% and 11.54% on Split CIFAR-100), the signature of catastrophic forgetting compounded by corrupted replay. Because these failures are heterogeneous rather than a shared loss of model capacity, we examine one representative case per attack direction below.

D.2 Two representative failure cases.

We attribute the reversed degradation to how each attack interacts with the specific defence mechanism a method relies on, and illustrate this with CL-LoRA and SR^2-LoRA. In the case of CL-LoRA, the method incorporates no explicit noise-detection mechanism and consumes incoming labels directly, relying on the low-rank constraint to regularize the update. Under the pairwise attack the distortion is systematic but confined: only two predefined, visually related classes are involved, the swapped categories lie in the immediate vicinity of one another, and the remaining categories retain clean supervision that anchors the shared representation, so a non-zero-mean shift that is compatible with a low-rank hypothesis can still be accommodated, which explains why the same method remains competitive in this regime. Under symmetric noise, by contrast, every class is struck with the same intensity and each label is scattered across all remaining classes of the task, so the distortion is diffuse and mutually contradictory across classes and no consistent low-rank update can absorb it. Because the corrupted samples lie far from the category they are assigned to, local geometry offers no coherent structure to consolidate, and the conflicting supervision is instead injected into every class simultaneously; once the shared representation is degraded in this way the damage is irreversible, since every subsequent task is adapted on top of it and earlier tasks are never revisited. In the mirrored direction, SR^2-LoRA fails because its correction signal is derived internally, from the consistency of inter-layer relations within the adapted model, and therefore requires the corruption to leave a detectable inconsistency. Under the pairwise attack the two swapped classes are strongly related and their samples are mutually nearest neighbours, so the relational structure on which the method relies endorses the erroneous assignment rather than contradicting it, and no internal signal is available to flag the distortion. Symmetric noise violates this precondition much less severely, since each label is scattered to an unrelated class and the resulting relational pattern is internally inconsistent, which the method can register as a potential error. This asymmetry confirms that internal corrective signals are only as trustworthy as the statistics they are estimated from: they are effective when the corruption leaves a trace of inconsistency, and remain silent when the corruption is self-consistent.

D.3 Why PTC defends against both.

The two failure cases above expose complementary blind spots: CL-LoRA is defeated by corruption that is diffuse and therefore globally contaminating, whereas SR^2-LoRA is defeated by corruption that is systematic and therefore locally self-consistent. PTC is designed to draw on evidence that is immune to both. Stage I (RTI) derives trust exclusively from the frozen pretrained representation space, where the encoder is never updated and the feature geometry is therefore unaffected by annotation corruption; its neighbourhood agreement term detects the diffuse, locally inconsistent flips of Attack II without consulting the classifier at all. Stage II (PGTR) then adds the global view, comparing each remaining sample against class prototypes that summarize the dominant semantic direction of a category rather than individual neighbours; this exposes pairwise-swapped samples that are locally self-consistent yet collectively point away from the category centre to which they

Table 4: Final accuracy (%) on Split CIFAR-10 under different label-corruption ratios. The 0% column corresponds to clean training and serves as the oracle upper bound of each method. "Drop" denotes the relative accuracy degradation with respect to the clean baseline, i.e., $\mathrm{Drop} = (A_{0\%} - A_p)/A_{0\%} \times 100\%$, so a smaller value indicates stronger robustness. Since every task of Split CIFAR-10 contains only two classes, the symmetric and pairwise attacks coincide, and a single column is reported for each corruption ratio. All results are averaged over three random seeds, reported as mean±std. The best accuracy and the smallest drop per column are in bold.

		20%Corruption		40% Corruption	
Method	0% (Clean)	Acc.	Drop↓	Acc.	Drop↓
DualPrompt	94.99±0.16	72.44±0.57	23.74	26.40±0.68	72.21
KeepLoRA	94.22±0.52	65.51±0.90	30.47	27.74±0.45	70.56
KeepLoRA+SCE	93.04±0.47	84.96±0.71	8.68	32.65±1.03	64.90
KeepLoRA+APL	91.48±0.20	87.46±0.36	4.39	68.66±0.18	24.94
CL-LoRA	97.15±0.28	86.69±0.91	10.77	80.90±0.41	16.73
PTC (Ours)	**97.52±0.19**	**97.41±0.34**	**0.11**	**97.32±0.21**	**0.21**

are assigned. Crucially, the two stages act as a conjunction rather than as alternatives—Stage II only re-examines samples that already passed Stage I, $\mathcal{C}^{(2)} = \mathcal{C}^{(1)} \setminus \mathcal{S}^{(2)}$, so a global criterion cannot overturn a locally supported decision, while the candidate pool $\mathcal{P} = \mathcal{S}^{(1)} \cup \mathcal{S}^{(2)}$ accumulates precisely those samples that neither local relations nor global prototypes can endorse. Since the two criteria are computed in disjoint ways—one from neighbourhood structure, the other from class-level aggregation—a corruption pattern that evades one is exposed by the other, which is the mechanism behind the consistent robustness observed in Table 1.

D.4 Sensitivity to the Corruption Ratio

Table 4 repeats Split CIFAR-10 at 0%, 20% and 40% corruption. PTC attains the best clean accuracy (97.52%), statistically comparable to CL-LoRA (97.15%) and clearly above DualPrompt (94.99%) and KeepLoRA (94.22%), so its gains under corruption are not bought by sacrificing in-distribution accuracy; it then loses only 0.20 points from 0% to 40%, within its own seed variation, whereas CL-LoRA drops 16.25 points and DualPrompt and KeepLoRA lose 68.59 and 66.48 points, ending far above both the 10% chance level and the 20% that a learner retaining only its most recent task would obtain. Normalised by corruption level, the decay is 0.10 points per additional 20% for PTC against 8.13 for CL-LoRA, and the intermediate column separates two degradation profiles: PTC is flat in both halves (0.11 and 0.09 points), CL-LoRA loses most of its total drop early (10.46 of 16.25 points at 20%), while DualPrompt and KeepLoRA lose more in the second half (46.04 and 37.77 points) than in the first (22.55 and 28.71), i.e. their degradation accelerates as corruption accumulates. This insensitivity follows from the design: the calibration of Eq. (7) estimates the partition from the trust distribution rather than assuming a rate, so a change in ε moves the threshold rather than the criterion.

D.5 Why the Defence is Almost Complete on Split CIFAR-10

The margin over the strongest competitor is markedly larger on Split CIFAR-10 ($+16.42$ under Attack I,II) than on the finer-grained benchmarks ($+6.66$ on Split CIFAR-100 and $+16.04$ on CUB-200 under Attack I; $+5.67$ and $+4.84$ under Attack II), because Split CIFAR-10 is the regime most favourable to purification-style defences. With only ten coarse classes and a frozen ViT-B/16 encoder the class-conditional distributions are far apart—the clean accuracy of every method in Table 4 exceeds 94%, and PTC reaches 97.52% without corruption—so the semantic neighbourhood of a sample is dominated by its true class, Assumption B.2 holds with a large margin, and the pairwise attack degenerates because each task contains exactly two classes, making Attacks I and II coincide. The per-class budget is equally generous: each task contributes $5{,}000$ images over two classes, i.e. about $2{,}500$ per class, so the prototype and expert statistics stay well estimated even after the conservative Stage I filter and the residual loss is governed by the rate of mistakenly removed clean samples rather than by the corruption ratio—consistent with the flat curve of Table 4, where PTC loses only

Table 5: Efficiency comparison on Split CIFAR-10 under the 40% poisoning ratio. Training time denotes the total time over all 5 tasks, measured on a single NVIDIA RTX 3090. Inference FLOPs are measured per forward pass on a single 224×224 image.

Method	Trainable Params	FLOPs	Peak GPU Memory	Training Time
L2P	54 K	18.1 G	4070 MiB	184 min
DualPrompt	590K	18.2 G	4309 MiB	200 min
KeepLoRA	983K	18.0 G	15444 MiB	131 min
CL-LoRA	221K	17.7 G	14762 MiB	278 min
Ours	221K	17.7 G	5378 MiB	68 min

0.20 points from 0% to 40%. We therefore regard the fine-grained benchmarks as the realistic regime and treat the near-complete defence on Split CIFAR-10 as the best case of the framework.

D.6 Efficiency and Parameter Analysis

Table 5 reports the efficiency of PTC on Split CIFAR-10 under the 40% poisoning ratio. PTC introduces only 221K trainable parameters, matching CL-LoRA and remaining $4.4\times$ and $2.7\times$ below KeepLoRA (983K) and DualPrompt (590K), since the three trust stages add no trainable module and operate on cached ℓ_2-normalized features and class prototypes. Because the backbone stays frozen, the inference cost equals that of the other adapter-based methods (17.7G FLOPs per 224×224 forward pass), whereas peak GPU memory falls to 5378 MiB, about 35% of CL-LoRA and KeepLoRA. Training requires 68 min over the five tasks, the shortest among all compared methods and $4.1\times$ faster than CL-LoRA, because the expert is optimized on the purified subset $\mathcal{C}^{(2)}$ rather than the full corrupted stream and the EDTE stopping rule suspends re-optimization once the trust scores stabilize.

E Ablation Study

E.1 Energy Metric.

For a sample i, let $\mathbf{g}_i^{\text{used}}$ denote the gradient computed with the label actually used for that sample (the corrupted annotation, or its corrected counterpart once the sample is reclaimed in Stage III), and let $\mathbf{g}_i^{\text{true}}$ denote the gradient computed with its ground-truth label. In gradient-based training the per-sample gradient is precisely the direction in which a sample's supervision moves the trainable parameters, and the update applied to a mini-batch is the sum of these per-sample directions. The energy therefore has a direct operational reading: it is the per-sample orientation of the supervision carried by the label in use, measured against the orientation that the true category would induce. We define it as

$$e_i = 1 - \max\big(\cos\big(\mathbf{g}_i^{\text{used}}, \mathbf{g}_i^{\text{true}}\big),\, 0\big)^2, \qquad \cos\big(\mathbf{g}_i^{\text{used}}, \mathbf{g}_i^{\text{true}}\big) = \frac{\mathbf{g}_i^{\text{used}} \cdot \mathbf{g}_i^{\text{true}}}{\big\|\mathbf{g}_i^{\text{used}}\big\|\, \big\|\mathbf{g}_i^{\text{true}}\big\|}. \tag{43}$$

The energy is 0 when the two gradients align, and increases as the angle between them grows, reaching its maximum of 1 once they are orthogonal or opposed. The clamp on the cosine is required rather than cosmetic: without it, $1 - \cos^2 = \sin^2$ assigns zero energy—perfect reliability—to a fully opposed gradient, which is the most harmful case. The clamp never overstates a sample's alignment, and on the non-aligned side it saturates at 1, so all samples whose update conflicts with the true objective are treated as equally unreliable; this is a deliberately conservative choice rather than a discriminating one. Lower energy indicates more reliable supervision.

Because corruption is confined to the annotation channel, the image content of a poisoned sample and its frozen features are untouched: the sample carries evidence that is visually intact and semantically meaningful, but is bound to an update direction that conflicts with its true category. Stage III changes exactly this binding and nothing else. It introduces neither additional data nor additional optimization rounds: the recovered samples already belong to the candidate pool $\mathcal{P}$ and are merely relabelled and merged into $\mathcal{C}$ (Algorithm 1), and the rectification rounds are executed within the same per-task epoch budget E. Enabling or disabling Stage III therefore changes only the labels

attached to $\mathcal{R}$, so any accuracy difference between the two configurations (Table 2) must originate from the orientation of the supervision those samples carry—precisely the quantity measured by the energy. Fig. 3(c) shows that this quantity improves on exactly the samples Stage III returns, which ties the reclaimed supervision to the observed accuracy gain. We do not claim a quantitative law mapping energy to accuracy; the energy serves as a per-sample diagnostic of supervision quality, not as a predictor of the final score.