Uncertainty-Aware Probabilistic Constrained Clustering from Entangled Pairwise Supervision

Shaojie Zhang **Ke Chen**
Department of Computer Science, The University of Manchester, Manchester M13 9PL, U.K.
{shaojie.zhang,ke.chen}@manchester.ac.uk

Abstract

Pairwise constrained clustering typically relies on hard must-link/cannot-link labels, whereas realistic pairwise supervision may be real-valued and entangle intrinsic ambiguity, expert judgment, and stochastic corruption. Existing deep constrained clustering (DCC) methods mainly target hard, expert-agnostic constraints, treating soft labels mostly numerically rather than semantically. We formalize this setting as uncertainty-aware probabilistic constrained clustering (UPCC), defining a canonical aleatoric relation through a heterogeneous observation process and analyzing its conditional identifiability. We introduce ProbPair, an angular pairwise objective for probabilistic relations, and build ECI-PP, an estimator–corrector–integrator framework that refines imperfect supervision via belief estimation, correction, and reliability-aware integration. Across challenging probabilistic supervision settings, experiments on diverse benchmarks show that ECI-PP outperforms state-of-the-art DCC methods and remains robust with a shared default configuration. Code is available at our repository.

1 Introduction

Clustering is a fundamental tool for uncovering structure in data, yet its unsupervised nature often yields partitions that may not align with domain knowledge [1, 2, 3]; pairwise constrained clustering (CC) addresses this by using weak instance-level pairwise supervision [4, 5]. Traditional CC methods [6, 7, 8, 9, 10, 11], however, often struggle with high-dimensional and complex data, motivating deep constrained clustering (DCC). Existing DCC methods can be broadly organized into two paradigms: end-to-end DCC [12, 13, 14, 15], which reformulates clustering as a pseudo-classification problem with anchors, and deep constraint embedding [16, 17], which instead learns clustering-friendly representations from pairwise supervision and, in more recent advances, further moves pairwise learning from Euclidean to angular space to better reconcile positive and negative relations [3].

Despite these advances, the supervision setting remains largely idealized. Existing work typically assumes hard binary pairwise relations, whereas supervision in practice often lies on a continuum. Moreover, such real-valued pairwise labels may simultaneously reflect intrinsic *aleatoric* ambiguity, expert-conditioned and pair-dependent *epistemic* judgment [18], and additional *stochastic* corruption from annotation or recording pipelines. For example, when comparing two clinical cases, a recorded observation may reflect ambiguity in the cases themselves, systematic clinician judgment bias, and documentation or data-entry errors. These considerations motivate what we term *uncertainty-aware probabilistic constrained clustering* (UPCC): an unexplored, holistic, and practically realistic setting in which probabilistic pairwise relations jointly reflect intrinsic ambiguity, expert-dependent judgment, and stochastic corruption, more closely matching real-world supervision than binary or merely real-valued constraints.

We specify an observation model that distinguishes the canonical aleatoric relation from expert-conditioned judgment and stochastic corruption, and analyze when this relation is identifiable on

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

observable pairs. To learn under UPCC, we first introduce *ProbPair*, a deep constraint embedding approach linking probabilistic pairwise relations to angular representation learning. Built on it, we further develop an *Estimator–Corrector–Integrator ProbPair* (ECI-PP) framework, in which Estimators provide model beliefs, Correctors perform structured correction and suppress residual discordance, and the Integrator learns from the resulting surrogate supervision toward the underlying aleatoric relation.

Our main contributions are: (i) We formalize UPCC through a heterogeneous observation model and analyze identifiability of the canonical aleatoric relation on observable pairs. (ii) We introduce *ProbPair*, a deep constraint embedding approach linking probabilistic pairwise relations to angular embeddings. (iii) We develop *ECI-PP*, a practical framework for aleatoric relation learning under entangled supervision. (iv) We empirically show across diverse benchmarks that ECI-PP outperforms state-of-the-art DCC methods and is robust under fallible, heterogeneous, and corrupted supervision.

2 Related work

Beyond binary pairwise constraints. Existing CC methods either extend binary constraints only numerically, without semantically modeling intermediate supervision, or assign intermediate values to the confidence of a pre-defined relation type rather than a unified probability. Advanced deep methods typically employ logistic losses [19, 3] compatible with real-valued constraints in $[0, 1]$. However, *end-to-end DCC* methods [12, 14, 20, 21, 22, 15] remain tied to hard anchor semantics, with attendant anchor misalignment [23] and error propagation [24, 25], while *deep constraint embedding* methods [3, 17, 16] avoid anchors but still require a binary regime for their theoretical geometric guarantees. Likewise, the *deep generative* approach [2] encodes constraints as a signed prior over discrete assignments, precluding a unified treatment of real-valued relations. Earlier non-deep approaches considered intermediate supervision more explicitly: (i) *probabilistic* treatments [26, 27] encoded the activation probability of a positive same-cluster tie, but offered neither negative relations nor a unified notion of pairwise affinity; and (ii) *fuzzy-constraint* formulations attach an intermediate belief [28], degree [29], or penalty [30, 31, 32, 33] only after a discrete relation type has been specified. Taken together, these lines of work do not address uncertain relational judgment expressed by probabilistic labels under a unified semantics. By contrast, our *ProbPair* formulation, built on deep constraint embedding, directly models such unified probabilistic supervision, accommodating both vague annotations in practice and inherently ambiguous aleatoric relations.

Imperfect pairwise supervision. While imperfect annotations in CC have long been recognized [34, 35, 36, 37, 38, 39, 1, 2, 15], more realistic observation settings remain underexplored. Existing studies treat simplified noise through constraint-set robustness [34, 36], coarse prior uncertainty over assignments [2], or expert-level reliability in semi-crowdsourced settings [38, 1]. VolMaxDCC [15] further introduces confusion-based structured annotation modeling beyond unstructured label degradation or flipping. However, it still lacks heterogeneous multi-expert modeling and remains restricted to discrete binary-noise settings, leaving expert-specific, pair-dependent epistemic distortion unmodeled. By contrast, we consider genuinely entangled observations, arising from intrinsic *aleatoric* relations, expert-conditioned *epistemic* judgments, and *stochastic* corruption. Such a setting is reminiscent of *learning from crowds*, where expert behavior is commonly modeled under a fixed class-label scheme through expert-specific confusion or output layers [40, 41], or through expertise coupled with instance difficulty [42, 43]. Yet these strategies rely on a restricted class-label channel and are therefore not directly transferable to deep constraint embedding, where a highly flexible pairwise learner can readily absorb structured distortion and random corruption into its learned geometry. We therefore develop *ECI-PP*, a multi-role iterative framework built on *ProbPair* that uses model-side beliefs to correct structured distortion and filter residual inconsistency, yielding refined surrogate supervision that progressively guides learning toward the underlying aleatoric relation.

3 UPCC formulation and canonical relation

We study uncertainty-aware probabilistic constrained clustering (UPCC), where expert-indexed real-valued pairwise labels arise from a heterogeneous observation process. Let $\mathcal{X} = \{x_j\}_{j=1}^{|\mathcal{X}|}$ be a dataset to be partitioned into C clusters, and let $\mathcal{C} = \{(a_i, b_i, e_i, y_i)\}_{i=1}^{|\mathcal{C}|}$ denote the observed soft constraints, where (a_i, b_i) indexes the pair (x_{a_i}, x_{b_i}), $e_i \in \{1, \dots, E\}$ records the expert identity, and $y_i \in [0, 1]$

is the observed pairwise relation. The formulation targets a canonical pairwise probabilistic relation, while a final hard partition is treated as a downstream summary induced by that relation.

3.1 Observation model

We view each observed pairwise relation as arising from three factors: (i) the *canonical aleatoric relation* in the data (e.g., intrinsic clinical similarities between a common cold and the flu), (ii) an expert-conditioned *epistemic judgment* built on that relation (e.g., a novice doctor systematically overrating their similarity due to limited knowledge), and (iii) a subsequent *stochastic corruption* stage accounting for accidental recording noise (e.g., random clerical errors in medical records).

Aleatoric relation. Let $\mathcal{S}$ be the set of all C-partitions of $\mathcal{X}$, and let $S \in \mathcal{S}$ be a latent random partition. For any index pair (a, b), define the aleatoric co-membership relation

$$R^\star_{ab} := \Pr\big(x_a \text{ and } x_b \text{ belong to the same cluster under } S \mid \mathcal{X}\big) \in [0, 1].$$

This quantity defines the canonical aleatoric relation and remains well defined even when the latent partition is intrinsically ambiguous.

Epistemic judgment. Let $\sigma(\cdot)$ denote the sigmoid function, and define its clamped inverse $\mathrm{logit}_\varepsilon(u) := \mathrm{logit}\big(\min\{\max\{u, \varepsilon\}, 1-\varepsilon\}\big)$ with $\varepsilon \ll 1$. We also introduce a deterministic pair descriptor $\phi_{ab} = \phi(x_a, x_b)$, which serves as the covariate through which expert-dependent effects are parameterized. For expert e, let $m_e(\phi_{ab}) \in \mathbb{R}$ denote a structured mean effect, and let

$$u_{e,ab} \mid e, \phi_{ab} \sim \mathcal{N}\big(0, \tau_e^2(\phi_{ab})\big), \qquad \tau_e(\phi_{ab}) > 0 \text{ (conditional standard deviation)},$$

be a centered residual perturbation. The latent expert judgment is generated as

$$y^{\mathrm{jud}}_{e,ab} := \sigma\big(\mathrm{logit}_\varepsilon(R^\star_{ab}) + m_e(\phi_{ab}) + u_{e,ab}\big) \in (0, 1). \tag{1}$$

Under this construction, the judgment channel has the logistic-normal density

$$p_{\mathrm{jud}}\big(y \mid R^\star_{ab}, e, \phi_{ab}\big) = \frac{1}{y(1-y)}\, \varphi\Big(\mathrm{logit}(y);\, \mathrm{logit}_\varepsilon(R^\star_{ab}) + m_e(\phi_{ab}),\, \tau_e^2(\phi_{ab})\Big) \tag{2}$$

for $y \in (0, 1)$, where $\varphi(\cdot; \mu, s^2)$ denotes the Gaussian density; see Appendix A.1 for the derivation. The same judgment channel also gives rise to the corresponding mean distortion in probability space,

$$\bar{\epsilon}_{e,ab}(R^\star_{ab}, \phi_{ab}) := \mathbb{E}\Big[y^{\mathrm{jud}}_{e,ab} - R^\star_{ab} \mid R^\star_{ab}, e, \phi_{ab}\Big]. \tag{3}$$

Stochastic corruption. To capture unstructured recording noise beyond the latent judgment, we introduce a corruption indicator $c_{e,ab} \sim \mathrm{Bernoulli}(\pi_e)$, where $\pi_e \in [0, 1)$ is the corruption probability for expert e, and define the final observed relation by

$$y_{e,ab} \mid c_{e,ab} = 0 \sim p_{\mathrm{jud}}(\cdot \mid R^\star_{ab}, e, \phi_{ab}), \qquad y_{e,ab} \mid c_{e,ab} = 1 \sim \mathrm{Uniform}(0, 1).$$

Equivalently, letting $\mathbf{1}_{[0,1]}(\cdot)$ denote the uniform density on $[0, 1]$, the conditional observation law is

$$p\big(y \mid R^\star_{ab}, e, \phi_{ab}\big) = (1 - \pi_e)\, p_{\mathrm{jud}}\big(y \mid R^\star_{ab}, e, \phi_{ab}\big) + \pi_e\, \mathbf{1}_{[0,1]}(y). \tag{4}$$

3.2 Canonical relation and identifiability

We now formalize the canonical relation within a structural setup for identifiability. To obtain a finite-dimensional canonical estimand, we parameterize

$$R^\star_{ab} = r_\theta(x_a, x_b), \qquad m_e(\phi_{ab}) = m_{\eta_e}(\phi_{ab}), \qquad \tau_e(\phi_{ab}) = \tau_{\zeta_e}(\phi_{ab}),$$

and collect the parameters into $\Xi = (\theta, \{\eta_e\}_{e=1}^E, \{\zeta_e\}_{e=1}^E, \{\pi_e\}_{e=1}^E)$. This parameterization makes the canonical relation explicit under the specific observation family. Since ϕ_{ab} is introduced only as a deterministic pair descriptor without additional representational constraints, the identification of the canonical relation term $r_\theta(x_a, x_b)$ is governed by the structural conditions introduced below.

We first fix the trivial additive ambiguity in the location decomposition:

Assumption 3.1 (Centering on observable support)**.** *For each expert e with nonzero observation probability, $\mathbb{E}_{(a,b)|e}[m_{\eta_e}(\phi_{ab})] = 0$, where the expectation is taken over the observable-pair distribution conditional on expert e.*

This centering removes any remaining expert-specific constant shift on the observable support:

Lemma 3.2 (Gauge fixing)**.** *Suppose both $\{m_{\eta_e}\}_{e=1}^{E}$ and $\{m_{\eta'_e}\}_{e=1}^{E}$ satisfy Assumption 3.1. If for each expert e with nonzero observation probability there exists a constant $k_e \in \mathbb{R}$ such that $m_{\eta_e}(\phi_{ab}) = m_{\eta'_e}(\phi_{ab}) - k_e$ for all observable triples (a, b, e), then $k_e = 0$ for every such expert e.*

The proof is given in Appendix A.2. The logistic-normal clean channel in Eq. (2), combined with the observation law in Eq. (4), yields the following injectivity property:

Lemma 3.3 (Injectivity of the observation family)**.** *Let $\mu := \mathrm{logit}_\varepsilon(R^\star_{ab}) + m_e(\phi_{ab})$, $s := \tau_e(\phi_{ab})$, and $\pi := \pi_e$. For the observation law rewritten as*

$$p(y \mid \mu, s, \pi) := (1-\pi)\,\frac{1}{y(1-y)}\,\varphi\big(\mathrm{logit}(y); \mu, s^2\big) + \pi\,\mathbf{1}_{[0,1]}(y), \qquad y \in (0,1), \tag{5}$$

the mapping $(\mu, s, \pi) \mapsto p(\cdot \mid \mu, s, \pi)$ is injective on $\mathbb{R} \times (0, \infty) \times [0, 1)$.

The proof is given in Appendix A.3. The canonical formulation is completed by the following structural separability condition on the location term $\mathrm{logit}_\varepsilon\big(r_\theta(x_a, x_b)\big) + m_{\eta_e}(\phi_{ab})$:

Assumption 3.4 (Structural separability)**.** *Under Assumption 3.1, if for all observable triples (a, b, e),*

$$\mathrm{logit}_\varepsilon\big(r_\theta(x_a, x_b)\big) + m_{\eta_e}(\phi_{ab}) = \mathrm{logit}_\varepsilon\big(r_{\theta'}(x_a, x_b)\big) + m_{\eta'_e}(\phi_{ab}),$$

then for each expert e there exists a constant $k_e \in \mathbb{R}$ such that for all such observable triples,

$$\mathrm{logit}_\varepsilon\big(r_\theta(x_a, x_b)\big) = \mathrm{logit}_\varepsilon\big(r_{\theta'}(x_a, x_b)\big) + k_e, \qquad m_{\eta_e}(\phi_{ab}) = m_{\eta'_e}(\phi_{ab}) - k_e.$$

Assumption 3.4 provides the key structural requirement for the identifiability argument; boundary cases clarifying this condition are discussed in Appendix A.4. Writing the support of observable pairs as $\mathcal{P}_{\mathrm{obs}} := \{(a, b) : \exists e \text{ such that } (a, b, e) \text{ is observable}\}$, we then have the identifiability result:

Theorem 3.5 (Conditional canonical identifiability of the aleatoric relation)**.** *Let Ξ and Ξ' satisfy the observation model and Assumptions 3.1 and 3.4. Assume further that $r_\theta(x_a, x_b)$, $r_{\theta'}(x_a, x_b) \in [\varepsilon, 1-\varepsilon]$ for all $(a, b) \in \mathcal{P}_{\mathrm{obs}}$. If the induced conditional laws of the observed response coincide under Ξ and Ξ' for every observable triple (a, b, e), then $r_\theta(x_a, x_b) = r_{\theta'}(x_a, x_b)$ for all $(a, b) \in \mathcal{P}_{\mathrm{obs}}$. Hence the canonical aleatoric relation $R^\star_{ab} = r_\theta(x_a, x_b)$ is identifiable on observable pairs.*

The proof is in Appendix A.5. Under the specific observation family, Theorem 3.5 gives conditional identifiability of the canonical aleatoric relation on observable pairs at the law level. In the practical single-observation regime, however, estimation still relies on the shared parametric structure imposed across pairs and experts rather than repeated observations of the same pair–expert combination.

Implication for learning. The analysis above specifies the expert-independent relation that learning should target and clarifies when it is distinguishable, at the observation-law level, from structured expert effects and stochastic corruption. It does not imply that these latent components can be separately recovered from a single observation of each pair–expert triple. Instead, it motivates a practical learning strategy that exploits shared structure across pairs and experts: construct a model-side proxy for the canonical relation, use expert-conditioned regularities to correct predictable observational distortion, and treat the remaining unexplained discrepancy conservatively as a reliability signal. Section 4 operationalizes these principles through practical surrogate learning rather than explicit estimation of the full latent observation model.

4 Method

Section 3 provides both the learning target and the structural guidance for learning from entangled supervision: the canonical relation is the expert-independent quantity of interest, whereas expert-conditioned distortion and residual corruption should not be indiscriminately absorbed into the learned pairwise geometry. We translate this guidance into two stages. We first introduce *ProbPair*, a base representation learner that gives unified probabilistic pairwise relations a direct angular learning objective. We then develop *ECI-PP*, which uses model-side relation estimates, expert-conditioned correction, reliability-aware screening, and iterative integration to refine the observed supervision toward the canonical relation without explicitly fitting every latent component of the observation law.

4.1 ProbPair: probability-aware pairwise representation learning

We establish a representation learning formulation under probabilistic pairwise supervision. Given supervised pairs $\{(a_i, b_i, y_i)\}_{i=1}^{|\mathcal{C}|}$, where $y_i \in [0, 1]$ denotes the probabilistic relation for pair (x_{a_i}, x_{b_i}), we aim to learn clustering-oriented representations. To this end, we adopt a deep constraint embedding formulation with encoder $f_\psi : \mathcal{X} \to \mathcal{Z} \subset \mathbb{R}^D$ and latent code $z_j = f_\psi(x_j)$. We work in angular space, where bounded cosine similarity provides a natural geometric quantity for probabilistic calibration. To connect probabilistic pairwise relations with the latent representation space, we introduce the following probabilistic readout:

$$\hat{y}_{a_i b_i} := \sigma\big((\cos(z_{a_i}, z_{b_i}) - m)/T\big) \in (0, 1), \tag{6}$$

where $m \in \mathbb{R}$ and $T > 0$ are learnable parameters controlling the operating point and sharpness of the mapping. The resulting pairwise objective, referred to as the *ProbPair* loss, is given by

$$\mathcal{L}_{\mathrm{PP}} = -\frac{1}{|\mathcal{C}|} \sum_{i=1}^{|\mathcal{C}|} \Big[y_i \log \hat{y}_{a_i b_i} + (1 - y_i) \log(1 - \hat{y}_{a_i b_i}) \Big]. \tag{7}$$

Unlike hard-constraint objectives, Eq. (7) directly accommodates non-binary relation strengths. As a result, the induced geometry need not collapse to a purely binary regime, but can preserve both confident and uncertain pairwise structure.

We further include a reconstruction regularizer to avoid degenerate representations and preserve the global structure of the data. Since the representation is learned in angular space, reconstruction is performed from normalized latent embeddings, as in [3]. Specifically, with decoder $g_{\psi'} : \mathcal{Z} \to \mathcal{X}$ and $\mathrm{Norm}(\cdot)$ denoting ℓ_2 normalization, we decode from $\hat{x}_j = g_{\psi'}(\mathrm{Norm}(z_j))$ and use reconstruction loss $\mathcal{L}_{\mathrm{rec}} = \frac{1}{|\mathcal{X}|} \sum_{j=1}^{|\mathcal{X}|} \|x_j - \hat{x}_j\|_2^2$. With $\lambda_{\mathrm{rec}} > 0$ balancing pairwise relation fitting and reconstruction regularization, the overall objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{PP}} + \lambda_{\mathrm{rec}} \mathcal{L}_{\mathrm{rec}}. \tag{8}$$

Minimizing Eq. (8) yields optimal normalized embeddings $\mathcal{Z}_{\mathrm{norm}} = \{\mathrm{Norm}(z_j)\}_{j=1}^{|\mathcal{X}|}$ that are shaped by probabilistic pairwise relations while being regularized by reconstruction. These angular embeddings can then be used by a downstream clustering algorithm to obtain the final partition, and can also produce probabilistic relation estimates for instance pairs through Eq. (6).

4.2 Estimator–Corrector–Integrator ProbPair

Building on ProbPair, ECI-PP operationalizes the decomposition in Section 3 through practical proxies: the *Estimator* supplies an out-of-fold model-side proxy for the canonical relation; the *Corrector* captures predictable expert-conditioned mean discrepancy; and the *Integrator* treats the remaining inconsistency as a reliability signal when constructing refined supervision. The resulting supervision is then fed back to refresh the model-side reference, yielding a finite-stage iterative refinement procedure. These modules are motivated by the roles of the components in the observation model, but are deliberately practical surrogates rather than explicit estimators of every latent term in Eqs. (1) to (4). Figure 1 summarizes this correspondence and the resulting ECI-PP pipeline.

Estimator. To enable the subsequent learning of structured observational effects from entangled supervision, ECI-PP first constructs a model-side reference that is less coupled to the raw observations. In flexible pairwise representation learning, beliefs fitted on the same constraints used for training can absorb noisy observations in-sample and are therefore unreliable for this role. We accordingly partition $\mathcal{C}$ into K disjoint folds $\{\mathcal{C}^{(k)}\}_{k=1}^{K}$ and instantiate K non-shared Estimators $\{\mathrm{Est}^{(k)}\}_{k=1}^{K}$, where $\mathrm{Est}^{(k)} = (f_{\omega_k}, g_{\omega'_k}, m_k, T_k)$ denotes the encoder, decoder, and fold-specific readout parameters. Each $\mathrm{Est}^{(k)}$ is trained on $\mathcal{C} \setminus \mathcal{C}^{(k)}$ by minimizing $\mathcal{L}$ in Eq. (8), and then produces holdout beliefs on $\mathcal{C}^{(k)}$ through Eq. (6). Aggregating these predictions over all folds yields a full set of out-of-fold beliefs $\{\hat{y}_i^{\mathrm{oof}}\}_{i=1}^{|\mathcal{C}|}$, which serves as the model-side reference. To initialize the estimator, we weight each observed pair in ProbPair loss by an information-based decisiveness score, defined as

$$\kappa_i = \frac{D_{\mathrm{KL}}\big(\mathrm{Bern}(y_i) \,\|\, \mathrm{Bern}(\bar{y})\big)}{(1 - y_i)\, D_{\mathrm{KL}}\big(\mathrm{Bern}(0) \,\|\, \mathrm{Bern}(\bar{y})\big) + y_i\, D_{\mathrm{KL}}\big(\mathrm{Bern}(1) \,\|\, \mathrm{Bern}(\bar{y})\big)} \in [0, 1], \tag{9}$$

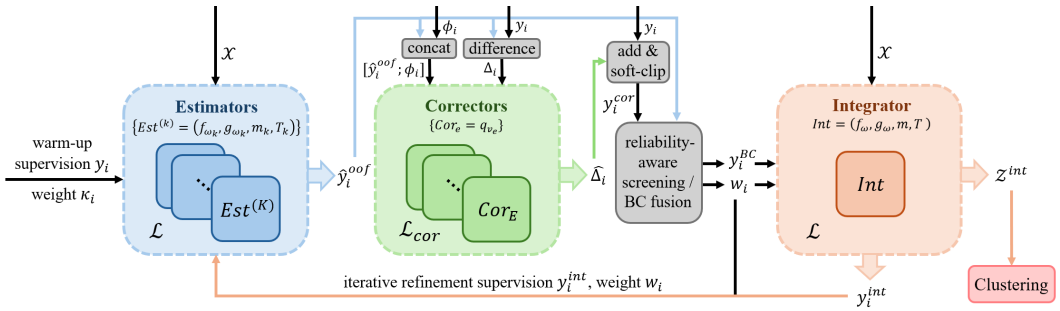


Figure 1: Overview of the ECI-PP pipeline. The **Estimators** are trained with the ProbPair objective $\mathcal{L}$ (Eq. (8)) and produce out-of-fold beliefs $\hat{y}_i^{\text{oof}}$ via the readout in Eq. (6). The **Correctors** take $[\hat{y}_i^{\text{oof}}; \phi_i]$ as input, with ϕ_i the pair-i descriptor, and learn probability-space corrections $\hat{\Delta}_i$ via Eq. (10), yielding corrected relations y_i^{cor}. **Reliability-aware screening** produces reliability weights w_i, and **BC fusion** forms the refined supervision y_i^{BC} via Eqs. (11) and (12). The **Integrator** learns embeddings from the refined supervision; its Integrator-side relations y_i^{int} are fed back to the **Estimators** for iterative refinement, while the final embeddings $\mathcal{Z}^{\text{int}}$ are **clustered** into the final partition.

where $\bar{y} = \frac{1}{|\mathcal{C}|}\sum_{i=1}^{|\mathcal{C}|} y_i$, $D_{\text{KL}}(\cdot\|\cdot)$ denotes the Kullback–Leibler divergence, and $\text{Bern}(\cdot)$ denotes the Bernoulli law. This weighting emphasizes more decisive judgments, which are typically expressed with greater expert confidence, and thereby stabilizes the initial out-of-fold beliefs.

Corrector. Following the theory-to-surrogate mapping above, the Corrector targets the *predictable mean effect* induced by the expert-conditioned judgment channel, rather than attempting to recover an individual residual realization or the full conditional distribution in Eqs. (1) to (3). For a constraint $i = (e_i, a_i, b_i)$, we define the empirical discrepancy $\Delta_i := \hat{y}_i^{\text{oof}} - y_i \in (-1, 1)$ and introduce expert-specific Correctors $\{\text{Cor}_e\}_{e=1}^{E}$, where each Cor_e is parameterized by a small network q_{ν_e} on $\mathcal{I}_e := \{i : e_i = e\}$ and conditioned on the out-of-fold belief and pair descriptor $[\hat{y}_i^{\text{oof}}; \phi_i]$. Here we instantiate $\phi_i = [x_{a_i} \odot x_{b_i};\ |x_{a_i} - x_{b_i}|]$, with $\odot$ denoting element-wise product, and write the Corrector output as $\hat{\Delta}_i := q_{\nu_{e_i}}([\hat{y}_i^{\text{oof}}; \phi_i]) \in (-1, 1)$. Although the structured expert effect is defined in logit space in Section 3.1, we perform correction in probability space for practical stability: small probability differences near 0 or 1 can correspond to disproportionately large logit residuals. Thus, $q_{\nu_{e_i}}$ approximates a probability-space correction for the predictable mean distortion induced by the expert-conditioned channel, rather than explicitly estimating its logit-space residual or heteroscedastic variance $\tau_{e_i}^2(\phi_i)$. Importantly, Δ_i need not contain only structured expert effects; it may also include stochastic corruption and Estimator proxy error. We therefore use a small regularized Corrector to favor recurring expert-conditioned structure over arbitrary pair-specific fluctuations, trained as

$$\mathcal{L}_{\text{cor}}^{(e)} = \frac{1}{|\mathcal{I}_e|}\sum_{i\in\mathcal{I}_e}\Big[\rho(\hat{\Delta}_i - \Delta_i) + \lambda_{\text{cor}}|\hat{\Delta}_i|^2\Big]. \tag{10}$$

Here, $\rho(\cdot)$ is the Huber loss and $\lambda_{\text{cor}} > 0$ weights the ℓ_2 regularizer, discouraging spurious corrections.

Integrator. The Integrator, parameterized as $\text{Int} = (f_\omega, g_{\omega'}, m, T)$, is the learner that integrates the refined supervision into a clustering-oriented embedding. To build this refined supervision from the Corrector output, we first form a corrected relation y_i^{cor}, quantify the remaining inconsistency by gap_i, and convert it into a screening weight w_i, with $\gamma > 0$ controlling the sharpness of down-weighting:

$$y_i^{\text{cor}} := \text{softclip}_\xi(y_i + \hat{\Delta}_i), \qquad \text{gap}_i := \left|\text{softclip}_\xi(\hat{y}_i^{\text{oof}}) - y_i^{\text{cor}}\right|, \qquad w_i := (1 - \text{gap}_i)^\gamma. \tag{11}$$

Here, $\text{softclip}_\xi(u) := \xi^{-1}\,\text{softplus}(\xi u) - \xi^{-1}\,\text{softplus}(\xi(u-1))$ with $\xi > 0$ softly maps values into $(0, 1)$ while preserving gradient flow near the boundaries; applying the same map to $\hat{y}_i^{\text{oof}}$ keeps the gap computation in the same bounded space. The residual gap_i quantifies the unexplained discrepancy that remains after structured correction. While the observation model associates this discrepancy with stochastic corruption and residual epistemic variation, gap_i may also reflect proxy error in $\hat{y}_i^{\text{oof}}$ and imperfect Corrector fitting in practice. We therefore treat it uniformly as a reliability signal for heuristic screening through w_i. Accordingly, w_i should not be interpreted as a posterior

corruption probability. We then set $n_i := n_0 w_i$ with $n_0 > 0$, and construct the Bayesian-confidence supervision

$$y_i^{\mathrm{BC}} := (y_i + n_i\, y_i^{\mathrm{cor}})/(1 + n_i), \tag{12}$$

which keeps y_i as a fixed prior while letting y_i^{cor} contribute according to its screened reliability. The Integrator then optimizes $\mathcal{L}$ (Eq. (8)) using y_i^{BC} as the pairwise target, with w_i multiplying each per-constraint term in $\mathcal{L}_{\mathrm{PP}}$ (Eq. (7)) before averaging, yielding Integrator embeddings.

Iterative refinement and deployment. ECI-PP operates by passing supervision through the Estimator–Corrector–Integrator pipeline to produce, via Eq. (6), Integrator-side relations y_i^{int}, and then feeding $(y_i^{\mathrm{int}}, w_i)$ back to the Estimators for iterative refinement. Warm-up initializes this loop by starting the Estimators from (y_i, κ_i); subsequent rounds replace this with $(y_i^{\mathrm{int}}, w_i)$. At each round, the refreshed Estimator beliefs are passed again through the Corrector–Integrator pipeline to update y_i^{cor}, gap_i, w_i, and y_i^{BC}, thereby yielding a practical finite-stage procedure that iteratively refines the surrogate supervision toward the canonical relation. After refinement, the final normalized Integrator embeddings $\mathcal{Z}^{\mathrm{int}} := \{\mathrm{Norm}(f_\omega(x_j))\}_{j=1}^{|\mathcal{X}|}$ encode the learned pairwise relation structure; applying an unsupervised clustering algorithm to $\mathcal{Z}^{\mathrm{int}}$ then yields the final partition. Appendix B provides algorithmic details and complexity analysis.

5 Experiments

5.1 Experimental settings

Our experiments evaluate ECI-PP under UPCC along four axes: (i) comparison with DCC and probabilistic pairwise baselines, (ii) robustness and complementary supervision tests, (iii) held-out diagnostics of internal dynamics, and (iv) ablation and sensitivity analysis.

Datasets. We adopt six image benchmarks (CIFAR100-20, CIFAR10 [44], FMNIST [45], ImageNet10 [46], MNIST [47], STL10 [48]) and two text benchmarks (Reuters subset [49], RCV1-10 [3]), covering varying class counts and class imbalance. We additionally use CIFAR-10H [50] for evaluation under human-derived probabilistic supervision. Dataset details are in Appendix C.1.

Compared methods. We compare our progressive models (ProbPair, Weighted ProbPair, and the full ECI-PP framework) against four external baselines: VanillaDCC [19] as a basic logistic pairwise baseline, VolMaxDCC [15] with explicit noise modeling, and CIDEC [14] and SpherePair [3] as state-of-the-art end-to-end DCC and deep constraint embedding methods, respectively. ProbPair uses only $\mathcal{L}$ in Eq. (8), while Weighted ProbPair additionally uses the information-based weight in Eq. (9).

Constraint generation. To instantiate UPCC, we generate constraints with trained expert classifiers rather than deriving binary labels from ground-truth classes. Experts are trained on controlled labeled subsets; uniformly sampled constraint pairs assigned to expert e are given a judgment $y_{e,ab}^{\mathrm{jud}} = \langle \boldsymbol{p}_a^e, \boldsymbol{p}_b^e \rangle \in [0, 1]$ from the predictive distributions, and the recorded y_i is then produced through the corruption channel in Section 3.1. For single-expert supervision, settings `lv0.1`/`lv0.01`/`lv0.001` vary the labeled-data fraction used to train the expert. Multi-expert settings `multi2`/`multi3`/`multi10` use multiple experts with complementary familiar/unfamiliar categories, where unfamiliar categories act as expert blind spots. All training constraints are generated from the corresponding training split only. See Appendix C.3 for the full procedure.

Protocol. For a comparative study, we report clustering metrics (ACC/NMI/ARI) over five trials; embedding-based methods use K-means on learned representations, while end-to-end DCC baselines use native outputs. The main comparison uses the default `lv0.01` single-expert and `multi3` multi-expert regimes, both with corruption probability 0.3 and 9k constraints in total (3k per expert under `multi3`). Robustness tests vary single-expert quality, corruption probability, and multi-expert configuration with matched budgets; complementary tests probe generator misspecification through alternative relation constructions and corruption mechanisms, as well as human-derived supervision. We also conduct held-out diagnostics, ablation, and sensitivity analysis.

Implementation. For fair comparison, we follow established fully connected architectures and the unsupervised pretraining protocol in [3, 15]: encoder-based methods use a 500–500–2000 backbone with embedding dimension 10 except 20 for CIFAR100-20, while VanillaDCC and VolMaxDCC use their native 512–512 classifier architectures. For baselines, we use reported best settings where

Table 1: Comparative performance (%) (ACC, NMI, ARI) across datasets for external baselines and our progressive models (ProbPair, Weighted ProbPair, and ECI-PP) under `lv0.01` and `multi3` settings, with corruption probability 0.3 and 9k constraints. Blue and black represent training and test results, respectively. Best results are in **bold**, and second-best are underlined.

			External baselines				Our progressive models		
			Vanilla-DCC	VolMax-DCC	CIDEC	SpherePair	ProbPair	Weighted ProbPair	**ECI-PP** (Full)
Single-expert `lv0.01`	CIFAR100-20	ACC	16.8, 17.0	46.1, 45.9	14.3, 13.6	50.2, 50.5	49.4, 49.5	50.3, 50.8	**52.7**, **52.9**
		NMI	13.5, 14.6	45.6, 46.1	11.0, 11.2	43.0, 44.2	43.2, 44.2	44.5, 45.7	**49.2**, **49.9**
		ARI	6.0, 6.3	30.4, 30.5	2.8, 2.3	32.1, 32.6	30.6, 30.9	31.5, 32.1	**36.2**, **36.5**
	CIFAR10	ACC	38.3, 38.7	82.5, 82.5	44.3, 44.0	84.9, 85.7	85.9, 86.0	86.6, 86.7	**87.8**, **87.8**
		NMI	29.9, 31.2	71.9, 72.2	36.1, 37.6	72.4, 74.1	75.0, 75.4	76.2, 76.8	**79.0**, **79.1**
		ARI	23.2, 24.0	68.3, 68.3	21.1, 21.3	71.2, 72.5	73.0, 73.2	74.3, 74.5	**76.7**, **76.8**
	FMNIST	ACC	39.8, 40.2	66.5, 65.5	36.1, 35.7	72.0, 71.2	73.6, 72.5	**74.6**, **73.6**	72.4, 71.1
		NMI	30.1, 31.3	57.2, 56.8	29.7, 30.5	60.4, 60.9	62.9, 62.9	63.9, 64.0	**65.7**, **65.0**
		ARI	21.5, 22.2	46.4, 45.2	15.7, 15.6	53.0, 52.3	54.8, 53.7	55.3, **54.3**	**55.5**, 53.9
	ImageNet10	ACC	39.0, 41.5	87.4, 88.1	63.5, 68.3	85.1, 89.1	89.2, 90.5	89.1, 90.3	**92.8**, **93.1**
		NMI	27.6, 33.7	77.8, 79.2	47.2, 57.5	71.0, 79.2	80.1, 83.2	80.3, 82.7	**88.1**, **88.5**
		ARI	21.1, 25.2	75.2, 76.1	35.5, 42.2	70.7, 78.2	78.5, 80.7	78.4, 80.2	**85.8**, **86.2**
	MNIST	ACC	38.9, 40.3	75.6, 76.9	47.0, 46.7	87.8, 89.1	89.0, 90.0	89.9, 90.8	**90.7**, **91.0**
		NMI	29.5, 32.2	59.3, 61.6	41.4, 43.7	75.4, 78.0	76.9, 79.0	78.3, 80.2	**80.7**, **81.6**
		ARI	23.1, 24.8	56.5, 58.7	24.7, 24.2	75.4, 77.8	77.4, 79.4	79.1, 80.9	**80.8**, **81.5**
	REUTERS	ACC	69.7, 77.0	71.7, 76.8	72.9, 79.3	77.9, 81.8	78.2, 80.8	80.4, 82.7	**85.1**, **86.6**
		NMI	30.5, 43.3	34.3, 43.4	35.7, 48.1	48.7, 56.3	53.2, 58.3	54.5, 59.2	**61.4**, **64.4**
		ARI	36.8, 50.4	40.9, 50.4	41.8, 54.9	52.9, 61.1	55.5, 61.0	58.6, 63.4	**66.6**, **70.0**
	STL10	ACC	35.5, 37.8	79.0, 80.1	57.8, 63.3	76.1, 79.7	81.6, 83.4	81.6, 83.1	**85.4**, **85.8**
		NMI	22.9, 27.6	66.7, 69.3	39.3, 48.5	59.5, 66.5	68.2, 72.1	67.9, 71.6	**75.8**, **76.6**
		ARI	17.1, 19.8	62.8, 64.7	29.8, 37.3	57.2, 63.3	65.8, 68.9	65.8, 68.4	**72.8**, **73.4**
	RCV1-10	ACC	46.2, 46.4	61.3, 61.8	42.5, 44.3	**71.2**, **71.4**	52.7, 53.0	55.2, 55.1	69.0, 68.9
		NMI	15.5, 15.9	35.7, 36.7	29.5, 30.0	60.5, 61.3	53.8, 54.1	57.4, 57.5	**65.5**, **65.8**
		ARI	19.8, 20.1	44.7, 45.4	26.2, 28.3	61.0, 61.4	43.6, 43.7	47.1, 47.0	**62.0**, **62.0**
Multi-expert `multi3`	CIFAR100-20	ACC	15.4, 15.8	42.8, 42.5	12.8, 12.1	44.3, 44.4	43.3, 43.4	46.1, 46.5	**48.6**, **48.8**
		NMI	12.8, 13.6	43.4, 43.8	10.0, 10.5	40.5, 41.5	40.0, 40.9	41.8, 43.0	**48.1**, **48.5**
		ARI	5.8, 5.8	27.1, 27.1	2.1, 2.0	27.1, 27.6	26.0, 26.2	28.0, 28.5	**32.9**, **32.9**
	CIFAR10	ACC	32.4, 32.8	76.5, 76.6	33.2, 31.6	73.8, 74.5	81.3, 81.7	81.4, 81.7	**81.6**, **82.1**
		NMI	29.1, 30.9	73.1, 73.2	28.7, 29.1	64.7, 66.5	71.5, 72.3	72.0, 72.6	**77.8**, **77.8**
		ARI	21.2, 22.4	65.8, 65.9	12.3, 12.1	59.0, 60.5	67.1, 67.8	67.9, 68.4	**73.1**, **73.0**
	FMNIST	ACC	35.2, 35.6	59.4, 59.1	28.6, 28.8	61.0, 61.3	66.6, 66.7	**68.0**, **67.6**	65.3, 64.8
		NMI	31.4, 32.8	59.2, 59.0	28.3, 30.6	56.6, 57.8	60.7, 61.2	61.7, 62.0	**66.0**, **65.6**
		ARI	20.4, 21.3	45.0, 44.3	14.5, 15.4	46.8, 47.4	49.9, 50.0	51.9, 51.8	**53.4**, **52.8**
	ImageNet10	ACC	31.8, 34.2	89.9, 90.5	39.4, 38.8	80.7, 85.0	87.4, 89.2	90.1, 91.3	**92.4**, **92.6**
		NMI	22.0, 26.8	**87.8**, **88.7**	31.0, 34.8	65.6, 73.8	76.1, 80.1	80.1, 83.0	87.7, 88.1
		ARI	15.8, 18.7	83.9, 84.9	14.3, 15.6	63.4, 71.0	75.0, 78.3	79.8, 82.0	**85.4**, **85.7**
	MNIST	ACC	32.2, 33.3	67.1, 67.8	31.8, 31.0	78.6, 80.0	80.5, 81.8	**85.1**, **86.1**	84.7, 85.1
		NMI	26.5, 29.3	57.5, 58.9	28.8, 30.2	70.8, 74.0	72.1, 74.7	75.0, 77.2	**79.9**, **80.7**
		ARI	18.4, 20.1	50.9, 52.1	13.2, 13.4	66.3, 69.1	68.7, 71.2	73.4, 75.5	**77.1**, **77.5**
	REUTERS	ACC	55.3, 59.2	67.5, 68.0	57.3, 59.1	62.1, 69.9	74.5, 79.0	81.0, 83.9	**88.3**, **89.6**
		NMI	14.7, 22.8	45.4, 46.5	18.3, 25.7	25.6, 37.2	44.6, 52.8	51.5, 57.9	**65.4**, **68.7**
		ARI	19.2, 26.2	47.0, 47.9	20.4, 23.1	28.0, 40.7	50.2, 58.6	57.9, 64.2	**72.9**, **76.0**
	STL10	ACC	29.2, 30.7	71.0, 71.1	31.4, 31.6	67.8, 71.4	81.1, 83.0	83.1, 84.3	**87.9**, **87.7**
		NMI	21.5, 26.0	70.2, 69.9	24.9, 31.7	53.2, 59.7	66.1, 70.2	69.4, 72.0	**78.3**, **78.5**
		ARI	15.2, 17.7	59.5, 59.1	13.6, 16.9	47.8, 53.3	63.7, 67.1	67.4, 69.4	**75.9**, **75.8**
	RCV1-10	ACC	44.3, 44.6	48.3, 49.0	32.3, 32.0	**55.7**, **55.9**	43.4, 43.5	47.8, 47.9	53.4, 53.3
		NMI	14.3, 14.7	31.4, 32.1	14.1, 13.5	51.9, 52.7	46.6, 47.4	51.6, 52.0	**58.9**, **59.0**
		ARI	17.8, 18.0	30.2, 30.9	10.8, 10.3	43.6, 44.2	34.4, 34.7	38.2, 38.4	**46.2**, **46.1**

available; VolMaxDCC follows its original validation-based search with 1,000 training samples held out. Except in sensitivity analysis, ECI-PP uses **one global setting** without dataset-specific tuning: 5 estimator folds, Corrector 64–16, $\lambda_{\text{cor}} = 0.5$, $\gamma = 10$, $n_0 = 10$, $\xi = 20$, and $\lambda_{\text{rec}} = 0.02$.

Full experimental details are provided in Appendix C to support reproducibility.

5.2 Experimental results

Main comparison. Table 1 compares our ECI-PP against baselines under the `lv0.01` single-expert and `multi3` multi-expert settings, both with corruption probability 0.3 and 9k constraints; additional constraint-budget results appear in Appendix D.1. Overall, ECI-PP is strongest, ranking first in 41 out of 48 test entries (2 expert regimes$\times$8 datasets$\times$3 metrics) and within the top two in $46/48$, while one of the ProbPair-family methods is best in $45/48$, supporting the ProbPair formulation under UPCC. Under single-expert supervision, ECI-PP outperforms the strong non-ProbPair baseline, SpherePair, in most entries (4–10% absolute NMI margins); further comparisons without pretraining make this pattern clearer (Appendix D.2). Under multi-expert supervision, the NMI margin over SpherePair widens to 6–32%. Appendix D.3 further compares baselines with an expert-aware ensemble wrapper. As an ablation signal, Weighted ProbPair shows a clearer advantage over ProbPair in the multi-expert setting, suggesting that information-based weighting is more useful when low-decisiveness constraints can indicate expert unfamiliarity. A remaining caveat is the severely imbalanced RCV1-10: SpherePair retains the best ACC, reflecting the advantage of its angular geometry for imbalanced pair structures, while ECI-PP still gives the best NMI/ARI, indicating stronger grouping consistency.

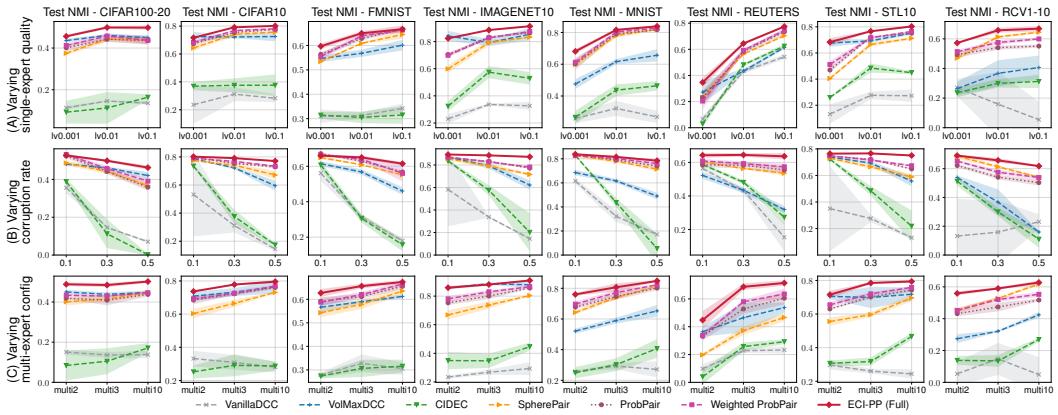


Figure 2: Test NMI performance (mean±std over 5 runs) of external baselines and our progressive models (ProbPair, Weighted ProbPair, and ECI-PP) across datasets under varying (A) expert quality, (B) corruption probability, and (C) multi-expert configuration, with 9k constraints.

Table 2: CIFAR-10H performance under human-derived supervision (%; mean±std over 5 runs). † marks pretrain-capable methods without pretraining; best/second-best are bold/underlined.

	External baselines				**Our progressive models**		
	VanillaDCC	VolMaxDCC	CIDEC†	SpherePair†	ProbPair†	Weighted ProbPair†	**ECI-PP** (Full)†
ACC	81.2±4.0	81.9±7.0	73.7±6.5	85.5±0.8	86.3±0.5	**87.1±0.6**	**87.1±0.7**
NMI	76.5±2.0	75.6±4.1	72.3±3.4	76.3±0.8	**78.4±0.4**	**78.4±0.5**	78.1±0.5
ARI	71.0±3.1	69.7±7.3	63.6±5.2	73.2±1.0	74.8±0.4	**75.3±0.8**	**75.3±0.9**

Robustness across supervision conditions. We further stress-test all methods by varying one supervision factor in UPCC at a time. Around the 9k default supervision in Table 1, Fig. 2 reports test NMI as we vary (A) expert quality, (B) corruption probability, and (C) multi-expert configuration; ACC/ARI results are deferred to Appendix D.4. Across the three sweeps, the trends largely follow the expected supervision-quality ordering: stronger experts, lower corruption, and multi-expert settings with reduced expert blind spots (as detailed in Appendix C.3) generally yield better performance, while the gap between ECI-PP and strong ProbPair-family baselines often narrows in easier regimes. ECI-PP is broadly stable, especially as corruption increases, where most baselines degrade sharply.

Complementary tests further examine whether this robustness extends beyond the default simulated supervision process. On CIFAR-10H, ECI-PP remains among the strongest methods across all three clustering metrics under human-derived probabilistic supervision (Table 2). Across all eight main datasets, ECI-PP also retains the strongest overall performance under generator misspecification with alternative relation constructions and corruption mechanisms (Appendix D.5), indicating that its advantage is not tied to a particular supervision generator.

Held-out diagnostics. We examine whether ECI-PP's internal signals evolve as intended on sample-disjoint held-out constraints. Fig. 3 shows the MNIST case under `multi3` with corruption rate 0.3, using benchmark labels only to form an external hard co-membership reference for diagnostics, rather than the canonical aleatoric relation $R^\star$. Panels (A–C) report Brier scores of the *estimated*, *corrected*, and *integrated* relations, and panel (D) reports the *residual discrepancy* between corrected relations and estimator beliefs on uncorrupted constraints; lower is better. Panels (E,F) compare reliability-aware screening with the injected corruption indicator using AUC/AP, where higher values indicate better corruption separation. (G,H) repeat the same screening diagnostic with oracle-generated clean supervision before corruption, removing the expert epistemic uncertainty present in (E,F). The curves improve rapidly and then stabilize: relation Brier scores decrease, the post-correction residual discrepancy drops, and the screening signal aligns with corruption above chance, while the oracle-setting curves in (G,H) more closely track corruption as expected. Overall, these held-out trends are consistent with the intended behavior of the ECI framework, from internal relation estimates to reliability-aware screening; full cross-dataset diagnostics are in Appendix D.6.

Component ablation and sensitivity. We assess the contribution of the main ECI-PP components under the component-level protocol detailed in Appendix C.4.

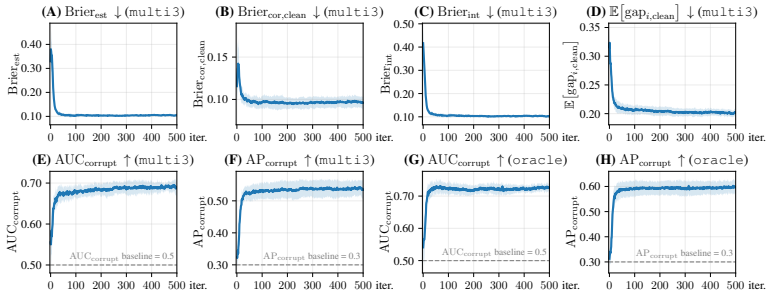


Figure 3: Held-out diagnostic evolution over 500 training iterations on MNIST under `multi3` with corruption rate 0.3 (mean±std over 5 runs). (A–C) Brier scores for estimated, corrected, and integrated relations; (D) residual discrepancy between corrected relations and estimator beliefs on uncorrupted constraints; (E,F) AUC/AP measuring the alignment between reliability-aware screening and injected corruption; (G,H) the same screening diagnostic with oracle-generated pairwise supervision before corruption. Lower is better for (A–D), while higher is better for (E–H).

Table 3: Ablation under the pretrained `multi3` setting ($p_{\text{corr}} = 0.3$, 9k constraints). Results are mean±std over five runs; best and second-best results are bold and underlined, respectively.

Dataset	Metric	Weighted ProbPair	w/o Cross-fitting	w/o Corrector	w/o Screening	1-pass ECI	**ECI-PP** (Full)
CIFAR100-20	ACC	46.4±1.9	45.8±2.5	48.2±1.5	45.8±1.9	48.8±1.2	**49.1±0.9**
	NMI	43.2±0.6	42.1±1.1	46.3±0.6	45.7±0.7	47.7±0.6	**48.1±0.6**
	ARI	28.4±1.0	28.4±1.7	31.9±1.2	30.6±1.3	33.4±0.9	**33.5±0.6**
REUTERS	ACC	83.7±4.0	81.6±2.2	85.7±0.8	88.1±1.6	87.9±0.4	**89.5±1.1**
	NMI	59.2±3.8	53.5±3.7	61.0±1.9	68.0±2.0	65.5±0.6	**69.7±2.3**
	ARI	65.0±5.7	60.4±3.9	67.7±1.8	74.5±3.1	73.2±0.9	**76.5±2.5**

Using CIFAR100-20 and Reuters as complementary image and text benchmarks with different class-balance profiles, Table 3 shows that full ECI-PP achieves the strongest performance across all three metrics on both datasets. Removing cross-fitting produces the largest or near-largest degradation overall, particularly on Reuters, highlighting the importance of out-of-fold beliefs as a reliable reference for subsequent refinement. Removing either the Corrector or reliability screening also consistently weakens performance, supporting their complementary roles in structured correction and reliability-aware integration. The 1-pass ECI variant (a single ECI cycle without iterative feedback) remains competitive but consistently below full ECI-PP, while Weighted ProbPair is clearly weaker, indicating additional benefits from iterative feedback and the complete ECI pipeline.

We further examine sensitivity around the shared default configuration in Appendix D.7. The results show broad stability: larger K yields only mild gains relative to its cross-fitting cost, information-based warm-up is mildly helpful, small Correctors with moderate regularization are sufficient, and the screening and boundary parameters (γ, n_0, ξ) are generally insensitive. Reconstruction is more dataset-dependent, while ECI-PP remains robust without dataset-specific hyperparameter tuning.

6 Conclusion

This paper addressed a realistic probabilistic constrained clustering setting, where pairwise supervision entangles aleatoric, epistemic, and stochastic factors. We formalized the constraint-observation process and its conditional identifiability, introduced ProbPair for angular learning from probabilistic relations, and built ECI-PP as a practical framework for such supervision. While existing methods largely rely on idealized constraints, ECI-PP outperforms state-of-the-art alternatives and remains robust under fallible, heterogeneous, and corrupted supervision without dataset-specific tuning.

These results come with several qualifications. The identifiability analysis is population-level and conditional; fully factor-specific finite-sample estimation would require stronger assumptions or additional observations beyond the current surrogate residual structure. Accordingly, the Corrector and reliability signal are practical surrogates rather than estimators of individual latent factors; a more theory-aligned extension would model structured correction, residual epistemic variation, and corruption-aware screening separately. Finally, richer Corrector pair descriptors and principled stopping criteria may further improve robustness and training adaptivity.

Acknowledgments and Disclosure of Funding

We are grateful to the anonymous reviewers for their comments, which improved the presentation of this paper.

Funding. S.J. Zhang was supported by the University of Manchester–China Scholarship Council Joint Scholarship. We also acknowledge the assistance provided by Research IT and the use of the Computational Shared Facility at The University of Manchester.

Competing interests. The authors declare no competing interests.

References

[1] Yucen Luo, Tian Tian, Jiaxin Shi, Jun Zhu, and Bo Zhang. Semi-crowdsourced clustering with deep generative models. *Advances in Neural Information Processing Systems*, 31, 2018. 1, 2

[2] Laura Manduchi, Kieran Chin-Cheong, Holger Michel, Sven Wellmann, and Julia Vogt. Deep conditional gaussian mixture model for constrained clustering. *Advances in Neural Information Processing Systems*, 34:11303–11314, 2021. 1, 2, 25

[3] Shaojie Zhang and Ke Chen. Angular constraint embedding via spherepair loss for constrained clustering. In *Advances in Neural Information Processing Systems 39*, 2025. 1, 2, 5, 7, 19, 20, 25, 26, 27, 42

[4] Ian Davidson and Sugato Basu. A survey of clustering with instance level constraints. *ACM Transactions on Knowledge Discovery from data*, 1(1-41):2–42, 2007. 1

[5] Germán González-Almagro, Daniel Peralta, Eli De Poorter, José-Ramón Cano, and Salvador García. Semi-supervised constrained clustering: an in-depth overview, ranked taxonomy and future research directions. *Artificial Intelligence Review*, 58:157, 2025. 1

[6] Kiri Wagstaff and Claire Cardie. Clustering with instance-level constraints. *AAAI/IAAI*, 1097(577-584):197, 2000. 1

[7] Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schrödl, et al. Constrained k-means clustering with background knowledge. In *ICML*, volume 1, pages 577–584, 2001. 1

[8] Zhengdong Lu and Todd Leen. Semi-supervised learning with penalized probabilistic clustering. *Advances in Neural Information Processing Systems*, 17, 2004. 1

[9] Mikhail Bilenko, Sugato Basu, and Raymond J Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of the twenty-first International Conference on Machine Learning*, page 11, 2004. 1

[10] Sugato Basu, Arindam Banerjee, and Raymond J Mooney. Active semi-supervision for pairwise constrained clustering. In *Proceedings of the 2004 SIAM International Conference on Data Mining*, pages 333–344. SIAM, 2004. 1

[11] Brian Kulis, Sugato Basu, Inderjit Dhillon, and Raymond Mooney. Semi-supervised graph clustering: a kernel approach. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 457–464, 2005. 1

[12] Yen-Chang Hsu, Zhaoyang Lv, Joel Schlosser, Phillip Odom, and Zsolt Kira. A probabilistic constrained clustering for transfer learning and image category discovery. In *Proceedings of the CVPR 2018 Deep-Vision Workshop*, 2018. 1, 2

[13] Yazhou Ren, Kangrong Hu, Xinyi Dai, Lili Pan, Steven CH Hoi, and Zenglin Xu. Semi-supervised deep embedded clustering. *Neurocomputing*, 325:121–130, 2019. 1, 19, 25, 26

[14] Hongjing Zhang, Tianyang Zhan, Sugato Basu, and Ian Davidson. A framework for deep constrained clustering. *Data Mining and Knowledge Discovery*, 35:593–620, 2021. 1, 2, 7, 19, 20, 25, 26

[15] Tri Nguyen, Shahana Ibrahim, and Xiao Fu. Deep clustering with incomplete noisy pairwise annotations: A geometric regularization approach. In *International Conference on Machine Learning*, pages 25980–26007. PMLR, 2023. 1, 2, 7, 19, 25, 27

[16] Sharon Fogel, Hadar Averbuch-Elor, Daniel Cohen-Or, and Jacob Goldberger. Clustering-driven deep embedding with pairwise constraints. *IEEE computer graphics and applications*, 39(4):16–27, 2019. 1, 2

[17] Abu Quwsar Ohi, Muhammad Firoz Mridha, Farisa Benta Safir, Md Abdul Hamid, and Muhammad Mostafa Monowar. Autoembedder: A semi-supervised dnn embedding system for clustering. *Knowledge-Based Systems*, 204:106190, 2020. 1, 2

[18] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110(3):457–506, 2021. 1

[19] Yen-Chang Hsu, Zhaoyang Lv, Joel Schlosser, Phillip Odom, and Zsolt Kira. Multi-class classification without multi-class labels. In *International Conference on Learning Representations*, 2019. 2, 7, 19

[20] Marek Śmieja, Łukasz Struski, and Mário AT Figueiredo. A classification-based approach to semi-supervised clustering with pairwise constraints. *Neural Networks*, 127:193–203, 2020. 2

[21] Tian Tian, Jie Zhang, Xiang Lin, Zhi Wei, and Hakon Hakonarson. Model-based deep embedding for constrained clustering analysis of single cell rna-seq data. *Nature communications*, 12(1):1873, 2021. 2

[22] Jann Goschenhofer, Bernd Bischl, and Zsolt Kira. Constraintmatch for semi-constrained clustering. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10. IEEE, 2023. 2

[23] Yi Wen, Suyuan Liu, Xinhang Wan, Siwei Wang, Ke Liang, Xinwang Liu, Xihong Yang, and Pei Zhang. Efficient multi-view graph clustering with local and global structure preservation. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 3021–3030, 2023. 2

[24] Jian Dai, Zhenwen Ren, Yunzhi Luo, Hong Song, and Jian Yang. Tensorized anchor graph learning for large-scale multi-view clustering. *Cognitive Computation*, 15(5):1581–1592, 2023. 2

[25] Suyuan Liu, Qing Liao, Siwei Wang, Xinwang Liu, and En Zhu. Robust and consistent anchor graph learning for multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering*, 2024. 2

[26] Martin HC Law, Alexander Topchy, and Anil K Jain. Model-based clustering with probabilistic constraints. In *Proceedings of the 2005 SIAM International Conference on Data Mining*, pages 641–645. SIAM, 2005. 2

[27] Martin HC Law, Alexander Topchy, and Anil K Jain. Clustering with soft and group constraints. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 662–670. Springer, 2004. 2

[28] Irene Diaz-Valenzuela, M Amparo Vila, and Maria J Martin-Bautista. On the use of fuzzy constraints in semisupervised clustering. *IEEE Transactions on Fuzzy Systems*, 24(4):992–999, 2015. 2

[29] Zhen Wang, Shan-Shan Wang, Lan Bai, Wen-Si Wang, and Yuan-Hai Shao. Semisupervised fuzzy clustering with fuzzy pairwise constraints. *IEEE Transactions on Fuzzy Systems*, 30(9):3797–3811, 2021. 2

[30] Jian-Ping Mei, Huajiang Lv, Jiuwen Cao, and Weihua Gong. Pairwise constrained fuzzy clustering: Relation, comparison and parallelization. *International Journal of Fuzzy Systems*, 21(6):1938–1949, 2019. 2

[31] Jian-Ping Mei and Huajiang Lv. Semi-supervised fuzzy c-means regularized with pairwise constraints. In *2018 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, pages 781–786. IEEE, 2018. 2

[32] Violaine Antoine, Benjamin Quost, M-H Masson, and Thierry Denoeux. Cevclus: evidential clustering with instance-level constraints for relational data. *Soft Computing*, 18(7):1321–1335, 2014. 2

[33] Feng Li, Shoumei Li, and Thierry Denœux. k-cevclus: Constrained evidential clustering of large dissimilarity data. *Knowledge-Based Systems*, 142:29–44, 2018. 2

[34] Dan Pelleg and Dorit Baras. K-means with large and noisy constraint sets. In *European Conference on Machine Learning*, pages 674–682. Springer, 2007. 2

[35] Thiago F Covoes, Eduardo R Hruschka, and Joydeep Ghosh. A study of k-means-based algorithms for constrained clustering. *Intelligent Data Analysis*, 17(3):485–505, 2013. 2

[36] Xiatian Zhu, Chen Change Loy, and Shaogang Gong. Constrained clustering with imperfect oracles. *IEEE transactions on neural networks and learning systems*, 27(6):1345–1357, 2015. 2

[37] Hongfu Liu, Zhiqiang Tao, and Yun Fu. Partition level constrained clustering. *IEEE transactions on pattern analysis and machine intelligence*, 40(10):2469–2483, 2017. 2

[38] Yale Chang, Junxiang Chen, Michael H Cho, Peter J Castaldi, Edwin K Silverman, and Jennifer G Dy. Multiple clustering views from multiple uncertain experts. In *International Conference on Machine Learning*, pages 674–683. PMLR, 2017. 2

[39] Hongjing Zhang, Sugato Basu, and Ian Davidson. A framework for deep constrained clustering-algorithms and advances. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2019, Würzburg, Germany, September 16–20, 2019, Proceedings, Part I*, pages 57–72. Springer, 2020. 2

[40] Alexander Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1):20–28, 1979. 2

[41] Filipe Rodrigues and Francisco Pereira. Deep learning from crowds. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018. 2

[42] Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier Movellan, and Paul Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. *Advances in Neural Information Processing Systems*, 22, 2009. 2

[43] Peter Welinder, Steve Branson, Pietro Perona, and Serge Belongie. The multidimensional wisdom of crowds. *Advances in Neural Information Processing Systems*, 23, 2010. 2

[44] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 7, 18

[45] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 7, 18

[46] Jianlong Chang, Lingfeng Wang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. Deep adaptive image clustering. In *Proceedings of the IEEE international conference on computer vision*, pages 5879–5887, 2017. 7, 18

[47] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 7, 18

[48] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. 7, 19

[49] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International Conference on Machine Learning*, pages 478–487. PMLR, 2016. 7, 18, 19, 20, 26

[50] Joshua C Peterson, Ruairidh M Battleday, Thomas L Griffiths, and Olga Russakovsky. Human uncertainty makes classification more robust. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9617–9626, 2019. 7, 19

[51] Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. Double/debiased machine learning for treatment and structural parameters. *Econometrics Journal*, 21:C1–C68, 2018. 18

[52] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009. 18

[53] Xifeng Guo, Long Gao, Xinwang Liu, and Jianping Yin. Improved deep embedded clustering with local structure preservation. In *IJCAI*, volume 17, pages 1753–1759, 2017. 19, 20, 26

[54] Yunfan Li, Peng Hu, Zitao Liu, Dezhong Peng, Joey Tianyi Zhou, and Xi Peng. Contrastive clustering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 8547–8555, 2021. 25

[55] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 25

[56] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. 26

[57] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine learning*, pages 1096–1103. ACM, 2008. 26

A Additional theoretical details

A.1 Derivation of the logistic-normal judgment density

Starting from Equation (1), define

$$\mu_{e,ab} := \mathrm{logit}_\varepsilon(R^\star_{ab}) + m_e(\phi_{ab}).$$

Conditionally on $(R^\star_{ab}, e, \phi_{ab})$, we then have

$$\mathrm{logit}\left(y^{\mathrm{jud}}_{e,ab}\right) = \mu_{e,ab} + u_{e,ab}, \qquad u_{e,ab} \sim \mathcal{N}\big(0, \tau^2_e(\phi_{ab})\big),$$

so that

$$\mathrm{logit}\left(y^{\mathrm{jud}}_{e,ab}\right) \sim \mathcal{N}\big(\mu_{e,ab}, \tau^2_e(\phi_{ab})\big).$$

For a generic $y \in (0,1)$, write $v = \mathrm{logit}(y)$. By the change-of-variables formula,

$$p_{\mathrm{jud}}(y \mid R^\star_{ab}, e, \phi_{ab}) = \varphi\Big(v;\, \mu_{e,ab},\, \tau^2_e(\phi_{ab})\Big) \left|\frac{dv}{dy}\right|, \qquad v = \mathrm{logit}(y), \quad y \in (0,1).$$

Using $\frac{dv}{dy} = \frac{d\,\mathrm{logit}(y)}{dy} = \frac{1}{y(1-y)}$, we obtain

$$p_{\mathrm{jud}}(y \mid R^\star_{ab}, e, \phi_{ab}) = \frac{1}{y(1-y)}\, \varphi\Big(\mathrm{logit}(y);\, \mathrm{logit}_\varepsilon(R^\star_{ab}) + m_e(\phi_{ab}),\, \tau^2_e(\phi_{ab})\Big), \qquad y \in (0,1),$$

which is exactly Equation (2).

A.2 Proof of Lemma 3.2

Proof. Fix an expert e with nonzero observation probability. By assumption,

$$m_{\eta_e}(\phi_{ab}) = m_{\eta'_e}(\phi_{ab}) - k_e$$

for all observable triples (a, b, e). Taking expectation over the observable-pair distribution conditional on e yields

$$\mathbb{E}_{(a,b)|e}[m_{\eta_e}(\phi_{ab})] = \mathbb{E}_{(a,b)|e}[m_{\eta'_e}(\phi_{ab})] - k_e.$$

Both expectations vanish by Assumption 3.1, hence $k_e = 0$.[1] Since e was arbitrary over experts with nonzero observation probability, the conclusion holds for every such expert. □

A.3 Proof of Lemma 3.3

Proof. By Equations (2) and (4), the observation law takes the form Equation (5).

Assume that

$$p(\cdot \mid \mu, s, \pi) = p(\cdot \mid \mu', s', \pi')$$

on $(0, 1)$, where $p(y \mid \mu, s, \pi)$ is the density defined in Equation (5). Since $\mathbf{1}_{[0,1]}(y) = 1$ for $y \in (0, 1)$, equality of the two densities means that

$$(1-\pi)\,\frac{1}{y(1-y)}\,\varphi\big(\operatorname{logit}(y); \mu, s^2\big) + \pi = (1-\pi')\,\frac{1}{y(1-y)}\,\varphi\big(\operatorname{logit}(y); \mu', {s'}^2\big) + \pi'$$

for all $y \in (0, 1)$.

Write $v = \operatorname{logit}(y)$, so that $y = \sigma(v)$ and $y(1-y) = \sigma(v)(1-\sigma(v))$. Multiplying both sides by $y(1-y)$ gives

$$(1-\pi)\,\varphi(v; \mu, s^2) + \pi\,\sigma(v)(1-\sigma(v)) = (1-\pi')\,\varphi(v; \mu', {s'}^2) + \pi'\,\sigma(v)(1-\sigma(v))$$

for all $v \in \mathbb{R}$.

As $v \to +\infty$, the Gaussian terms decay faster than e^{-v}, whereas

$$\sigma(v)(1-\sigma(v)) = \frac{e^{-v}}{(1+e^{-v})^2} \sim e^{-v}.$$

Therefore,

$$\lim_{v \to +\infty} e^{v}\Big[(1-\pi)\,\varphi(v; \mu, s^2) + \pi\,\sigma(v)(1-\sigma(v))\Big] = \pi,$$

and similarly the right-hand side tends to π'. Hence $\pi = \pi'$.

Subtracting the common contamination term then yields

$$\varphi(v; \mu, s^2) = \varphi(v; \mu', {s'}^2) \qquad \text{for all } v \in \mathbb{R}.$$

Injectivity of the Gaussian family implies $\mu = \mu'$ and $s = s'$. Therefore $(\mu, s, \pi) = (\mu', s', \pi')$, proving the claim. □

A.4 Boundary cases for structural separability

Assumption 3.4 concerns the decomposition of the identifiable location term $\operatorname{logit}_\varepsilon(r_\theta(x_a, x_b)) + m_{\eta_e}(\phi_{ab})$ into a canonical relation component and an expert-specific epistemic component. Two boundary cases clarify this requirement. (i) If the pair descriptor ϕ_{ab} is uninformative, for example constant over observable pairs for a given expert, then $m_{\eta_e}(\phi_{ab})$ reduces to an expert-specific constant and cannot capture pair-dependent epistemic effects; the intended epistemic component is then not represented by the chosen model class. (ii) Conversely, if ϕ_{ab} is overly rich and the function class for $m_{\eta_e}(\phi_{ab})$ overlaps with that of $\operatorname{logit}_\varepsilon(r_\theta(x_a, x_b))$, then shared pair-dependent variation can be moved between the two terms while preserving their sum. For example, a sufficiently small zero-mean shift lying in this overlap can be added to the canonical location term and subtracted from the expert-specific term without violating centering or changing the observation law. The condition therefore rules out such interchangeability, ensuring that canonical aleatoric variation and structured expert-specific variation play distinct explanatory roles on the observable support.

[1] The centering is imposed conditionally on each expert because the remaining additive ambiguity after fixing the total location is expert-wise: global centering would only fix the average shift across experts, not each expert-specific shift. Thus, under the present assumptions, global centering would not be sufficient for identifiability unless one further assumes that different experts observe sufficiently overlapping pair regions.

A.5 Proof of Theorem 3.5

Proof. Assume that the induced conditional laws coincide under Ξ and Ξ' for every observable triple (a, b, e). For each such triple, define

$$\mu_\Xi(a,b,e) := \mathrm{logit}_\varepsilon\big(r_\theta(x_a,x_b)\big) + m_{\eta_e}(\phi_{ab}), \qquad s_\Xi(a,b,e) := \tau_{\zeta_e}(\phi_{ab}),$$

and define $\mu_{\Xi'}(a,b,e)$ and $s_{\Xi'}(a,b,e)$ analogously. By Equations (4) and (5), the conditional law of the observed response for (a, b, e) is exactly

$$p\big(y \mid \mu_\Xi(a,b,e), s_\Xi(a,b,e), \pi_e\big),$$

and similarly under Ξ'. Since the two conditional laws coincide, Lemma 3.3 implies that for every observable triple (a, b, e),

$$\mu_\Xi(a,b,e) = \mu_{\Xi'}(a,b,e), \qquad s_\Xi(a,b,e) = s_{\Xi'}(a,b,e), \qquad \pi_e = \pi'_e.$$

In particular,

$$\mathrm{logit}_\varepsilon\big(r_\theta(x_a,x_b)\big) + m_{\eta_e}(\phi_{ab}) = \mathrm{logit}_\varepsilon\big(r_{\theta'}(x_a,x_b)\big) + m_{\eta'_e}(\phi_{ab})$$

for all observable triples (a, b, e).

Applying Assumption 3.4, for each expert e there exists a constant $k_e \in \mathbb{R}$ such that for all observable triples (a, b, e),

$$\mathrm{logit}_\varepsilon\big(r_\theta(x_a,x_b)\big) = \mathrm{logit}_\varepsilon\big(r_{\theta'}(x_a,x_b)\big) + k_e, \qquad m_{\eta_e}(\phi_{ab}) = m_{\eta'_e}(\phi_{ab}) - k_e.$$

By Lemma 3.2, each k_e on the observable expert support must be zero. Therefore,

$$\mathrm{logit}_\varepsilon\big(r_\theta(x_a,x_b)\big) = \mathrm{logit}_\varepsilon\big(r_{\theta'}(x_a,x_b)\big) \qquad \text{for all } (a,b) \in \mathcal{P}_{\mathrm{obs}}.$$

Since $r_\theta(x_a,x_b), r_{\theta'}(x_a,x_b) \in [\varepsilon, 1-\varepsilon]$ on $\mathcal{P}_{\mathrm{obs}}$, the clamped logit satisfies $\mathrm{logit}_\varepsilon(u) = \mathrm{logit}(u)$ on this interval and is therefore injective there. It follows that

$$r_\theta(x_a,x_b) = r_{\theta'}(x_a,x_b) \qquad \text{for all } (a,b) \in \mathcal{P}_{\mathrm{obs}},$$

which proves the theorem. □

B Algorithm and complexity analysis

B.1 Algorithm

We provide the detailed ECI-PP procedure introduced in Section 4, summarized in Algorithm 1.

B.2 Computational complexity analysis

For an Estimator/Integrator backbone $\big(f, g, m, T\big)$, let c_f and c_g denote the per-instance training costs of the encoder f and decoder g, respectively, and let $c_{m,T} = O(1)$ denote the cost of updating the scalar readout parameters (m, T). We write $c_{\mathrm{pair}} = O(c_f + c_{m,T})$ for the amortized per-constraint cost of the ProbPair-style pairwise term, and $c_{\mathrm{rec}} = O(c_f + c_g)$ for the per-instance reconstruction cost. Thus, one backbone epoch over constraints $\mathcal{C}$ and instances $\mathcal{X}$ costs $O(|\mathcal{C}|c_{\mathrm{pair}} + |\mathcal{X}|c_{\mathrm{rec}})$. In particular, the pairwise term scales with the number of constraints, while the reconstruction term scales linearly with $|\mathcal{X}|$ as the usual autoencoder regularization cost. Notably, each Corrector $\mathrm{Cor}_e = q_{\nu_e}$ is a small expert-specific MLP, rather than an encoder–decoder backbone, and its capacity is intentionally kept more limited than that of the Estimator/Integrator networks so that it learns only shared structured expert-specific patterns rather than arbitrary fitting. We write c_{cor} for its amortized per-constraint update cost.

Under this notation, with N_{wu} warm-up epochs and N_{ep} total epochs, the warm-up stage costs $O\big(N_{\mathrm{wu}}(K+1)\big(|\mathcal{C}|c_{\mathrm{pair}} + |\mathcal{X}|c_{\mathrm{rec}}\big) + J_{\mathrm{cor}}|\mathcal{C}|c_{\mathrm{cor}}\big)$, where J_{cor} is the average number of optimization passes used when fitting the Correctors. Each of the $T_{\mathrm{ref}} := N_{\mathrm{ep}} - N_{\mathrm{wu}}$ subsequent refinement rounds costs $O\big((K+1)\big(|\mathcal{C}|c_{\mathrm{pair}} + |\mathcal{X}|c_{\mathrm{rec}}\big) + J_{\mathrm{cor}}|\mathcal{C}|c_{\mathrm{cor}}\big)$. The total training complexity is

$$O\Big(\big(N_{\mathrm{wu}} + T_{\mathrm{ref}}\big)(K+1)\big(|\mathcal{C}|c_{\mathrm{pair}} + |\mathcal{X}|c_{\mathrm{rec}}\big) + \big(1 + T_{\mathrm{ref}}\big)J_{\mathrm{cor}}|\mathcal{C}|c_{\mathrm{cor}}\Big).$$

Algorithm 1 ECI-PP for learning under UPCC

Require: training data $\mathcal{X} = \{x_j\}_{j=1}^{|\mathcal{X}|}$; constraints $\mathcal{C} = \{(a_i, b_i, e_i, y_i)\}_{i=1}^{|\mathcal{C}|}$; number of experts E; number of estimator folds K; total training epochs N_{ep}; warm-up epochs N_{wu}; hyperparameters λ_{rec} (reconstruction weight), λ_{cor} (Corrector regularization), ξ (soft-clipping sharpness), γ (screening sharpness), n_0 (Bayesian-confidence scale); clustering routine $\text{Clustering}(\cdot)$

Partition $\mathcal{C}$ into K disjoint folds $\{\mathcal{C}^{(k)}\}_{k=1}^{K}$ and define $\mathcal{I}_e := \{i : e_i = e\}$ for each expert e
Initialize $\{\text{Est}^{(k)} = (f_{\omega_k}, g_{\omega'_k}, m_k, T_k)\}_{k=1}^{K}$, $\{\text{Cor}_e = q_{\nu_e}\}_{e=1}^{E}$, and $\text{Int} = (f_\omega, g_{\omega'}, m, T)$

Stage I: Warm-up
Compute decisiveness weights $\{\kappa_i\}_{i=1}^{|\mathcal{C}|}$ by Eq. (9)
for $n = 1, \ldots, N_{\text{wu}}$ **do**
  for $k = 1, \ldots, K$ **do**
    Update $\text{Est}^{(k)}$ on $\mathcal{C} \setminus \mathcal{C}^{(k)}$ by gradient descent on Eq. (8) with y_i and weights κ_i
  end for
end for
Aggregate holdout predictions from $\{\text{Est}^{(k)}\}_{k=1}^{K}$ to obtain $\{\hat{y}_i^{\text{oof}}\}_{i=1}^{|\mathcal{C}|}$ via Eq. (6)
for $e = 1, \ldots, E$ **do**
  Update Cor_e on $\mathcal{I}_e$ by gradient descent on Eq. (10) until convergence
end for
Compute $\{y_i^{\text{cor}}, \text{gap}_i, w_i\}_{i=1}^{|\mathcal{C}|}$ by Eq. (11)
Set $n_i \leftarrow n_0 w_i$ and compute Bayesian-confidence supervision $\{y_i^{\text{BC}}\}_{i=1}^{|\mathcal{C}|}$ by Eq. (12)
for $n = 1, \ldots, N_{\text{wu}}$ **do**
  Update Int by gradient descent on Eq. (8) with y_i^{BC} and weights w_i
end for

Stage II: Iterative refinement
for $t = N_{\text{wu}} + 1, \ldots, N_{\text{ep}}$ **do**
  Read out Integrator-side relations $\{y_i^{\text{int}}\}_{i=1}^{|\mathcal{C}|}$ from the Integrator embeddings via Eq. (6)
  for $k = 1, \ldots, K$ **do**
    Update $\text{Est}^{(k)}$ on $\mathcal{C} \setminus \mathcal{C}^{(k)}$ by gradient descent on Eq. (8) with y_i^{int} and weights w_i
  end for
  Aggregate updated holdout predictions to refresh $\{\hat{y}_i^{\text{oof}}\}_{i=1}^{|\mathcal{C}|}$
  for $e = 1, \ldots, E$ **do**
    Update Cor_e on $\mathcal{I}_e$ by gradient descent on Eq. (10) until convergence
  end for
  Refresh $\{y_i^{\text{cor}}, \text{gap}_i, w_i\}_{i=1}^{|\mathcal{C}|}$ by Eq. (11)
  Set $n_i \leftarrow n_0 w_i$ and refresh Bayesian-confidence supervision $\{y_i^{\text{BC}}\}_{i=1}^{|\mathcal{C}|}$ by Eq. (12)
  Update Int by gradient descent on Eq. (8) with y_i^{BC} and weights w_i
end for

Stage III: Clustering and unseen-instance assignment
Obtain the final normalized Integrator embeddings $\mathcal{Z}^{\text{int}} \leftarrow \{\text{Norm}(f_\omega(x_j))\}_{j=1}^{|\mathcal{X}|}$
Apply $\text{Clustering}(\mathcal{Z}^{\text{int}})$ to obtain the final partition $\widehat{S}$
Compute cluster centroids $\{\mu_c\}$ from $\mathcal{Z}^{\text{int}}$ and $\widehat{S}$
for each unseen instance $\tilde{x}$ **do**
  Compute $\tilde{z}^{\text{int}} \leftarrow \text{Norm}(f_\omega(\tilde{x}))$
  Assign $\tilde{x}$ to the nearest centroid in $\{\mu_c\}$
end for

The dominant overhead comes from the K-fold cross-fitting of the Estimators, namely the repeated backbone updates scaled by $(K+1)\big(|\mathcal{C}|c_{\text{pair}} + |\mathcal{X}|c_{\text{rec}}\big)$. This overhead is justified by the need for out-of-fold beliefs: the Estimators are introduced not as a redundant ensemble, but to provide model-side references decoupled from the specific constraints they are later used to assess, thereby reducing self-confirmation and overfitting to noisy supervision. This use of held-out predictions is analogous in spirit to debiased/double machine learning [51]: both reduce in-sample contamination, although ECI-PP uses them to obtain cleaner belief proxies under noisy supervision rather than to pursue semiparametric efficiency. By contrast, the expert-specific correction stage does *not* scale as $O(E|\mathcal{C}|)$, because each constraint is routed only to its corresponding expert module and $\sum_{e=1}^{E} |\mathcal{I}_e| = |\mathcal{C}|$. Thus, for a fixed total constraint budget, increasing E does not create an $O(E|\mathcal{C}|)$ bottleneck, and the Corrector workload remains linear in $|\mathcal{C}|$. A separate statistical limitation arises when a large expert pool leaves some $|\mathcal{I}_e|$ too small for stable expert-specific correction; an amortized Corrector with shared parameters and expert embeddings or partial pooling is a natural extension for such sparse-expert regimes.

More broadly, the practical cost profile of ECI-PP should be interpreted together with its training protocol: it does not require an additional expert-aware wrapper in multi-expert settings (see Appendix C.4), and unlike methods such as VolMaxDCC, its performance is not tightly tied to an extensive hyperparameter-search stage. Meanwhile, Appendix D.2 shows that ECI-PP is less dependent on unsupervised pretraining, which can be advantageous in resource-limited settings, and an overall training-time comparison is reported in Appendix E. A discussion of the choice of K is provided in Appendix D.7.

C Details of experimental settings

We provide supplementary information for the experimental setup in Section 5.1, including the benchmark datasets, compared methods, constraint-generation procedure, experimental protocol, and implementation details.

C.1 Datasets

We use eight benchmark datasets covering image and text clustering tasks, with different numbers of classes and class-balance properties. We additionally use CIFAR-10H for evaluation under human-derived probabilistic supervision. Their details are as follows.

CIFAR100-20[2] [44]: A 20-class version of CIFAR100 obtained by using the 20 superclasses as clustering targets. The original CIFAR100 dataset contains 60,000 real-world 32×32 color images from 100 fine-grained classes, grouped into 20 superclasses. In our experiments, the 20 superclasses are treated as ground-truth clusters, with 3,000 images per superclass.

CIFAR10[3] [44]: A natural-image dataset containing 60,000 32×32 color images from 10 object categories, with 6,000 images per category.

FMNIST[4] [45]: A fashion-product image dataset containing 70,000 grayscale 28×28 images from 10 categories. The official split contains 60,000 training images and 10,000 test images.

ImageNet10[5] [46]: A 10-class subset of ImageNet [52], containing 13,000 color images in total, with 1,300 images per class.

MNIST[6] [47]: A handwritten-digit dataset containing 70,000 grayscale 28×28 images from 10 digit classes. The official split contains 60,000 training images and 10,000 test images.

Reuters subset[7] [49]: A text benchmark derived from the RCV1 corpus[8], using tf–idf features over the 2,000 most frequent words. It contains four root categories, *Corporate/Industrial* (CCAT), *Economics* (ECAT), *Government/Social* (GCAT), and *Markets* (MCAT), treated as ground-truth

[2] CIFAR100 webpage: https://www.cs.toronto.edu/~kriz/cifar.html
[3] CIFAR10 webpage: https://www.cs.toronto.edu/~kriz/cifar.html
[4] FMNIST repository: https://github.com/zalandoresearch/fashion-mnist
[5] ImageNet webpage: https://image-net.org/
[6] MNIST webpage: http://yann.lecun.com/exdb/mnist/
[7] Preprocessed Reuters subset: https://github.com/piiswrong/dec
[8] RCV1 webpage: https://trec.nist.gov/data/reuters/reuters.html

clusters. The four categories contain 4,840, 3,470, 2,673, and 1,017 samples, respectively, giving a mildly imbalanced benchmark. The official split contains 10,000 training samples and 2,000 test samples.

STL10[9] [48]: An image dataset containing 13,000 96×96 color images from 10 object categories, with 1,300 images per category.

RCV1-10[10] [3]: An imbalanced 10-category subset of RCV1 constructed from single-label articles in categories C14, C18, C313, C42, E21, E311, GDEF, GODD, GWELF, and M13. It contains 177,669 documents represented by tf–idf features over the 2,000 most frequent word stems. The class distribution is highly skewed, ranging from 903 documents in GWELF to 53,127 documents in M13, making it a benchmark for evaluating clustering under severe class imbalance.

CIFAR-10H[11] [50]: A human-annotated extension of the official 10,000-image CIFAR10 test set, providing aggregate human class-distribution annotations for each image. The images are evenly distributed across the 10 CIFAR10 classes. We use a stratified $80\%/20\%$ train/test split of these 10,000 images, yielding 8,000 training and 2,000 test images. CIFAR-10H is used only for evaluation under human-derived probabilistic supervision.

The eight benchmark datasets used in our main comparison follow common evaluation practice in deep clustering and constrained clustering: MNIST, FMNIST, Reuters subset, CIFAR10, ImageNet10, STL10, and CIFAR100-20 have been widely used in prior deep clustering/DCC studies [49, 53, 13, 14, 15, 3], while RCV1-10 follows the imbalanced text benchmark introduced in [3]. For MNIST, FMNIST, and Reuters subset, we use the official train/test splits, yielding 60,000/10,000, 60,000/10,000, and 10,000/2,000 training/test samples, respectively. For CIFAR100-20, CIFAR10, ImageNet10, STL10, and RCV1-10, we randomly split each dataset into 80% training and 20% testing, yielding 48,000/12,000, 48,000/12,000, 10,400/2,600, 10,400/2,600, and 142,135/35,534 training/test samples, respectively. For CIFAR-10H, we use the stratified $80\%/20\%$ split of its 10,000 annotated CIFAR10 test images described above. All probabilistic constraints for training are generated from the corresponding training split only.

C.2 Compared methods

We summarize the compared methods and their use under probabilistic pairwise supervision. Methods tied to hard binary constraints or discrete signed priors are not included, as they do not natively operate on the real-valued probabilistic supervision considered in UPCC.

VanillaDCC. VanillaDCC [19] is an end-to-end DCC method based on the Meta Classification Likelihood (MCL) loss. Given the soft cluster-assignment vector $\boldsymbol{q}_j = (q_{j1}, \ldots, q_{jC})$ for each instance x_j, the pairwise co-clustering probability of a constrained pair is $P_i^{\mathrm{co}} = \boldsymbol{q}_{a_i} \boldsymbol{q}_{b_i}^\top$. The MCL loss is

$$\mathcal{L}_{\mathrm{MCL}} = -\frac{1}{|\mathcal{C}|} \sum_{i=1}^{|\mathcal{C}|} \left[y_i \log P_i^{\mathrm{co}} + (1 - y_i) \log(1 - P_i^{\mathrm{co}}) \right].$$

Although originally used for binary constraints, this logistic/BCE-form objective can directly take real-valued targets $y_i \in [0, 1]$, so VanillaDCC serves as a simple end-to-end DCC baseline under probabilistic supervision.

VolMaxDCC. VolMaxDCC [15] extends MCL by introducing an explicit noise-modeling mechanism through a confusion-adjusted co-clustering probability. Concretely, it replaces P_i^{co} with $P_i'^{\mathrm{co}} = \boldsymbol{q}_{a_i} B \boldsymbol{q}_{b_i}^\top$, where B is derived from a learnable confusion matrix, and adds a volume maximization term:

$$\mathcal{L}_{\mathrm{VolMax}} = -\frac{1}{|\mathcal{C}|} \sum_{i=1}^{|\mathcal{C}|} \left[y_i \log P_i'^{\mathrm{co}} + (1 - y_i) \log(1 - P_i'^{\mathrm{co}}) \right] - \lambda \log \det(\mathcal{Q}^\top \mathcal{Q}),$$

where $\mathcal{Q}$ is the assignment matrix. The first term allows the model to account for systematic annotation confusion, while the volume term encourages separated and distinguishable cluster assignments. We

[9] STL10 webpage: https://cs.stanford.edu/~acoates/stl10/

[10] Preprocessed RCV1-10 subset released with [3]: https://github.com/spherepaircc/SpherePairCC

[11] CIFAR10H repository: https://github.com/jcpeterson/cifar-10h

therefore include VolMaxDCC as the compared baseline with explicit noise modeling. At the same time, its design is still rooted in noisy binary supervision and assignment-level separation, whereas UPCC requires preserving a continuum of probabilistic pairwise relations rather than only correcting corrupted hard relations.

CIDEC. CIDEC [14] extends the DEC/IDEC deep embedding clustering framework [49, 53] by incorporating pairwise constraints through the MCL loss. It uses an autoencoder to obtain latent embeddings, initializes C cluster anchors by K-means, and computes a soft assignment vector $\boldsymbol{q}_j = (q_{j1}, \ldots, q_{jC})$ for each instance. Its unsupervised clustering term follows the DEC-style KL objective

$$\mathcal{L}_{\mathrm{DEC}} = \sum_{j=1}^{|\mathcal{X}|} \sum_{c=1}^{C} p_{jc} \log \frac{p_{jc}}{q_{jc}},$$

where $\boldsymbol{p}_j$ is the sharpened target distribution derived from $\boldsymbol{q}_j$. CIDEC combines this clustering loss with a reconstruction loss and the MCL pairwise constraint loss. In our setting, its MCL component uses the real-valued y_i directly through the same BCE-form target as VanillaDCC, making CIDEC a strong end-to-end baseline compatible with probabilistic supervision.

SpherePair. SpherePair [3] is a deep constraint embedding method that learns normalized angular representations with an autoencoder. For binary hard constraints, it defines an angular BCE-form loss

$$\mathcal{L}_{\mathrm{ang}} = -\frac{1}{|\mathcal{C}|} \sum_{i=1}^{|\mathcal{C}|} \Big[y_i \log s^{+}_{a_i b_i} + (1 - y_i) \log(1 - s^{-}_{\mathrm{hard}, a_i b_i}) \Big],$$

where $s^{+}_{a_i b_i}$ is the must-link similarity and $s^{-}_{\mathrm{hard}, a_i b_i}$ is the cannot-link transformed similarity:

$$s^{+}_{a_i b_i} = \tfrac{1}{2}\big(\cos(\theta_{\boldsymbol{z}_{a_i}, \boldsymbol{z}_{b_i}}) + 1\big), \qquad s^{-}_{\mathrm{hard}, a_i b_i} = \tfrac{1}{2}\big(\cos(\min\{\omega \theta_{\boldsymbol{z}_{a_i}, \boldsymbol{z}_{b_i}}, \pi\}) + 1\big).$$

Here ω controls the negative zone of angular size π/ω; the SpherePair theory fixes $\omega = 2$, giving the $\pi/2$ negative-zone boundary and the corresponding conflict-free equidistant geometry for hard must-link/cannot-link constraints. Since this original formulation is defined for binary relation types, applying SpherePair to UPCC requires a continuous counterpart for intermediate labels. We therefore keep the same $\omega = 2$ angular scale and use

$$s^{-}_{\mathrm{soft}, a_i b_i} = \tfrac{1}{2}\big(\cos(2\theta_{\boldsymbol{z}_{a_i}, \boldsymbol{z}_{b_i}}) + 1\big) = \cos^2(\theta_{\boldsymbol{z}_{a_i}, \boldsymbol{z}_{b_i}})$$

in the BCE-form loss with each real-valued $y_i \in [0, 1]$:

$$\mathcal{L}^{\mathrm{soft}}_{\mathrm{ang}} = -\frac{1}{|\mathcal{C}|} \sum_{i=1}^{|\mathcal{C}|} \Big[y_i \log s^{+}_{a_i b_i} + (1 - y_i) \log(1 - s^{-}_{\mathrm{soft}, a_i b_i}) \Big].$$

Under this soft extension, the attractive endpoint $y_i = 1$ is minimized at $\theta_{\boldsymbol{z}_{a_i}, \boldsymbol{z}_{b_i}} = 0$, while the fully repulsive endpoint $y_i = 0$ is minimized at $\theta_{\boldsymbol{z}_{a_i}, \boldsymbol{z}_{b_i}} = \pi/2$, matching the orthogonal negative-zone boundary selected by the original $\omega = 2$ SpherePair geometry. For intermediate labels, the preferred pairwise angle varies continuously between these two endpoints; in the single-pair case, the optimum satisfies $\cos \theta_{\boldsymbol{z}_{a_i}, \boldsymbol{z}_{b_i}} = y_i/(2 - y_i)$. SpherePair also uses the normalized-embedding reconstruction regularizer $\mathcal{L}_{\mathrm{rec}} = \frac{1}{|\mathcal{X}|} \sum_j \|x_j - \hat{x}_j\|_2^2$ with $\hat{x}_j = g(\mathrm{Norm}(f(x_j)))$, which is also adopted by our ProbPair, Weighted ProbPair, and ECI-PP. Thus, the SpherePair baseline in our experiments is the natural real-valued angular extension of the original hard-constraint method, while its formal geometric guarantee remains tied to the binary setting.

ProbPair and Weighted ProbPair. ProbPair is the direct probabilistic pairwise baseline introduced in Section 4.1. It uses the probabilistic angular readout in Eq. (6) and minimizes the reconstruction-regularized objective $\mathcal{L}$ in Eq. (8), without our Estimator–Corrector–Integrator refinement. Weighted ProbPair further applies the information-based constraint weight κ_i in Eq. (9) to the pairwise term:

$$\mathcal{L}_{\mathrm{WPP}} = -\frac{1}{|\mathcal{C}|} \sum_{i=1}^{|\mathcal{C}|} \kappa_i \left[y_i \log \hat{y}_{a_i b_i} + (1 - y_i) \log(1 - \hat{y}_{a_i b_i}) \right] + \lambda_{\mathrm{rec}} \mathcal{L}_{\mathrm{rec}}.$$

Thus, ProbPair isolates the base probabilistic angular formulation, while Weighted ProbPair isolates the effect of decisiveness-based weighting. Comparing Weighted ProbPair with ECI-PP further tests the contribution of the Estimator–Corrector–Integrator pipeline beyond static reweighting.

Table 4: Unfamiliar-class counts per expert under the multi-expert settings. These counts describe the blind-spot distribution used to construct complementary experts.

Dataset type	`multi2`	`multi3`	`multi10`
20-class datasets	$(10, 10)$	$(7, 7, 6)$	$(2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$
10-class datasets	$(5, 5)$	$(4, 3, 3)$	$(1, 1, 1, 1, 1, 1, 1, 1, 1, 1)$
4-class datasets	$(2, 2)$	$(2, 1, 1)$	$(1, 1, 1, 1, 1, 1, 1, 1, 1, 1)$

C.3 Constraint generation

Here, we detail how probabilistic pairwise constraints are generated in our experiments. For simulated-expert experiments, clean probabilistic judgments are generated from trained expert classifiers, after which the Bernoulli–Uniform corruption stage in Section 3.1 is applied to obtain the recorded constraints.

Expert classifiers. For each dataset, simulated experts are trained on controlled labeled subsets of the training split and then used to generate predictive class distributions for queried pairs. The limited labeled-subset construction makes these classifiers expert-dependent judgment mechanisms, whose soft predictive distributions reflect both systematic bias and residual uncertainty. The subset selection rules for single- and multi-expert settings are described below, while the expert network architecture and optimization details are given in Appendix C.5.

Single-expert settings. In the single-expert setting, one expert classifier is trained per dataset. We use three expert-quality levels, denoted by `lv0.1`, `lv0.01`, and `lv0.001`. For `lv`r, the expert is trained using an r fraction of the labeled training samples from each class. The sampling is class-stratified: for every class, the number of selected samples is $\lfloor rn_c \rfloor$, where n_c is the number of available training samples in class c, with at least one sample retained for each class. Thus, smaller values of r produce weaker experts and hence more epistemically distorted probabilistic judgments.

Multi-expert settings. In the multi-expert setting, we construct a group of complementary experts. The three settings `multi2`, `multi3`, and `multi10` use 2, 3, and 10 experts, respectively. Each expert has a set of familiar classes and a set of unfamiliar classes. For all multi-expert settings, familiar classes use a high labeled-data fraction $h = 0.1$, while unfamiliar classes use a low labeled-data fraction $l = 0.0001$. The class-level training fraction for expert e is therefore

$$\rho_{e,c} = \begin{cases} h, & c \in \mathcal{F}_e, \\ l, & c \in \mathcal{U}_e, \end{cases}$$

where $\mathcal{F}_e$ and $\mathcal{U}_e$ denote the familiar and unfamiliar class sets of expert e.

The unfamiliar classes are assigned to make the experts complementary. Let the dataset contain C classes under the fixed label ordering used in expert generation. When the number of experts E satisfies $E \leq C$, the class list is split into E consecutive blocks that are as balanced as possible. Expert e takes the e-th block as its unfamiliar classes and treats all remaining classes as familiar. Equivalently, if $C = qE + r$ with $0 \leq r < E$, then the first r experts have $q + 1$ unfamiliar classes and the remaining experts have q unfamiliar classes. For example, on a 10-class dataset, `multi2` uses unfamiliar-class counts $(5, 5)$, `multi3` uses $(4, 3, 3)$, and `multi10` gives each expert exactly one unfamiliar class. For CIFAR100-20, the corresponding counts are $(10, 10)$, $(7, 7, 6)$, and $(2, \ldots, 2)$. For Reuters subset with four classes, `multi2` gives $(2, 2)$ and `multi3` gives $(2, 1, 1)$. If $E > C$, the experts are divided into full groups of size C, where each expert in a group is unfamiliar with one distinct class; any remaining experts form a partial group whose unfamiliar classes are sampled without replacement from the class list. This case occurs for Reuters subset under `multi10`. The resulting unfamiliar-class counts are summarized in Table 4.

Pair sampling and expert assignment. All queried pairs are sampled randomly and uniformly from the training split. In the single-expert setting, every sampled pair is assigned to the single available expert. In the multi-expert setting, constraints are generated separately for each expert, and the total constraint budget is distributed evenly across experts unless otherwise specified. For a

queried pair (x_a, x_b) assigned to expert e, the trained expert outputs predictive class distributions $\boldsymbol{p}_a^e$ and $\boldsymbol{p}_b^e$. The clean expert judgment is computed as

$$y_{e,ab}^{\mathrm{jud}} = \langle \boldsymbol{p}_a^e, \boldsymbol{p}_b^e \rangle = \sum_{c=1}^{C} p_{a,c}^e p_{b,c}^e \in [0,1].$$

Under independent categorical draws from the expert predictive distributions, this inner product equals the expert-implied probability that the two samples receive the same class. Importantly, this classifier-space inner product is used only to construct experimental supervision and is not the ProbPair readout in Eq. (6), which operates on learned latent embeddings using cosine similarity followed by sigmoid calibration. The resulting judgment is high for similar expert-assigned class distributions, low when the expert separates the samples, and intermediate under uncertainty or overlapping class mass.

Stochastic corruption. After the classifier-derived clean judgment is obtained, the Bernoulli–Uniform corruption stage in Section 3.1 is applied to produce the recorded constraint value. For each pair, an independent corruption indicator is drawn as $c_i \sim \mathrm{Bernoulli}(\pi)$, where π is the corruption probability used by the corresponding experiment. If $c_i = 0$, the recorded label is the clean expert judgment:

$$y_i = y_{e_i,a_i b_i}^{\mathrm{jud}}.$$

If $c_i = 1$, the clean judgment is replaced by an unstructured random value:

$$y_i \sim \mathrm{Uniform}(0,1).$$

Thus, the final constraints combine three factors: data-dependent soft pairwise ambiguity expressed through expert predictive distributions, structured expert-dependent epistemic distortion from class-specific familiarity, and unstructured stochastic corruption from the uniform replacement channel.

Alternative relation and corruption mechanisms. To test whether performance depends on the specific relation construction and corruption mechanism used above, we additionally vary these two components. For the clean relation, we replace the default dot-product judgment with normalized Jensen–Shannon similarity. Given expert predictive distributions $\boldsymbol{p}_a^e$ and $\boldsymbol{p}_b^e$, let

$$\bar{\boldsymbol{p}}_{ab}^e = \frac{1}{2}\left(\boldsymbol{p}_a^e + \boldsymbol{p}_b^e\right),$$

and define

$$y_{e,ab}^{\mathrm{JSD}} = 1 - \frac{\frac{1}{2} D_{\mathrm{KL}}\left(\boldsymbol{p}_a^e \| \bar{\boldsymbol{p}}_{ab}^e\right) + \frac{1}{2} D_{\mathrm{KL}}\left(\boldsymbol{p}_b^e \| \bar{\boldsymbol{p}}_{ab}^e\right)}{\log 2}.$$

For the corruption mechanism, we replace uniform random replacement with complementary flipping: when a constraint is corrupted, its clean judgment y is recorded as $1 - y$. Relative to the default Dot+Uniform generator, where Dot denotes the default inner-product relation and Uniform denotes the default uniform-replacement corruption mechanism, these alternatives yield three stress-test conditions: *Dot+Flip* changes only the corruption mechanism, *JSD+Uniform* changes only the clean relation construction, and *JSD+Flip* changes both.

Human-derived supervision. For CIFAR-10H, we use the aggregate human class-distribution annotation of each image to construct probabilistic pairwise supervision. Let $\boldsymbol{h}_j \in \Delta^9$ denote the normalized human class distribution for image x_j. For a sampled pair (x_a, x_b), we define

$$y_{ab}^{\mathrm{human}} = \langle \boldsymbol{h}_a, \boldsymbol{h}_b \rangle = \sum_{c=1}^{10} h_{a,c} h_{b,c}.$$

Under independent categorical draws from the two human response distributions, this quantity is the induced probability that the pair receives the same class label. Unlike the simulated-expert constraints above, CIFAR-10H supervision is constructed only from the aggregate human judgments, without annotator identities or additional synthetic corruption.

C.4 Experimental protocol

Given the dataset details in Appendix C.1 and the constraint-generation procedure in Appendix C.3, we specify the experimental protocol for comparative evaluations, expert-aware ensemble references, held-out diagnostics, ablation, and sensitivity analysis.

Comparative evaluations. For clustering-performance comparisons, we report *Accuracy* (ACC), *Normalized Mutual Information* (NMI), and *Adjusted Rand Index* (ARI) over five trials. For embedding-based methods, including SpherePair, ProbPair, Weighted ProbPair, and ECI-PP, the final partition is obtained by applying K-means to the learned representations, while end-to-end DCC baselines use their native clustering outputs.

Main comparison. For the full main comparison, we use `lv0.01` and `multi3` with corruption probability 0.3 and total constraint budgets 3k/6k/9k; under `multi3`, these correspond to 1k/2k/3k constraints per expert.

Robustness comparisons. Robustness comparisons use a total budget of 9k constraints and vary one supervision factor at a time. For single-expert robustness, we use `lv0.01` and corruption probability 0.3 as the reference setting: expert quality is varied over `lv0.1`/`lv0.01`/`lv0.001` while fixing corruption probability at 0.3, and corruption probability is varied over $0.1/0.3/0.5$ while fixing the expert level at `lv0.01`. For multi-expert robustness, we compare `multi2`, `multi3`, and `multi10` specified in Appendix C.3; under the 9k total budget, these correspond to $4{,}500$, $3{,}000$, and 900 constraints per expert, respectively. Moving from `multi2` to `multi10` increases both the number of experts and the average expert accuracy (Table 6), because the complementary blind-spot rule assigns fewer unfamiliar classes to each expert (see Table 4).

Human-derived evaluation. For CIFAR-10H, we uniformly sample $3{,}000$ constraints from the 8,000-image training split described in Appendix C.1, using the human-derived relations defined in Appendix C.3, and evaluate on the held-out 2,000 images; pretrain-capable methods are trained from random initialization without autoencoder pretraining. VolMaxDCC follows the validation-based hyperparameter search described in Appendix C.5.

Generator misspecification. We evaluate *Dot+Flip*, *JSD+Uniform*, and *JSD+Flip* as defined in Appendix C.3. Experiments cover all eight main datasets with pretrained initialization for pretrain-capable methods, `multi3`, a total budget of 9k constraints (3k per expert), and corruption probability 0.3, using five trials per condition. All methods otherwise retain their default `multi3` configuration; in particular, VolMaxDCC uses the dataset-specific `multi` value in Table 7, held fixed across the three misspecification conditions.

Expert-aware ensemble references. Under multi-expert supervision, ECI-PP uses expert identities through its expert-specific Correctors, whereas expert-agnostic baselines treat the merged constraints as a single supervision pool. As an additional reference, we also include expert-aware ensemble variants for baselines for multi-expert comparisons. For a baseline trained under an E-expert setting, we instantiate E members, each associated with one expert identity. All members are trained on the full merged constraint set $\mathcal{C} = \{(a_i, b_i, e_i, y_i)\}_{i=1}^{|\mathcal{C}|}$, but member e applies the multiplier

$$\alpha_i^{(e)} = \begin{cases} 1, & e_i = e, \\ \alpha, & e_i \neq e, \end{cases} \qquad \alpha \in [0, 1],$$

to the pairwise loss term of each constraint. Thus, $\alpha = 0$ makes each member use only constraints from its associated expert, while $\alpha = 1$ recovers the expert-agnostic merged-constraint training for every member. More generally, if the pairwise part of a baseline objective is decomposed into per-constraint terms ℓ_i^{pair}, member e uses

$$\mathcal{L}_{\mathcal{B},\text{pair}}^{(e)} = \frac{1}{|\mathcal{C}|} \sum_{i=1}^{|\mathcal{C}|} \alpha_i^{(e)} \ell_i^{\text{pair}},$$

while all non-pairwise components of the baseline objective are kept unchanged. After training, each member produces a hard partition $\widehat{S}^{(e)}$. We aggregate the members through the co-association matrix

$$A_{ab}^{\text{ens}} = \frac{1}{E} \sum_{e=1}^{E} \mathbb{I}\left[\widehat{S}^{(e)}(x_a) = \widehat{S}^{(e)}(x_b)\right],$$

then form $D^{\mathrm{ens}}_{ab} = 1 - A^{\mathrm{ens}}_{ab}$ and apply hierarchical clustering to obtain the final ensemble partition. The non-target expert weight α is selected separately for each dataset–baseline–multi-expert configuration using the reserved 1,000-sample validation split: we search $\alpha \in \{0, 0.01, 0.05, 0.1, 0.5, 1\}$, run each candidate three times, and choose the value with the highest mean validation ACC before test evaluation. These ensemble variants give expert-agnostic baselines access to expert identities, but they also alter the training and aggregation structure through multiple expert-weighted members; they are therefore interpreted as auxiliary expert-aware references rather than fully symmetric replacements for the native baselines.

Held-out diagnostics. The held-out diagnostics examine whether the internal quantities of ECI-PP show trends consistent with their intended roles: Estimator-side beliefs, corrected relations, and integrated relations are compared with an external pairwise reference, and the post-correction gap is tested as a residual-discrepancy signal. To this end, we construct a diagnostic constraint set $\mathcal{C}^{\mathrm{diag}}$ where $|\mathcal{C}^{\mathrm{diag}}| = 1000$ from the test split. These constraints are sample-disjoint from the training constraints and are used only for diagnosis, not for training, early stopping, or hyperparameter selection. For a held-out pair $i = (a_i, b_i)$ with ground-truth class labels t_{a_i} and t_{b_i}, we define the external hard oracle

$$o^\star_i = \mathbb{I}[t_{a_i} = t_{b_i}],$$

which serves only as a diagnostic reference and is **not identified with the canonical aleatoric relation $R^\star_{a_i b_i}$ in Section 3**. On $\mathcal{C}^{\mathrm{diag}}$, which is not assigned to any cross-fitting fold, the Estimator diagnostic belief is computed by averaging the predictions of all K fold-specific Estimators; we still denote this held-out Estimator belief by $\hat{y}^{\mathrm{oof}}_i$ for notational consistency. We then evaluate $\hat{y}^{\mathrm{oof}}_i$, the corrected relation y^{cor}_i, and the Integrator belief y^{int}_i against this external oracle using Brier scores. For any held-out score $v_i \in [0, 1]$ and index set $\mathcal{I}$, we use

$$\mathrm{Brier}(v; \mathcal{I}) = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} (v_i - o^\star_i)^2.$$

The Estimator and Integrator diagnostics are computed on all diagnostic constraints, whereas Corrector-oracle alignment is evaluated on the clean subset $\mathcal{I}_{\mathrm{clean}} = \{i \in \mathcal{C}^{\mathrm{diag}} : c_i = 0\}$. We also compute the clean post-correction discrepancy

$$\mathrm{gap}_{i,\mathrm{clean}} = \left| \mathrm{softclip}_\xi(\hat{y}^{\mathrm{oof}}_i) - y^{\mathrm{cor}}_i \right|, \qquad i \in \mathcal{I}_{\mathrm{clean}},$$

to measure the residual disagreement between the Estimator belief and the corrected relation on uncorrupted held-out judgments. For corruption-screening diagnostics, the corresponding gap score is computed on all diagnostic constraints and used to rank corrupted records ($c_i = 1$) against clean records ($c_i = 0$), from which we report $\mathrm{AUC}_{\mathrm{corrupt}}$ and $\mathrm{AP}_{\mathrm{corrupt}}$. Because this gap captures unexplained discrepancy after correction, it may reflect both stochastic corruption and residual epistemic uncertainty, rather than stochastic corruption alone. We therefore also include an oracle-supervision diagnostic, where the clean relation is given by $o^\star_i$ before applying the same corruption channel; this provides a reference case in which expert-dependent epistemic distortion is removed.

Ablation. We conduct a direct component-level ablation on CIFAR100-20 and Reuters, complementary image and text benchmarks with different class-balance profiles, under the default pretrained `multi3` setting with a total budget of 9k constraints and corruption probability 0.3, using five trials. We compare full ECI-PP with five variants:

- *w/o Cross-fitting*: replaces the out-of-fold Estimator beliefs with in-sample beliefs while leaving the subsequent correction, screening, and integration stages unchanged.
- *w/o Corrector*: disables the Corrector by setting $\hat{\Delta}_i = 0$, so that $y^{\mathrm{cor}}_i = \mathrm{softclip}_\xi(y_i)$, while retaining the remaining screening and integration stages.
- *w/o Screening*: sets $w_i = 1$ while retaining correction and integration.
- *1-pass ECI*: executes one complete Estimator–Corrector–Integrator cycle initialized from (y_i, κ_i) but disables feedback of $(y^{\mathrm{int}}_i, w_i)$ to the Estimators, while matching the total optimization budget of full ECI-PP.
- *Weighted ProbPair*: serves as the reference with the ECI pipeline removed.

Sensitivity analysis. Sensitivity analysis is conducted for ECI-PP under the default single-expert setting, namely `lv0.01` with corruption probability 0.3. Each study is repeated over five trials and varies one factor at a time while keeping the remaining hyperparameters and the constraint-generation protocol fixed. Here, we examine ECI-PP-specific sensitivity factors and implementation choices: the number of estimator folds K and the use of the information-based warm-up score κ_i in Eq. (9); the Corrector regularization weight λ_{cor} and its capacity; the reliability-screening parameters γ, n_0, and soft-clipping sharpness ξ; and the reconstruction weight λ_{rec}.

C.5 Implementation

We provide implementation details[12] for the experimental pipeline, including the computing environment, data preprocessing, expert classifiers and generated constraints, model backbones, pretraining, optimization, and hyperparameter settings.

Software and hardware. All model training code is implemented in Python 3.7 and PyTorch 1.5.1[13]. For methods whose final prediction is obtained by K-means on learned representations, we use the K-means implementation in scikit-learn[14]. For the hierarchical clustering step used by the expert-aware ensemble references in Appendix C.4, we use the `fastcluster` package[15]. Each training run is executed on a single NVIDIA A100 40GB or NVIDIA H200 141GB GPU.

Data preprocessing. We follow the same preprocessing pipeline as [3]. For Reuters subset and RCV1-10, we directly use the preprocessed tf–idf representations over the $2{,}000$ most frequent words or word stems, as described in Appendix C.1. For MNIST and FMNIST, the 28×28 grayscale images are flattened into 784-dimensional vectors, following the standard fully connected input format used in prior deep clustering/DCC studies [13, 14, 2, 3]. For CIFAR10, CIFAR-10H, CIFAR100-20, STL10, and ImageNet10, we adopt the unsupervised feature extraction strategy in [54]: a ResNet-34 model [55] is trained for $1{,}000$ epochs, and the resulting 512-dimensional representations are used as input features. This image preprocessing follows the setting used in DCC studies [15, 3] and keeps the inputs compatible with fully connected architectures across datasets. The same preprocessed inputs are used for expert classifier training and for all compared clustering methods.

Table 5: Test accuracy (%) of single-expert classifiers used for constraint generation.

Dataset	`lv0.1`	`lv0.01`	`lv0.001`
CIFAR100-20	73.08	65.23	46.58
CIFAR10	91.01	88.49	73.83
FMNIST	84.45	74.77	59.83
ImageNet10	96.27	92.62	87.62
MNIST	93.67	88.76	65.20
Reuters	92.90	83.60	49.75
STL10	90.04	84.31	72.77
RCV1-10	94.26	90.47	77.62

Expert classifiers and generated constraints. The expert classifiers used in Appendix C.3 are implemented as MLP classifiers with two hidden layers of size 512–512, ReLU activations, dropout rate 0.2, and a final softmax classification layer. They are trained with cross-entropy loss using Adam, with learning rate 10^{-3}, batch size 256, a maximum of 100 epochs, and an early-stopping patience of 10 based on training accuracy. The resulting test accuracies of single experts are reported in Table 5, and the average test accuracies of multi-expert groups are reported in Table 6. These accuracies characterize the effective quality of the simulated experts produced by the subset-selection and blind-spot protocols in Appendix C.3. After the corruption channel in Section 3.1, these expert judgments yield the recorded constraint values used for training. Fig. 4 visualizes their per-dataset distributions for representative `lv0.01` single-expert and `multi3` multi-expert settings, both with corruption probability 0.3.

[12] **Our source code is available on GitHub:** https://github.com/ecipp/ECI-PP
[13] PyTorch 1.5.1: https://github.com/pytorch/pytorch/releases/tag/v1.5.1
[14] Scikit-learn webpage: https://scikit-learn.org/0.19/documentation.html
[15] fastcluster library: https://pypi.org/project/fastcluster/

Table 6: Average test accuracy (%) of multi-expert groups used for constraint generation. Each value is the arithmetic mean over experts in the corresponding group.

Dataset	`multi2`	`multi3`	`multi10`
CIFAR100-20	41.23	51.90	67.36
CIFAR10	52.72	65.94	82.74
FMNIST	54.57	60.68	77.64
ImageNet10	73.87	76.73	90.80
MNIST	50.45	65.89	86.13
Reuters	47.58	63.02	71.42
STL10	53.98	65.67	83.22
RCV1-10	49.06	64.57	85.96

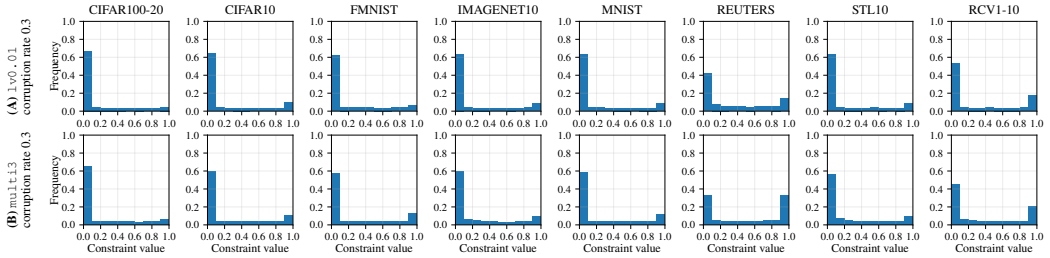


Figure 4: Distributions of recorded constraint values for representative generated-supervision settings. The top row shows `lv0.01` single-expert supervision and the bottom row shows `multi3` multi-expert supervision, both with corruption probability 0.3.

Model backbones and pretraining. For encoder-based methods, we follow the fully connected autoencoder backbone used in [13, 14, 3]: the encoder has hidden layers 500–500–2000, paired with a symmetric decoder when reconstruction is used. The embedding dimension is set to $D = 10$ for all datasets except CIFAR100-20, where $D = 20$, following [3]. VanillaDCC and VolMaxDCC use their native MLP classifier architectures with two hidden layers of size 512–512, and do not use autoencoder pretraining. For pretrain-capable methods, we use the two-stage stacked denoising autoencoder (SDAE) pretraining approach [56, 57], following the protocol used in [13, 14, 3]: hidden layers are first pretrained layer-wise as denoising autoencoders, followed by end-to-end autoencoder fine-tuning on the training split. Unless otherwise specified, all pretrain-capable methods use this pretrained initialization. For methods using angular reconstruction (SpherePair, ProbPair, Weighted ProbPair, ECI-PP), decoding is performed from normalized embeddings as in [3].

In ECI-PP, both the K Estimator backbones and the Integrator use the same autoencoder architecture described above. Each Estimator fold has its own encoder–decoder backbone and its own probabilistic readout parameters (m_k, T_k), while the Integrator has its own readout parameters (m, T) in Eq. (6). These scalar readout parameters are parameterized as $m = \tanh(\tilde{m})$ and $T = \mathrm{softplus}(\tilde{T})$ for numerical stability within the cosine range, with the same parameterization used for each Estimator fold. When pretraining is used, the same pretrained autoencoder weights initialize the Integrator backbone and all Estimator backbones, while the readout parameters and Correctors are initialized separately. Unless otherwise specified, each expert-specific Corrector is an MLP with hidden layers 64–16 and a $\tanh$ output, which keeps the predicted probability-space residual within $(-1, 1)$.

Optimization and hyperparameters. We use the reported or recommended hyperparameters for the baseline methods whenever available. Below, we summarize the optimization settings and implementation configurations for ECI-PP and all compared baselines.

- **VanillaDCC.** VanillaDCC optimizes the MCL loss $\mathcal{L}_{\mathrm{MCL}}$ using Adam with learning rate 10^{-3} and batch size 256. Training runs for at most 500 epochs, with early stopping triggered when the relative change in the soft cluster assignments on the training samples falls below 10^{-3} for two consecutive checks. This assignment-change stopping rule follows common practice in deep clustering and deep constrained clustering [49, 53, 13, 14].

Table 7: Selected VolMaxDCC geometric regularization weights for the eight main benchmark datasets under simulated-expert supervision. Values are selected by validation ACC over three runs for each dataset and supervision setting.

Dataset	lv0.1	lv0.01	lv0.001	multi
CIFAR100-20	10^{-3}	10^{-3}	10^{-1}	10^{-2}
CIFAR10	10^{-3}	10^{-3}	10^{-1}	10^{-2}
FMNIST	10^{-3}	10^{-3}	10^{-2}	10^{-2}
ImageNet10	0	10^{-4}	10^{-1}	10^{-2}
MNIST	10^{-2}	10^{-3}	10^{-2}	10^{-2}
Reuters	10^{-2}	0	10^{-1}	10^{-1}
STL10	10^{-3}	10^{-3}	10^{-2}	10^{-1}
RCV1-10	10^{-4}	10^{-4}	10^{-5}	10^{-3}

- **VolMaxDCC.** For VolMaxDCC, we follow its original optimization and validation-based hyperparameter-search procedure [15]. The learnable matrix B is parameterized elementwise as $B_{cc'} = (1+\exp(-B'_{cc'}))^{-1}$, where $B'_{cc'}$ is initialized to 1 for $c = c'$ and to -1 otherwise. Optimization uses SGD with learning rate 0.5 for the network parameters and 0.1 for B', with batch size 128. The VolMaxDCC geometric regularization weight is selected from $\{0, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$ using the reserved validation split; for each dataset and supervision setting, we choose the value with the highest mean validation ACC over three validation runs. For multi-expert supervision, a shared selected value is used for the multi-expert settings defined in Appendix C.3, where the familiar and unfamiliar labeled-data fractions are fixed as $h = 0.1$ and $l = 0.0001$. The selected values for the eight main benchmark datasets under simulated-expert supervision are reported in Table 7; for the CIFAR-10H human-derived evaluation, the same search selects $\lambda = 10^{-2}$.
- **CIDEC.** For CIDEC, we follow the authors' recommended settings: the reconstruction/clustering trade-off is set to $\lambda_1 = 1$, and the MCL constraint-balancing parameter is set to $\lambda_2 = 0.1$. The C cluster anchors are initialized by K-means. Optimization uses Adam with learning rate 10^{-3} and batch size 256. Training proceeds for at most 500 epochs, with early stopping applied when the relative change in the soft assignments falls below 10^{-3} over consecutive checks.
- **SpherePair.** For SpherePair, the reconstruction weight is set to 0.02, matching the setting used in [3]. In the original binary formulation, the negative-zone factor is fixed as $\omega = 2$, which yields the theoretically guaranteed $\pi/2$ negative-zone boundary. As described in Appendix C.2, we use the real-valued extension of SpherePair by keeping the same $\omega = 2$ angular scale and replacing the original clamped cannot-link score with the continuous counterpart $s^{-}_{\mathrm{soft},a_i b_i} = \frac{1}{2}\big(\cos(2\theta_{\boldsymbol{z}_{a_i},\boldsymbol{z}_{b_i}})+1\big)$. The observed $y_i \in [0,1]$ is then used directly in the BCE-form angular loss. Under this implementation, $y_i = 1$ favors $\theta_{\boldsymbol{z}_{a_i},\boldsymbol{z}_{b_i}} = 0$ and $y_i = 0$ favors $\theta_{\boldsymbol{z}_{a_i},\boldsymbol{z}_{b_i}} = \pi/2$, while intermediate labels continuously interpolate between these two angular endpoints through the two BCE terms. Optimization uses Adam with learning rate 10^{-3} and constraint mini-batch size 256. The instance mini-batch size for reconstruction is set according to the number of constraint mini-batches, following [3]. Training runs for at most 500 epochs, with early stopping after the first 100 epochs if the relative change in the total loss remains below 0.1 for five consecutive checks. For the expert-aware ensemble reference in Appendix C.4, the expert-aware hyperparameter α is selected on the reserved validation split; the selected values are reported in Table 8.
- **ProbPair variants and ECI-PP (Full).** ProbPair, Weighted ProbPair, and ECI-PP use the ProbPair-style readout in Eq. (6) and the reconstruction weight $\lambda_{\mathrm{rec}} = 0.02$. ProbPair optimizes Eq. (8), while Weighted ProbPair additionally applies the information-based weight κ_i in Eq. (9). For the expert-aware ensemble reference based on Weighted ProbPair in Appendix C.4, the expert-aware hyperparameter α is selected on the reserved validation split; the selected values are reported in Table 8. The shared base optimizer is Adam with learning rate 10^{-3} and batch size 256. The probabilistic readout parameters, including (m, T) and the fold-wise (m_k, T_k) when present, are initialized as $(0, 0.1)$ and optimized with learning rate 10^{-2}. We use a fixed outer training budget of 500 epochs; principled

Table 8: Selected expert-aware ensemble hyperparameter α. Values are chosen by mean validation ACC over three runs and shared across the multi-expert configurations defined in Appendix C.3.

Dataset	SpherePair	Weighted ProbPair
CIFAR100-20	0.05	0.05
CIFAR10	1.00	0.05
FMNIST	1.00	0.50
ImageNet10	0.00	0.10
MNIST	0.05	0.50
Reuters	0.05	0.05
STL10	0.01	0.01
RCV1-10	0.01	0.05

stopping criteria for iterative refinement remain an open question for future study. For any weighted ProbPair-style objective, including Weighted ProbPair and the weighted Estimator/Integrator updates in ECI-PP, the weight is applied inside the constraint average:

$$\mathcal{L}_{\mathrm{PP}}^{w} = -\frac{1}{|\mathcal{C}|}\sum_{i=1}^{|\mathcal{C}|} w_i \Big[\tilde{y}_i \log \hat{y}_{a_i b_i} + (1-\tilde{y}_i)\log(1-\hat{y}_{a_i b_i})\Big],$$

where $\tilde{y}_i$ denotes the corresponding training target, such as y_i, y_i^{int}, or y_i^{BC}.

For ECI-PP, we use one global hyperparameter setting unless explicitly varied in sensitivity analysis: $K = 5$ estimator folds, $\lambda_{\mathrm{cor}} = 0.5$, $\gamma = 10$, $n_0 = 10$, soft-clipping sharpness $\xi = 20$, and information-based warm-up enabled. The default warm-up trains the Estimators and Integrator for 50 epochs before iterative refinement. The initial $\hat{y}_i^{\mathrm{oof}}$ is obtained by strict fold-heldout prediction, and later refinement rounds use refreshed Estimator beliefs under the Integrator-updated targets while retaining the same fold structure. Each Corrector update uses the shared base optimizer setting, with the Huber quadratic-to-linear transition in Eq. (10) fixed at 0.1, holds out 10% of its assigned constraints for internal validation, runs for at most 50 epochs, and stops early if the validation loss does not improve for 10 consecutive epochs. Between refinement iterations, Gaussian noise with standard deviation 0.01 is added to the final linear layers of the Estimators, Correctors, and Integrator to mildly perturb the next refinement step.

Notably, the validation-based selections of the VolMaxDCC geometric regularization weight and the expert-aware ensemble hyperparameter α rely on ground-truth class labels on the validation split, which are **typically unavailable** in constrained clustering and are not part of the observable supervision under UPCC. In contrast, ECI-PP uses **one global hyperparameter setting** across datasets and handles expert identities directly through expert-specific Correctors, without dataset-specific tuning or an additional hyperparameterized ensemble wrapper.

D Additional experimental results

D.1 Comparison under different constraint budgets

Tables 9 and 10 report the full main-comparison results under the default `lv0.01` single-expert and `multi3` multi-expert regimes, with corruption probability fixed at 0.3. The total constraint budget is varied over 3k/6k/9k; under `multi3`, these correspond to 1k/2k/3k constraints per expert. All results in this subsection use pretrained backbones for methods with a pretraining stage, while VanillaDCC and VolMaxDCC have no pretraining variant.

The main-text conclusion remains stable across budgets. Across the 144 test entries in the two tables, ECI-PP ranks first in $122/144$ entries and within the top two in $134/144$, while one ProbPair-family method is best in $137/144$ entries. The budget-wise pattern is not completely uniform, as expected: with only 3k constraints, all methods are less stable and the separation between methods is smaller, whereas at 9k several strong baselines also improve and narrow some ACC gaps. Even under these effects, ECI-PP remains the leading method at every budget, ranking first in $38/48$, $43/48$, and $41/48$ entries under 3k, 6k, and 9k constraints, respectively.

Table 9: Comparative performance (%) (ACC, NMI, ARI) across datasets for methods under the `1v0.01` setting, with corruption probability 0.3 and 3k/6k/9k constraints. Blue and black represent training and test results, respectively. Best results are in **bold**, and second-best are underlined.

			External baselines				Our progressive models		
			Vanilla-DCC	VolMax-DCC	CIDEC	SpherePair	ProbPair	Weighted ProbPair	**ECI-PP** (Full)
CIFAR100-20	3k	ACC	14.7, 14.7	42.8, 42.5	14.9, 14.6	45.7, 45.7	44.5, 44.4	45.2, 45.2	**48.1**, **47.8**
		NMI	11.2, 11.8	42.5, 42.9	13.1, 13.6	42.8, 43.3	42.4, 42.9	43.4, 43.7	**46.0**, **46.5**
		ARI	4.0, 4.1	26.2, 26.0	2.6, 2.1	28.5, 28.5	27.6, 27.6	28.6, 28.5	**31.3**, **31.4**
	6k	ACC	16.5, 16.6	45.4, 45.3	16.6, 15.6	49.0, 49.2	46.1, 46.4	48.3, 48.5	**51.2**, **51.1**
		NMI	12.6, 13.5	43.9, 44.5	15.0, 15.1	43.8, 44.9	42.4, 43.4	43.7, 44.5	**48.3**, **48.8**
		ARI	5.4, 5.6	28.2, 28.2	3.4, 2.9	31.5, 32.0	28.7, 29.0	30.2, 30.5	**34.7**, **34.7**
	9k	ACC	16.8, 17.0	46.1, 45.9	14.3, 13.6	50.2, 50.5	49.4, 49.5	50.3, 50.8	**52.7**, **52.9**
		NMI	13.5, 14.6	45.6, 46.1	11.0, 11.2	43.0, 44.2	43.2, 44.2	44.5, 45.7	**49.2**, **49.9**
		ARI	6.0, 6.3	30.4, 30.5	2.8, 2.3	32.1, 32.6	30.6, 30.9	31.5, 32.1	**36.2**, **36.5**
CIFAR10	3k	ACC	34.6, 34.6	79.9, 79.8	54.1, 52.7	84.6, 84.9	84.8, 85.0	**85.8**, **85.7**	85.5, 85.4
		NMI	27.9, 28.1	69.3, 69.4	40.8, 40.7	74.3, 74.7	74.9, 75.3	76.0, 76.0	**77.0**, **76.9**
		ARI	20.2, 20.3	64.4, 64.4	23.0, 20.5	71.7, 72.0	71.9, 72.1	73.4, 73.2	**73.5**, **73.3**
	6k	ACC	35.5, 35.7	84.7, 84.6	51.3, 51.0	84.8, 85.3	86.0, 86.0	86.5, 86.5	**87.3**, **87.2**
		NMI	27.5, 28.3	73.6, 73.7	39.8, 41.1	73.1, 74.2	75.6, 75.9	76.3, 76.5	**78.6**, **78.7**
		ARI	20.7, 21.2	71.0, 70.9	22.9, 22.8	71.5, 72.3	73.5, 73.5	74.3, 74.2	**76.0**, **76.0**
	9k	ACC	38.3, 38.7	82.5, 82.5	44.3, 44.0	84.9, 85.7	85.9, 86.0	86.6, 86.7	**87.8**, **87.8**
		NMI	29.9, 31.2	71.9, 72.2	36.1, 37.6	72.4, 74.1	75.0, 75.4	76.2, 76.8	**79.0**, **79.1**
		ARI	23.2, 24.0	68.3, 68.3	21.1, 21.3	71.2, 72.5	73.0, 73.2	74.3, 74.5	**76.7**, **76.8**
FMNIST	3k	ACC	36.6, 36.5	63.3, 62.5	35.4, 34.4	**71.1**, **70.0**	70.5, 69.5	71.0, **70.0**	70.2, 69.2
		NMI	31.1, 31.1	55.5, 55.2	31.6, 31.3	61.0, 60.7	61.0, 60.6	61.3, 61.0	**64.0**, **63.4**
		ARI	19.7, 19.5	44.4, 43.6	13.5, 12.4	53.2, 52.0	52.1, 51.0	53.1, 52.1	**53.8**, **52.6**
	6k	ACC	38.6, 38.5	66.2, 65.1	33.8, 33.3	70.9, 69.8	72.7, 71.5	**73.6**, **72.5**	71.3, 70.2
		NMI	30.9, 31.3	56.7, 56.3	26.9, 27.1	60.6, 60.5	62.4, 62.1	63.1, 62.9	**65.3**, **64.5**
		ARI	20.7, 20.9	46.6, 45.5	12.2, 11.9	53.2, 52.1	53.9, 52.6	54.6, 53.5	**55.1**, **53.8**
	9k	ACC	39.8, 40.2	66.5, 65.5	36.1, 35.7	72.0, 71.2	73.6, 72.5	**74.6**, **73.6**	72.4, 71.1
		NMI	30.1, 31.3	57.2, 56.8	29.7, 30.5	60.4, 60.9	62.9, 62.9	63.9, 64.0	**65.7**, **65.0**
		ARI	21.5, 22.2	46.4, 45.2	15.7, 15.6	53.0, 52.3	54.8, 53.7	55.3, **54.3**	**55.5**, 53.9
ImageNet10	3k	ACC	35.4, 37.3	86.7, 86.9	69.0, 71.2	88.0, 89.2	89.6, 89.8	90.0, 90.1	**92.2**, **92.5**
		NMI	27.7, 30.8	78.8, 79.4	53.9, 58.7	78.6, 81.4	81.8, 82.3	82.8, 83.3	**87.7**, **88.1**
		ARI	19.1, 20.8	74.1, 73.9	43.4, 47.1	76.8, 78.9	79.4, 79.6	80.5, 80.5	**85.0**, **85.4**
	6k	ACC	40.6, 42.0	87.2, 87.5	67.0, 71.2	87.2, 89.3	89.7, 90.5	89.7, 90.2	**92.7**, **92.9**
		NMI	30.6, 35.2	78.3, 79.3	50.3, 58.7	75.4, 80.4	81.2, 83.3	81.6, 82.8	**88.3**, **88.5**
		ARI	23.2, 26.0	75.1, 75.6	40.2, 46.3	74.7, 78.7	79.3, 80.7	79.5, 80.0	**85.9**, **86.1**
	9k	ACC	39.0, 41.5	87.4, 88.1	63.5, 68.3	85.1, 89.1	89.2, 90.5	89.1, 90.3	**92.8**, **93.1**
		NMI	27.6, 33.7	77.8, 79.2	47.2, 57.5	71.0, 79.2	80.1, 83.2	80.3, 82.7	**88.1**, **88.5**
		ARI	21.1, 25.2	75.2, 76.1	35.5, 42.2	70.7, 78.2	78.5, 80.7	78.4, 80.2	**85.8**, **86.2**
MNIST	3k	ACC	25.9, 26.1	65.6, 66.4	59.1, 56.5	87.3, 87.9	86.3, 86.6	**89.6**, **90.2**	84.2, 84.6
		NMI	16.0, 17.0	52.7, 54.1	51.9, 51.0	78.1, 79.3	75.5, 76.5	**78.4**, **79.7**	75.0, 76.1
		ARI	10.4, 10.8	47.8, 48.9	30.3, 26.7	76.8, 77.8	74.3, 74.9	**78.5**, **79.7**	72.3, 73.1
	6k	ACC	33.3, 33.9	73.4, 74.4	59.2, 57.0	89.3, 90.1	88.0, 89.2	**89.5**, 90.1	**89.5**, **90.3**
		NMI	24.2, 25.8	58.4, 60.5	52.0, 52.1	78.2, 80.2	75.8, 78.1	77.8, 79.2	**79.2**, **80.3**
		ARI	18.4, 19.4	55.2, 56.8	29.8, 26.4	78.2, 79.9	75.8, 78.1	78.4, 79.7	**78.9**, **80.1**
	9k	ACC	38.9, 40.3	75.6, 76.9	47.0, 46.7	87.8, 89.1	89.0, 90.0	89.9, 90.8	**90.7**, **91.0**
		NMI	29.5, 32.2	59.3, 61.6	41.4, 43.7	75.4, 78.0	76.9, 79.0	78.3, 80.2	**80.7**, **81.6**
		ARI	23.1, 24.8	56.5, 58.7	24.7, 24.2	75.4, 77.8	77.4, 79.4	79.1, 80.9	**80.8**, **81.5**
REUTERS	3k	ACC	53.2, 54.0	62.8, 63.2	74.4, 77.0	81.4, 83.4	79.6, 81.3	82.8, 84.4	**86.0**, **86.8**
		NMI	15.1, 17.9	25.6, 28.0	42.8, 49.8	54.5, 58.9	55.4, 58.5	57.8, 61.6	**63.8**, **65.6**
		ARI	17.0, 18.0	31.2, 32.4	50.4, 56.7	59.8, 64.4	58.9, 62.3	62.9, 66.1	**69.4**, **71.0**
	6k	ACC	66.0, 70.6	72.5, 76.8	74.2, 79.3	80.1, 83.2	79.6, 81.5	82.2, 83.9	**85.4**, **86.4**
		NMI	27.7, 36.3	36.0, 44.9	38.2, 48.8	52.0, 58.5	55.0, 58.9	56.7, 60.4	**62.4**, **65.0**
		ARI	34.0, 42.6	42.4, 50.8	44.6, 55.6	56.8, 63.2	58.1, 62.0	61.5, 65.3	**67.8**, **70.3**
	9k	ACC	69.7, 77.0	71.7, 76.8	72.9, 79.3	77.9, 81.8	78.2, 80.8	80.4, 82.7	**85.1**, **86.6**
		NMI	30.5, 43.3	34.3, 43.4	35.7, 48.1	48.7, 56.3	53.2, 58.3	54.5, 59.2	**61.4**, **64.4**
		ARI	36.8, 50.4	40.9, 50.4	41.8, 54.9	52.9, 61.1	55.5, 61.0	58.6, 63.4	**66.6**, **70.0**
STL10	3k	ACC	30.9, 31.5	72.8, 73.3	66.8, 69.4	80.7, 81.7	81.9, 83.0	83.3, 83.9	**85.0**, **85.3**
		NMI	21.1, 23.4	64.8, 66.5	49.9, 55.1	66.7, 69.1	69.7, 72.0	71.5, 73.0	**75.3**, **75.9**
		ARI	14.1, 15.0	56.3, 57.0	40.5, 44.8	64.2, 66.1	66.3, 68.2	68.8, 69.8	**72.0**, **72.5**
	6k	ACC	30.4, 31.6	76.8, 77.5	62.7, 66.7	79.0, 81.7	81.6, 82.9	82.8, 83.9	**85.4**, **85.5**
		NMI	19.5, 22.2	65.8, 67.7	44.4, 51.4	63.2, 68.2	68.7, 71.6	70.2, 72.5	**75.8**, **76.3**
		ARI	14.3, 15.7	60.0, 61.3	34.1, 40.0	61.2, 65.5	65.9, 68.2	67.9, 69.6	**72.7**, **73.0**
	9k	ACC	35.5, 37.8	79.0, 80.1	57.8, 63.3	76.1, 79.7	81.6, 83.4	81.6, 83.1	**85.4**, **85.8**
		NMI	22.9, 27.6	66.7, 69.3	39.3, 48.5	59.5, 66.5	68.2, 72.1	67.9, 71.6	**75.8**, **76.6**
		ARI	17.1, 19.8	62.8, 64.7	29.8, 37.3	57.2, 63.3	65.8, 68.9	65.8, 68.4	**72.8**, **73.4**
RCV1-10	3k	ACC	40.0, 40.1	46.3, 46.4	47.4, 48.7	**65.1**, **64.9**	54.0, 53.8	52.6, 52.4	61.7, 61.7
		NMI	12.3, 12.4	21.5, 21.9	33.3, 32.5	59.2, 59.6	54.2, 54.4	55.1, 55.3	**60.6**, **60.8**
		ARI	16.4, 16.5	21.2, 21.4	28.6, 29.1	**56.1**, **56.2**	43.3, 43.1	43.5, 43.4	53.3, 53.5
	6k	ACC	52.4, 52.6	62.1, 62.4	42.3, 43.5	**70.8**, **70.7**	51.5, 51.3	54.0, 53.9	63.7, 63.7
		NMI	21.8, 22.2	36.0, 36.7	30.5, 30.5	60.6, 61.1	54.1, 54.2	57.5, 57.8	**62.9**, **63.0**
		ARI	28.0, 28.2	42.9, 43.3	24.9, 26.7	**61.6**, **61.8**	42.3, 42.1	46.2, 46.1	55.9, 56.1
	9k	ACC	46.2, 46.4	61.3, 61.8	42.5, 44.3	**71.2**, **71.4**	52.7, 53.0	55.2, 55.1	69.0, 68.9
		NMI	15.5, 15.9	35.7, 36.7	29.5, 30.0	60.5, 61.3	53.8, 54.1	57.4, 57.5	**65.5**, **65.8**
		ARI	19.8, 20.1	44.7, 45.4	26.2, 28.3	61.0, 61.4	43.6, 43.7	47.1, 47.0	**62.0**, **62.0**

Table 10: Comparative performance (%) (ACC, NMI, ARI) across datasets for methods under the `multi3` setting, with corruption probability 0.3 and 3k/6k/9k total constraints. Blue and black represent training and test results, respectively. Best results are in **bold**, and second-best are underlined.

			External baselines				Our progressive models		
			Vanilla-DCC	VolMax-DCC	CIDEC	SpherePair	ProbPair	Weighted ProbPair	**ECI-PP** (Full)
CIFAR100-20	3k	ACC	12.2, 12.4	43.7, 43.3	19.5, 18.4	42.0, 42.1	41.5, 41.6	43.1, 43.3	**45.6**, **45.3**
		NMI	9.5, 9.9	43.4, 43.6	20.5, 20.5	41.5, 42.1	40.2, 41.1	41.9, 42.4	**45.0**, **45.5**
		ARI	2.6, 2.5	26.8, 26.6	3.9, 3.2	26.1, 26.3	24.8, 25.3	26.7, 26.8	**29.7**, **29.8**
	6k	ACC	14.2, 14.4	43.1, 42.5	17.4, 16.6	42.4, 42.4	42.0, 42.1	44.1, 44.4	**49.1**, **49.3**
		NMI	11.3, 12.2	44.3, 44.5	16.2, 16.3	40.0, 40.9	39.8, 40.6	41.1, 41.9	**47.1**, **47.7**
		ARI	4.6, 4.7	27.7, 27.5	3.2, 2.5	26.2, 26.5	25.2, 25.4	26.4, 26.7	**32.1**, **32.4**
	9k	ACC	15.4, 15.8	42.8, 42.5	12.8, 12.1	44.3, 44.4	43.3, 43.4	46.1, 46.5	**48.6**, **48.8**
		NMI	12.8, 13.6	43.4, 43.8	10.0, 10.5	40.5, 41.5	40.0, 40.9	41.8, 43.0	**48.1**, **48.5**
		ARI	5.8, 5.8	27.1, 27.1	2.1, 2.0	27.1, 27.6	26.0, 26.2	28.0, 28.5	**32.9**, **32.9**
CIFAR10	3k	ACC	28.8, 28.8	76.0, 75.8	38.4, 36.8	74.8, 75.1	79.8, 79.8	82.0, 82.1	**82.8**, **82.5**
		NMI	25.2, 25.5	71.0, 70.9	32.3, 31.7	69.5, 70.2	72.0, 72.1	73.2, 73.3	**76.0**, **75.9**
		ARI	16.5, 16.6	63.3, 63.0	13.4, 11.9	63.0, 63.5	67.0, 66.8	68.7, 68.7	**70.8**, **70.7**
	6k	ACC	33.6, 33.9	78.0, 78.1	36.5, 34.9	76.5, 76.8	80.7, 81.0	80.9, 81.0	**84.0**, **84.0**
		NMI	30.9, 32.0	72.6, 72.9	31.9, 31.9	66.9, 68.1	71.2, 71.6	71.9, 72.2	**77.0**, **77.1**
		ARI	22.4, 23.1	65.7, 65.8	14.2, 14.0	61.7, 62.5	66.2, 66.6	67.8, 67.9	**72.6**, **72.6**
	9k	ACC	32.4, 32.8	76.5, 76.6	33.2, 31.6	73.8, 74.5	81.3, 81.7	81.4, 81.7	**81.6**, **82.1**
		NMI	29.1, 30.9	73.1, 73.2	28.7, 29.1	64.7, 66.5	71.5, 72.3	72.0, 72.6	**77.8**, **77.8**
		ARI	21.2, 22.4	65.8, 65.9	12.3, 12.1	59.0, 60.5	67.1, 67.8	67.9, 68.4	**73.1**, **73.0**
FMNIST	3k	ACC	33.7, 33.7	58.1, 57.7	30.1, 29.6	61.3, 61.0	60.9, 60.3	**62.6**, **62.5**	60.1, 59.2
		NMI	32.9, 33.0	56.7, 56.4	29.8, 30.1	58.1, 58.2	59.2, 58.8	59.1, 59.1	**62.7**, **61.9**
		ARI	20.7, 20.6	42.7, 42.1	14.1, 14.0	45.9, 45.6	46.4, 45.8	46.4, 46.2	**49.5**, **48.4**
	6k	ACC	34.2, 34.3	58.2, 57.9	30.4, 30.1	65.4, 65.3	64.1, 63.8	**66.1**, **66.0**	62.0, 61.1
		NMI	32.1, 32.8	58.3, 58.1	28.1, 29.6	58.9, 59.7	59.6, 59.7	60.1, 60.3	**64.3**, **63.8**
		ARI	21.0, 21.2	44.4, 43.7	14.2, 14.7	48.7, 48.9	47.8, 47.7	49.6, 49.5	**51.0**, **51.0**
	9k	ACC	35.2, 35.6	59.4, 59.1	28.6, 28.8	61.0, 61.3	66.6, 66.7	**68.0**, **67.6**	65.3, 64.8
		NMI	31.4, 32.8	59.2, 59.0	28.3, 30.6	56.6, 57.8	60.7, 61.2	61.7, 62.0	**66.0**, **65.6**
		ARI	20.4, 21.3	45.0, 44.3	14.5, 15.4	46.8, 47.4	49.9, 50.0	51.9, 51.8	**53.4**, **52.8**
ImageNet10	3k	ACC	33.9, 34.7	88.8, 88.9	50.7, 49.2	81.0, 81.5	87.8, 88.4	89.5, 89.7	**91.5**, **91.8**
		NMI	29.0, 31.4	85.9, 85.8	41.2, 43.3	72.7, 74.2	78.8, 80.3	81.6, 82.3	**86.9**, **87.3**
		ARI	19.4, 20.7	81.9, 81.7	21.7, 20.4	68.2, 69.1	76.7, 77.7	79.7, 79.9	**84.1**, **84.4**
	6k	ACC	32.7, 34.6	87.6, 88.1	43.0, 41.6	81.8, 83.7	88.6, 89.4	89.6, 90.1	**91.9**, **92.1**
		NMI	25.3, 30.1	86.5, 87.1	35.4, 38.5	69.2, 73.5	78.3, 80.4	80.2, 81.6	**87.2**, **87.8**
		ARI	17.9, 20.9	81.9, 82.4	17.3, 17.9	66.3, 70.0	77.2, 78.6	79.3, 80.1	**84.6**, **85.0**
	9k	ACC	31.8, 34.2	89.9, 90.5	39.4, 38.8	80.7, 85.0	87.4, 89.2	90.1, 91.3	**92.4**, **92.6**
		NMI	22.0, 26.8	**87.8**, **88.7**	31.0, 34.8	65.6, 73.8	76.1, 80.1	80.1, 83.0	87.7, 88.1
		ARI	15.8, 18.7	83.9, 84.9	14.3, 15.6	63.4, 71.0	75.0, 78.3	79.8, 82.0	**85.4**, **85.7**
MNIST	3k	ACC	27.0, 27.3	54.6, 55.4	44.9, 41.8	83.6, 84.4	81.2, 82.2	**86.5**, **87.2**	80.9, 81.5
		NMI	19.2, 20.5	48.5, 49.8	45.6, 44.1	75.2, 76.9	73.0, 74.7	75.2, 76.5	**75.7**, **77.1**
		ARI	11.5, 12.2	38.6, 39.5	20.4, 17.8	71.5, 72.4	69.4, 70.9	**73.9**, **74.9**	71.9, 73.2
	6k	ACC	28.1, 29.0	62.5, 63.3	39.8, 38.1	78.8, 79.9	81.0, 82.3	**86.5**, **87.4**	86.1, 86.4
		NMI	22.0, 23.8	55.1, 56.6	39.2, 39.3	72.8, 75.3	72.2, 74.4	75.3, 76.9	**79.8**, **80.5**
		ARI	14.7, 15.7	47.1, 48.4	16.8, 15.4	68.2, 70.3	68.7, 70.8	73.9, 75.4	**77.8**, **78.3**
	9k	ACC	32.2, 33.3	67.1, 67.8	31.8, 31.0	78.6, 80.0	80.5, 81.8	**85.1**, **86.1**	84.7, 85.1
		NMI	26.5, 29.3	57.5, 58.9	28.8, 30.2	70.8, 74.0	72.1, 74.7	75.0, 77.2	**79.9**, **80.7**
		ARI	18.4, 20.1	50.9, 52.1	13.2, 13.4	66.3, 69.1	68.7, 71.2	73.4, 75.5	**77.1**, **77.5**
REUTERS	3k	ACC	42.8, 41.4	66.9, 67.3	61.4, 62.1	72.8, 75.2	78.6, 79.9	75.7, 78.0	**85.0**, **86.3**
		NMI	3.2, 3.7	40.8, 42.5	26.3, 32.2	40.2, 44.6	48.0, 50.3	49.8, 53.4	**62.0**, **65.0**
		ARI	4.7, 4.0	42.8, 43.9	28.1, 28.9	45.2, 50.1	55.0, 57.5	53.8, 58.1	**68.3**, **71.1**
	6k	ACC	51.4, 52.9	67.0, 68.2	58.7, 60.4	66.7, 72.9	78.6, 81.4	81.0, 83.4	**87.3**, **88.5**
		NMI	11.8, 15.9	44.3, 46.5	19.8, 27.0	32.8, 43.2	48.3, 54.0	52.9, 57.7	**63.8**, **67.2**
		ARI	14.0, 16.4	46.5, 48.7	22.6, 25.1	37.2, 48.2	54.5, 60.3	58.8, 63.6	**71.3**, **74.3**
	9k	ACC	55.3, 59.2	67.5, 68.0	57.3, 59.1	62.1, 69.9	74.5, 79.0	81.0, 83.9	**88.3**, **89.6**
		NMI	14.7, 22.8	45.4, 46.5	18.3, 25.7	25.6, 37.2	44.6, 52.8	51.5, 57.9	**65.4**, **68.7**
		ARI	19.2, 26.2	47.0, 47.9	20.4, 23.1	28.0, 40.7	50.2, 58.6	57.9, 64.2	**72.9**, **76.0**
STL10	3k	ACC	25.6, 26.4	74.6, 74.8	47.4, 47.4	72.7, 73.4	79.2, 79.9	80.4, 80.8	**85.7**, **85.9**
		NMI	19.7, 21.7	71.5, 72.0	38.3, 41.5	61.7, 64.0	66.5, 68.4	68.9, 70.0	**75.7**, **76.3**
		ARI	12.3, 13.1	62.3, 62.8	21.8, 22.4	55.0, 56.4	61.6, 62.9	64.7, 65.3	**72.2**, **72.5**
	6k	ACC	29.8, 31.2	72.7, 72.6	37.6, 37.2	67.7, 69.8	81.0, 82.0	83.1, 84.2	**87.4**, **87.5**
		NMI	22.3, 26.4	71.1, 71.2	31.2, 37.2	54.8, 58.8	67.4, 69.9	69.5, 72.0	**77.5**, **78.1**
		ARI	15.5, 17.7	61.1, 60.6	15.8, 18.2	48.6, 51.7	64.0, 65.7	67.4, 69.4	**75.1**, **75.4**
	9k	ACC	29.2, 30.7	71.0, 71.1	31.4, 31.6	67.8, 71.4	81.1, 83.0	83.1, 84.3	**87.9**, **87.7**
		NMI	21.5, 26.0	70.2, 69.9	24.9, 31.7	53.2, 59.7	66.1, 70.2	69.4, 72.0	**78.3**, **78.5**
		ARI	15.2, 17.7	59.5, 59.1	13.6, 16.9	47.8, 53.3	63.7, 67.1	67.4, 69.4	**75.9**, **75.8**
RCV1-10	3k	ACC	29.2, 29.2	31.7, 31.9	33.5, 34.9	51.1, 51.0	47.9, 47.8	47.0, 46.9	**55.3**, **55.2**
		NMI	2.5, 2.6	23.0, 23.6	19.1, 18.3	51.2, 51.7	50.0, 50.3	51.4, 51.7	**56.8**, **56.9**
		ARI	3.2, 3.1	10.8, 11.1	14.3, 14.6	39.5, 39.6	37.8, 37.9	37.3, 37.4	**45.6**, **45.4**
	6k	ACC	43.2, 43.4	46.3, 46.6	31.8, 32.0	51.2, 51.4	46.6, 46.6	46.9, 46.8	**55.2**, **55.0**
		NMI	13.7, 14.0	30.2, 30.7	13.5, 13.2	51.1, 51.7	48.1, 48.5	51.6, 51.8	**58.4**, **58.6**
		ARI	16.6, 16.7	26.3, 26.6	9.8, 9.7	40.9, 41.1	35.7, 35.8	38.0, 37.9	**46.8**, **46.8**
	9k	ACC	44.3, 44.6	48.3, 49.0	32.3, 32.0	**55.7**, **55.9**	43.4, 43.5	47.8, 47.9	53.4, 53.3
		NMI	14.3, 14.7	31.4, 32.1	14.1, 13.5	51.9, 52.7	46.6, 47.4	51.6, 52.0	**58.9**, **59.0**
		ARI	17.8, 18.0	30.2, 30.9	10.8, 10.3	43.6, 44.2	34.4, 34.7	38.2, 38.4	**46.2**, **46.1**

The advantage is clearest on structure-sensitive metrics. ECI-PP gives the best NMI in $46/48$ entries and the best ARI in $43/48$, compared with $33/48$ for ACC. Since NMI and ARI better reflect grouping consistency than matched-label accuracy alone, this suggests that ECI-PP most reliably improves the learned clustering structure. For example, on single-expert CIFAR100-20, ECI-PP raises NMI from roughly 44–46% for the strongest non-ECI competitors to 47–50% across budgets; on multi-expert Reuters, it raises NMI from about 53–58% for Weighted ProbPair to 65–69%. Dataset-wise, ECI-PP is best in all entries on CIFAR100-20, Reuters, and STL10, and in nearly all entries on CIFAR10 and ImageNet10. FMNIST and MNIST show more competition from Weighted ProbPair or SpherePair, especially on ACC and low-budget MNIST, but ECI-PP remains strongest or near-strongest on NMI/ARI in most cases.

The ProbPair variants show a consistent ablation pattern. Weighted ProbPair improves over ProbPair in most entries, and this effect is stronger under multi-expert supervision ($67/72$ entries) than under single-expert supervision ($62/72$ entries). This supports the interpretation that information-based weighting is especially useful when low-decisiveness constraints may reflect expert unfamiliarity. ECI-PP further improves over Weighted ProbPair in $129/144$ entries, showing that weighting alone does not replace explicit estimation, correction, and integration.

Notably, VolMaxDCC also provides a useful noise-aware reference among the non-ProbPair baselines. It improves over VanillaDCC and is competitive on some datasets, but reaches the top two in only $15/144$ entries. This suggests that explicit noise modeling is beneficial, yet its hard-noise-oriented formulation is not sufficient for the expert-conditioned probabilistic supervision in UPCC.

The main caveat remains the severely imbalanced RCV1-10 benchmark. SpherePair remains the main ACC competitor there, consistent with the advantage of its angular geometry for imbalanced pair structures. Nevertheless, ECI-PP gives the best NMI across all RCV1-10 budgets and regimes, and is often strongest in ARI. Thus, ECI-PP corrects much of the weakness of plain ProbPair/Weighted ProbPair on this benchmark, while the ACC results indicate that severe class imbalance remains a limitation for the probabilistic formulation.

D.2 Comparison without pretraining

Table 11 repeats the single-expert comparison without the autoencoder pretraining stage for pretrain-capable methods, denoted by † in the table: CIDEC†, SpherePair†, ProbPair†, Weighted ProbPair†, and ECI-PP†. These methods are trained from random initialization, while VanillaDCC and VolMaxDCC are unchanged because they have no pretraining variant. All other settings match Table 9: `lv0.01` supervision, corruption probability 0.3, and 3k/6k/9k constraints.

The main effect of removing pretraining is not a collapse of ECI-PP†, but a sharper separation from the other pretrain-capable methods. ECI-PP† ranks first in $68/72$ test entries and within the top two in $71/72$, compared with $59/72$ and $66/72$ in the pretrained comparison. It is also best in every entry on seven of the eight datasets; the only non-best entries occur on RCV1-10 ACC/ARI at 6k/9k. This indicates that ECI-PP† is less dependent on a favorable autoencoder initialization than the compared deep baselines.

The contrast with SpherePair† is also clearer without pretraining. ECI-PP† outperforms SpherePair† in $68/72$ entries, compared with $62/72$ in the pretrained setting, and achieves the best NMI in all cases. Its absolute NMI margins over SpherePair† range from about 3–12%, with visible gains on FMNIST, MNIST, and STL10. This suggests that the Estimator–Corrector–Integrator structure helps form a more reliable grouping structure when the representation is learned from scratch.

The ProbPair variants show a more mixed response to random initialization. Weighted ProbPair† still improves over ProbPair† in most entries, so information-based weighting remains useful. However, Weighted ProbPair† no longer compares as consistently with SpherePair†: it beats SpherePair† in only $33/72$ entries, down from $53/72$ with pretraining. By contrast, ECI-PP† improves over Weighted ProbPair† in all entries, showing that weighting alone is not sufficient when the representation starts from a weaker initialization.

The remaining caveat is again the severely imbalanced RCV1-10 benchmark. SpherePair† keeps stronger ACC at 9k and stronger ARI at 6k/9k, consistent with the robustness of its angular geometry under severe imbalance. Nevertheless, ECI-PP† gives the best NMI at every budget and substantially

Table 11: Comparative performance (%) (ACC, NMI, ARI) across datasets for methods under the `1v0.01` setting with corruption probability 0.3 and 3k/6k/9k constraints. † indicates models without pretraining. Blue and black represent training and test results, respectively. Best results are in **bold**, and second-best are underlined.

			External baselines				Our progressive models		
			Vanilla-DCC	VolMax-DCC	CIDEC†	SpherePair†	ProbPair†	Weighted ProbPair†	**ECI-PP†** (Full)
CIFAR100-20	3k	ACC	14.7, 14.7	42.8, 42.5	14.4, 14.2	44.0, 43.8	43.2, 43.0	42.0, 41.9	**46.6**, **46.5**
		NMI	11.2, 11.8	42.5, 42.9	11.2, 11.5	41.9, 42.4	39.7, 40.4	38.9, 39.5	**45.5**, **46.2**
		ARI	4.0, 4.1	26.2, 26.0	3.7, 3.5	27.5, 27.5	25.7, 25.8	24.9, 25.0	**30.5**, **30.8**
	6k	ACC	16.5, 16.6	45.4, 45.3	12.3, 12.2	47.6, 47.8	44.7, 44.9	45.2, 45.5	**49.3**, **49.1**
		NMI	12.6, 13.5	43.9, 44.5	8.6, 9.4	42.9, 43.9	39.9, 40.8	40.2, 41.0	**47.8**, **48.3**
		ARI	5.4, 5.6	28.2, 28.2	2.7, 2.7	30.5, 30.9	26.9, 27.3	27.1, 27.6	**33.9**, **33.9**
	9k	ACC	16.8, 17.0	46.1, 45.9	12.0, 11.9	48.9, 49.0	47.4, 47.5	47.8, 48.2	**52.2**, **52.5**
		NMI	13.5, 14.6	45.6, 46.1	8.1, 9.0	42.5, 43.6	40.9, 42.0	42.1, 43.1	**49.3**, **50.1**
		ARI	6.0, 6.3	30.4, 30.5	2.9, 2.9	31.0, 31.4	28.9, 29.2	30.0, 30.4	**35.9**, **36.3**
CIFAR10	3k	ACC	34.6, 34.6	79.9, 79.8	39.0, 39.0	80.4, 80.7	83.0, 83.1	82.8, 82.9	**84.5**, **84.4**
		NMI	27.9, 28.1	69.3, 69.4	29.1, 29.6	70.8, 71.1	71.8, 72.1	73.3, 73.6	**76.0**, **75.9**
		ARI	20.2, 20.3	64.4, 64.4	17.8, 17.7	67.2, 67.4	68.6, 68.7	70.3, 70.4	**72.3**, **72.1**
	6k	ACC	35.5, 35.7	84.7, 84.6	35.7, 35.9	84.2, 84.5	84.1, 84.4	85.1, 85.3	**87.0**, **87.0**
		NMI	27.5, 28.3	73.6, 73.7	27.4, 29.1	71.6, 72.5	73.1, 74.0	74.3, 74.7	**78.5**, **78.5**
		ARI	20.7, 21.2	71.0, 70.9	18.6, 19.0	70.2, 70.8	70.8, 71.2	72.4, 72.6	**75.8**, **75.7**
	9k	ACC	38.3, 38.7	82.5, 82.5	31.9, 32.1	83.3, 84.0	85.2, 85.6	86.1, 86.4	**87.7**, **87.7**
		NMI	29.9, 31.2	71.9, 72.2	23.9, 25.8	70.1, 71.7	73.9, 74.8	75.0, 75.8	**79.1**, **79.0**
		ARI	23.2, 24.0	68.3, 68.3	16.7, 17.3	68.9, 70.2	71.9, 72.5	73.5, 73.9	**76.8**, **76.7**
FMNIST	3k	ACC	36.6, 36.5	63.3, 62.5	36.3, 35.6	64.6, 63.6	58.0, 57.3	61.9, 61.3	**67.3**, **66.6**
		NMI	31.1, 31.1	55.5, 55.2	29.2, 28.9	57.7, 57.2	51.8, 51.4	53.2, 53.2	**63.1**, **62.4**
		ARI	19.7, 19.5	44.4, 43.6	15.7, 14.9	48.3, 47.3	40.1, 39.3	43.0, 42.6	**51.8**, **50.7**
	6k	ACC	38.6, 38.5	66.2, 65.1	34.1, 33.6	64.7, 63.7	65.8, 65.0	66.8, 66.0	**69.0**, **67.7**
		NMI	30.9, 31.3	56.7, 56.3	27.5, 27.8	54.1, 54.1	55.8, 55.7	56.1, 56.2	**64.2**, **63.4**
		ARI	20.7, 20.9	46.6, 45.5	16.4, 16.1	45.9, 45.4	46.6, 45.8	47.3, 46.6	**53.1**, **51.6**
	9k	ACC	39.8, 40.2	66.5, 65.5	34.7, 34.3	67.6, 66.9	69.6, 68.8	71.0, 69.9	**72.4**, **71.2**
		NMI	30.1, 31.3	57.2, 56.8	27.1, 28.0	55.5, 55.9	58.2, 58.6	59.2, 59.3	**65.6**, **64.7**
		ARI	21.5, 22.2	46.4, 45.2	15.9, 15.7	48.5, 48.0	49.7, 49.2	51.0, 50.1	**55.5**, **54.0**
ImageNet10	3k	ACC	35.4, 37.3	86.7, 86.9	50.8, 52.0	88.5, 89.8	87.3, 87.6	88.9, 89.1	**92.5**, **92.6**
		NMI	27.7, 30.8	78.8, 79.4	42.8, 46.5	78.8, 81.6	78.4, 79.3	80.6, 81.4	**87.9**, **88.2**
		ARI	19.1, 20.8	74.1, 73.9	30.1, 31.5	77.5, 79.7	75.7, 76.0	77.9, 78.2	**85.5**, **85.6**
	6k	ACC	40.6, 42.0	87.2, 87.5	32.0, 33.6	87.0, 89.4	88.8, 89.8	89.6, 90.1	**92.5**, **92.8**
		NMI	30.6, 35.2	78.3, 79.3	23.3, 29.4	74.9, 80.2	79.2, 81.5	80.9, 82.3	**87.8**, **88.4**
		ARI	23.2, 26.0	75.1, 75.6	16.1, 19.4	74.3, 78.7	77.7, 79.4	79.3, 79.9	**85.5**, **86.0**
	9k	ACC	39.0, 41.5	87.4, 88.1	31.9, 34.5	85.2, 88.9	88.4, 90.1	88.9, 90.0	**92.6**, **93.1**
		NMI	27.6, 33.7	77.8, 79.2	23.9, 32.4	71.1, 79.1	78.3, 82.2	79.4, 82.5	**87.9**, **88.5**
		ARI	21.1, 25.2	75.2, 76.1	17.1, 22.1	70.9, 77.9	76.8, 79.9	77.8, 79.8	**85.6**, **86.2**
MNIST	3k	ACC	25.9, 26.1	65.6, 66.4	41.6, 41.0	72.5, 73.2	67.6, 68.8	70.1, 70.7	**80.0**, **80.6**
		NMI	16.0, 17.0	52.7, 54.1	35.1, 35.6	61.5, 63.0	54.0, 55.9	57.2, 58.4	**68.6**, **69.6**
		ARI	10.4, 10.8	47.8, 48.9	19.6, 18.8	56.4, 57.7	48.7, 50.4	52.1, 53.1	**64.9**, **65.9**
	6k	ACC	33.3, 33.9	73.4, 74.4	36.4, 36.3	76.8, 77.9	76.6, 77.5	82.4, 83.2	**87.4**, **88.0**
		NMI	24.2, 25.8	58.4, 60.5	29.6, 31.0	63.3, 65.2	62.3, 64.2	67.5, 69.1	**75.6**, **76.8**
		ARI	18.4, 19.4	55.2, 56.8	18.2, 18.0	60.1, 61.7	59.3, 60.9	66.5, 68.0	**74.9**, **76.1**
	9k	ACC	38.9, 40.3	75.6, 76.9	36.0, 36.8	80.1, 81.3	82.2, 83.6	86.0, 87.2	**88.7**, **89.2**
		NMI	29.5, 32.2	59.3, 61.6	28.3, 30.9	67.0, 69.6	67.5, 70.2	72.2, 74.6	**77.4**, **78.5**
		ARI	23.1, 24.8	56.5, 58.7	19.1, 19.8	65.5, 67.7	66.4, 68.9	72.3, 74.6	**77.2**, **78.2**
REUTERS	3k	ACC	53.2, 54.0	62.8, 63.2	58.1, 56.0	76.5, 78.8	69.4, 71.5	71.6, 73.9	**80.0**, **80.7**
		NMI	15.1, 17.9	25.6, 28.0	26.2, 30.4	49.5, 54.7	44.4, 48.8	46.7, 51.2	**56.3**, **57.9**
		ARI	17.0, 18.0	31.2, 32.4	29.6, 29.7	54.0, 59.2	46.7, 51.3	48.7, 53.3	**61.6**, **63.2**
	6k	ACC	66.0, 70.6	72.5, 76.8	45.1, 46.1	78.4, 81.2	75.6, 78.2	74.5, 77.1	**83.7**, **85.2**
		NMI	27.7, 36.3	36.0, 44.9	9.9, 14.1	50.3, 57.1	48.1, 53.5	48.1, 53.5	**59.5**, **62.7**
		ARI	34.0, 42.6	42.4, 50.8	11.3, 14.3	54.5, 61.1	52.6, 57.9	51.2, 56.3	**65.0**, **68.1**
	9k	ACC	69.7, 77.0	71.7, 76.8	52.9, 54.5	77.5, 82.2	76.1, 80.1	75.3, 79.3	**81.9**, **83.1**
		NMI	30.5, 43.3	34.3, 43.4	16.8, 25.5	48.7, 57.0	49.5, 57.5	47.7, 54.6	**57.6**, **60.2**
		ARI	36.8, 50.4	40.9, 50.4	19.6, 25.5	52.6, 61.4	52.3, 60.0	51.3, 58.6	**62.0**, **64.7**
STL10	3k	ACC	30.9, 31.5	72.8, 73.3	49.2, 50.3	80.9, 82.9	80.3, 81.1	80.8, 81.5	**84.2**, **84.7**
		NMI	21.1, 23.4	64.8, 66.5	38.2, 41.9	67.2, 71.0	66.9, 69.2	67.5, 69.2	**74.5**, **75.1**
		ARI	14.1, 15.0	56.3, 57.0	27.0, 28.5	64.3, 67.8	64.0, 65.5	64.7, 66.0	**71.0**, **71.6**
	6k	ACC	30.4, 31.6	76.8, 77.5	35.1, 36.8	78.7, 81.7	80.3, 81.9	81.5, 82.7	**85.2**, **85.7**
		NMI	19.5, 22.2	65.8, 67.7	26.8, 31.7	62.6, 68.4	66.3, 69.6	67.9, 70.7	**75.2**, **76.2**
		ARI	14.3, 15.7	60.0, 61.3	18.6, 21.3	60.4, 65.6	63.8, 66.4	65.7, 67.8	**72.1**, **73.0**
	9k	ACC	35.5, 37.8	79.0, 80.1	31.8, 33.4	76.9, 80.8	80.3, 82.8	81.3, 82.7	**85.3**, **85.7**
		NMI	22.9, 27.6	66.7, 69.3	23.3, 28.8	59.7, 66.9	66.0, 70.8	67.1, 70.6	**75.6**, **76.3**
		ARI	17.1, 19.8	62.8, 64.7	16.4, 19.3	57.6, 64.1	63.6, 67.7	65.3, 67.8	**72.4**, **73.0**
RCV1-10	3k	ACC	40.0, 40.1	46.3, 46.4	39.4, 37.4	52.1, 52.2	45.2, 45.1	44.6, 44.6	**55.5**, **55.7**
		NMI	12.3, 12.4	21.5, 21.9	21.6, 21.4	51.5, 51.7	42.3, 42.5	43.9, 44.1	**54.7**, **55.1**
		ARI	16.4, 16.5	21.2, 21.4	18.3, 19.6	42.6, 42.7	34.5, 34.5	35.0, 35.0	**47.6**, **47.9**
	6k	ACC	52.4, 52.6	**62.1**, **62.4**	38.5, 37.0	62.0, 61.9	47.0, 47.1	48.2, 48.3	59.7, 59.9
		NMI	21.8, 22.2	36.0, 36.7	15.1, 14.9	55.8, 56.2	44.1, 44.5	47.4, 47.4	**58.5**, **58.8**
		ARI	28.0, 28.2	42.9, 43.3	16.2, 16.1	**52.8**, **52.9**	37.5, 37.6	39.0, 38.9	51.5, 51.7
	9k	ACC	46.2, 46.4	61.3, 61.8	41.6, 42.0	**66.6**, **66.8**	45.7, 45.5	49.8, 49.9	64.1, 64.1
		NMI	15.5, 15.9	35.7, 36.7	14.4, 14.9	58.0, 58.8	43.6, 44.0	49.7, 49.9	**61.4**, **61.6**
		ARI	19.8, 20.1	44.7, 45.4	15.8, 16.4	**57.8**, **58.4**	35.6, 35.6	42.9, 43.0	56.0, 56.0

improves over ProbPair†/Weighted ProbPair†, indicating stronger grouping consistency despite the remaining imbalance-related ACC/ARI limitation.

D.3 Comparison with expert-aware baseline extensions

Table 12: Comparative performance (%) (ACC, NMI, ARI) across datasets for ECI-PP, SP^{EA}, and WPP^{EA} under the `multi3` setting, with corruption probability 0.3 and 3k/6k/9k total constraints. SP^{EA} and WPP^{EA} denote the expert-aware ensemble extensions of SpherePair and Weighted ProbPair, respectively. Blue and black represent training and test results, respectively. Best results are in **bold**, and second-best are underlined.

		3k			6k			9k		
		ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
CIFAR100-20	SP^{EA}	42.2, 43.2	41.7, 42.5	26.6, 27.0	41.0, 40.9	39.7, 40.9	25.5, 25.8	41.9, 42.3	39.2, 40.5	26.0, 27.1
	WPP^{EA}	43.1, 43.4	41.4, 42.2	26.4, 26.5	41.5, 41.4	40.4, 41.3	25.8, 26.1	46.4, 46.4	42.9, 43.8	29.8, 29.9
	ECI-PP	**45.6**, **45.3**	**45.0**, **45.5**	**29.7**, **29.8**	**49.1**, **49.3**	**47.1**, **47.7**	**32.1**, **32.4**	**48.6**, **48.8**	**48.1**, **48.5**	**32.9**, **32.9**
CIFAR10	SP^{EA}	76.6, 77.1	70.7, 71.5	64.0, 64.8	75.5, 75.9	67.9, 69.4	62.4, 63.5	74.1, 75.2	67.0, 69.0	61.3, 62.9
	WPP^{EA}	82.6, **82.5**	74.1, 74.2	69.4, 69.2	81.2, 81.2	74.6, 74.9	69.6, 69.6	**82.5**, **82.8**	74.2, 74.5	70.0, 70.3
	ECI-PP	**82.8**, **82.5**	**76.0**, **75.9**	**70.8**, **70.7**	**84.0**, **84.0**	**77.0**, **77.1**	**72.6**, **72.6**	81.6, 82.1	**77.8**, **77.8**	**73.1**, **73.0**
FMNIST	SP^{EA}	63.2, 61.6	60.3, 59.7	48.4, 46.9	63.5, 62.5	59.4, 59.9	48.5, 48.3	62.5, 62.5	58.6, 60.2	49.0, 49.4
	WPP^{EA}	**64.5**, **64.3**	59.7, 59.8	47.9, 47.7	**68.0**, **68.3**	62.1, 62.4	**52.0**, **52.0**	**71.8**, **71.7**	64.6, 64.8	**55.2**, **55.1**
	ECI-PP	60.1, 59.2	**62.7**, **61.9**	**49.5**, **48.4**	62.0, 61.1	**64.3**, **63.8**	51.0, 51.0	65.3, 64.8	**66.0**, **65.6**	53.4, 52.8
ImageNet10	SP^{EA}	81.7, 85.3	78.0, 79.8	71.8, 75.2	82.1, 86.6	77.8, 79.6	72.8, 75.8	81.2, 81.4	75.3, 76.7	70.6, 71.3
	WPP^{EA}	90.5, 90.8	83.8, 84.2	81.8, 82.0	90.9, 91.6	83.4, 84.8	82.2, 83.4	91.2, 92.1	82.4, 84.7	82.2, 83.8
	ECI-PP	**91.5**, **91.8**	**86.9**, **87.3**	**84.1**, **84.4**	**91.9**, **92.1**	**87.2**, **87.8**	**84.6**, **85.0**	**92.4**, **92.6**	**87.7**, **88.1**	**85.4**, **85.7**
MNIST	SP^{EA}	80.5, 81.1	75.2, 76.7	70.7, 71.7	78.6, 79.6	72.7, 75.5	67.9, 70.2	77.5, 78.1	70.2, 73.7	65.7, 67.8
	WPP^{EA}	**87.9**, **88.7**	**77.2**, **78.5**	**76.2**, **77.4**	85.8, **86.4**	77.4, 79.0	75.5, 76.8	**89.2**, **90.4**	78.1, 80.4	**78.2**, **80.3**
	ECI-PP	80.9, 81.5	75.7, 77.1	71.9, 73.2	**86.1**, **86.4**	**79.8**, **80.5**	**77.8**, **78.3**	84.7, 85.1	**79.9**, **80.7**	77.1, 77.5
REUTERS	SP^{EA}	75.8, 76.1	41.8, 45.7	48.0, 50.2	72.0, 77.0	35.3, 43.8	40.9, 49.8	68.4, 77.0	30.6, 43.1	34.7, 49.6
	WPP^{EA}	82.9, 83.3	55.4, 58.3	62.4, 64.5	81.4, 83.7	53.6, 58.4	59.5, 64.4	77.7, 80.9	50.3, 56.0	54.8, 61.5
	ECI-PP	**85.0**, **86.3**	**62.0**, **65.0**	**68.3**, **71.1**	**87.3**, **88.5**	**63.8**, **67.2**	**71.3**, **74.3**	**88.3**, **89.6**	**65.4**, **68.7**	**72.9**, **76.0**
STL10	SP^{EA}	74.2, 76.7	66.5, 68.1	59.7, 61.1	73.0, 75.4	64.0, 67.3	57.4, 60.1	72.7, 72.9	62.5, 66.2	56.7, 58.7
	WPP^{EA}	84.0, 84.5	72.4, 73.8	69.0, 70.0	85.2, 85.6	73.7, 75.2	71.3, 72.0	85.1, 86.1	73.3, 75.4	71.1, 72.7
	ECI-PP	**85.7**, **85.9**	**75.7**, **76.3**	**72.2**, **72.5**	**87.4**, **87.5**	**77.5**, **78.1**	**75.1**, **75.4**	**87.9**, **87.7**	**78.3**, **78.5**	**75.9**, **75.8**
RCV1-10	SP^{EA}	**60.2**, **60.9**	52.6, 53.1	**48.9**, **49.5**	**68.2**, **63.5**	54.8, 53.8	**55.7**, **51.4**	**71.2**, **67.7**	54.8, 54.4	**58.8**, **55.9**
	WPP^{EA}	52.5, 47.6	52.6, 52.5	41.9, 38.3	54.6, 52.6	53.6, 53.7	45.0, 42.8	56.5, 58.4	53.5, 54.2	47.0, 49.2
	ECI-PP	55.3, 55.2	**56.8**, **56.9**	45.6, 45.4	55.2, 55.0	**58.4**, **58.6**	46.8, 46.8	53.4, 53.3	**58.9**, **59.0**	46.2, 46.1

Table 12 compares ECI-PP with SP^{EA} and WPP^{EA} under the `multi3` setting, where SP^{EA} and WPP^{EA} denote our expert-aware ensemble extensions of SpherePair and Weighted ProbPair, respectively. As described in Appendix C.4, these extensions instantiate multiple expert-weighted members for a baseline and aggregate their partitions through ensemble clustering. The expert-weighting parameter α is selected on the reserved validation split, with the selected values reported in Table 8. Thus, the comparison should be read as an auxiliary reference rather than a native baseline comparison: SP^{EA} and WPP^{EA} use expert identities, but they also introduce validation tuning and ensemble aggregation, making them not strictly comparable with the native ECI-PP protocol.

The extensions substantially strengthen the corresponding baselines, confirming that expert identities carry useful information in the multi-expert setting. WPP^{EA} benefits clearly on datasets such as MNIST, FMNIST, and STL10, while SP^{EA} gains competitiveness on Reuters and RCV1-10. However, these gains conflate two effects: using expert identities and ensembling multiple expert-weighted members. For this reason, the results are best interpreted as stronger expert-aware references, not as fully symmetric replacements for the native baselines in Table 1.

Even against these stronger references, ECI-PP remains the most reliable method overall. It obtains the best result in $55/72$ test entries and ranks within the top two in $68/72$. The advantage is clearest in NMI, where ECI-PP is best in $23/24$ cases, indicating that its expert-conditioned design most consistently improves grouping structure. Dataset-wise, ECI-PP is best in all entries on CIFAR100-20, ImageNet10, Reuters, and STL10, and remains strongest in most CIFAR10 entries. The main losses occur on FMNIST and MNIST, mostly on ACC/ARI, where WPP^{EA} is particularly strong, and on RCV1-10 ACC/ARI, where the angular geometry of SP^{EA} remains advantageous under severe imbalance. Overall, expert-aware ensembling makes strong baselines substantially more competitive, but ECI-PP still provides a more direct and generally stronger way to use expert identities without relying on a validation-tuned ensemble wrapper.

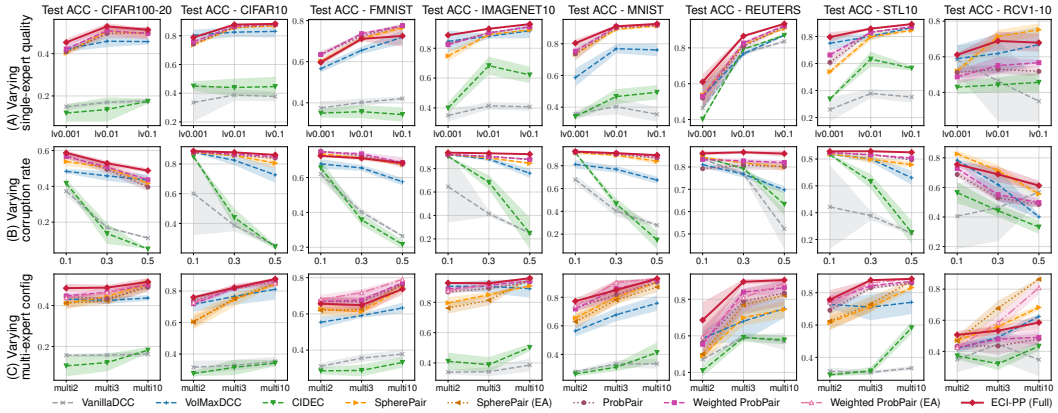


Figure 5: Full robustness results in ACC (mean±std over 5 runs) across datasets under varying (A) expert quality, (B) corruption probability, and (C) multi-expert configuration, with 9k constraints. The multi-expert row additionally includes expert-aware SpherePair and Weighted ProbPair extensions.

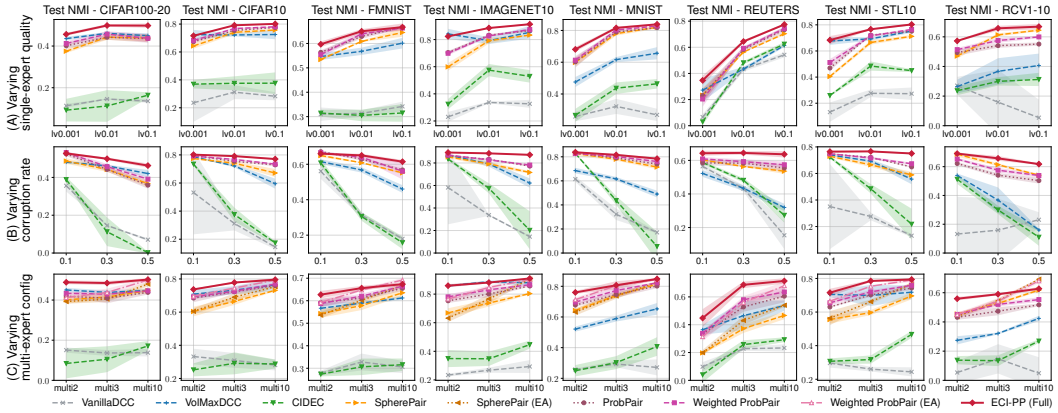


Figure 6: Full robustness results in NMI (mean±std over 5 runs) across datasets under varying (A) expert quality, (B) corruption probability, and (C) multi-expert configuration, with 9k constraints. The multi-expert row additionally includes expert-aware SpherePair and Weighted ProbPair extensions.

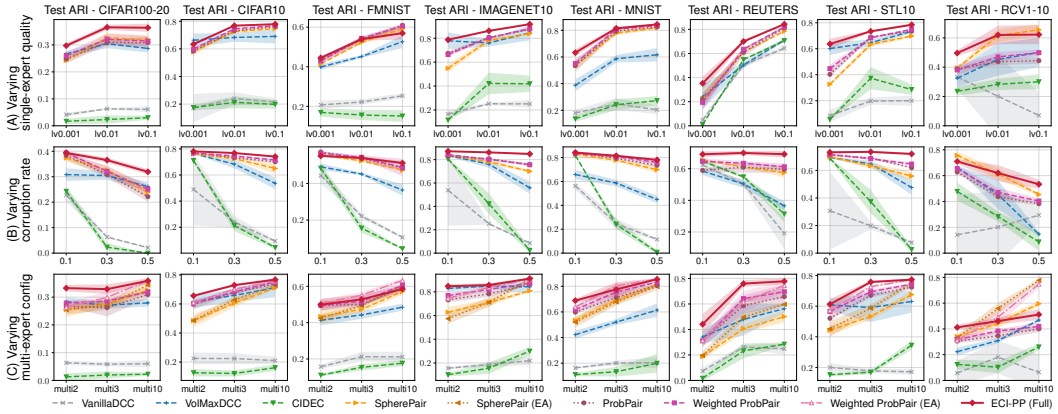


Figure 7: Full robustness results in ARI (mean±std over 5 runs) across datasets under varying (A) expert quality, (B) corruption probability, and (C) multi-expert configuration, with 9k constraints. The multi-expert row additionally includes expert-aware SpherePair and Weighted ProbPair extensions.

D.4 Robustness across supervision conditions

Figs. 5 to 7 report the full ACC, NMI, and ARI results corresponding to the robustness study in the main text. All three sweeps use a fixed budget of 9k constraints, and the central setting in each sweep, namely `lv0.01` expert quality, corruption probability 0.3, and `multi3`, matches the default setting used in Table 1. Thus, these figures also cover the main-comparison results in Table 1, while additionally reporting the standard deviations omitted from the table for space. For the multi-expert sweep, we further include SP^{EA} and WPP^{EA}, the expert-aware ensemble extensions introduced in Appendix D.3, as stronger references that also exploit expert identities.

Effects of supervision factors. Across the three sweeps, the results largely follow the expected supervision-quality ordering. In the single-expert sweep, stronger experts generally yield better clustering quality; in the corruption sweep, increasing corruption leads to clear degradation. In the multi-expert sweep, moving from `multi2` to `multi10` also tends to help the stronger methods, because the protocol progressively reduces expert blind spots and increases average expert accuracy, as quantified in Table 6. Since the total constraint budget is fixed throughout, the performance trends mainly reflect changes in supervision quality and heterogeneity rather than in the amount of supervision.

Baseline behavior. The non-ECI baselines show a clear hierarchy as supervision becomes less reliable. Plain end-to-end methods such as VanillaDCC and CIDEC often nearly collapse under low-quality or heterogeneous supervision, especially in NMI and ARI, suggesting that cluster-assignment training is vulnerable when pairwise supervision is inconsistent. VolMaxDCC is usually more resilient, consistent with the benefit of explicit noise modeling. SpherePair and the ProbPair-family baselines are substantially stronger than the plain end-to-end methods, reflecting the robustness of geometric representation learning under noisy supervision. Nevertheless, corruption still causes visible degradation for the stronger baselines, particularly on FMNIST, ImageNet10, and STL10.

ECI-PP robustness. Across the supervision sweeps, ECI-PP better preserves clustering quality than the native baselines, especially as expert quality decreases or corruption increases. This advantage is most consistent on NMI and ARI, suggesting that the correction-and-integration pipeline mainly improves the recovered grouping structure. ACC is more competitive and contains more exceptions, especially on FMNIST, MNIST, and RCV1-10, where Weighted ProbPair, SP^{EA}, or WPP^{EA} can sometimes match or exceed ECI-PP. This is consistent with the main-comparison results (Appendix D.1): matched-label accuracy can favor angular or ensemble-based baselines on some datasets, even when ECI-PP remains stronger or among the strongest on grouping-oriented metrics (NMI/ARI). The expert-aware extensions further clarify the role of expert identities: SP^{EA} and WPP^{EA} often improve over their native counterparts in the multi-expert sweep, confirming that expert identity information is useful under heterogeneous supervision. However, these extensions rely on validation-selected expert weighting and ensemble aggregation, whereas ECI-PP uses expert-conditioned correction within the learning pipeline. Overall, the full results support the main-text conclusion that ECI-PP better preserves clustering quality when supervision is fallible, heterogeneous, or corrupted.

D.5 Robustness to relation and corruption misspecification

We further examine whether ECI-PP's advantage depends on the default experimental constraint generator. The three conditions defined in Appendix C.3 isolate different departures from that generator: *Dot+Flip* changes only the corruption mechanism, *JSD+Uniform* changes only the clean relation construction using normalized Jensen–Shannon similarity, and *JSD+Flip* changes both.

As shown in Tables 13 and 14, ECI-PP provides the strongest overall performance across the full evaluation, ranking first in $60/72$ dataset–condition–metric entries and within the top two in $70/72$. The advantage holds under each type of generator shift: ECI-PP ranks first in $17/24$ entries under *Dot+Flip*, $22/24$ under *JSD+Uniform*, and $21/24$ under *JSD+Flip*. Notably, even in the Flip-based settings, which are naturally aligned with VolMaxDCC's binary-noise modeling, ECI-PP still ranks first in $38/48$ entries and within the top two in all 48. Together, these results show that ECI-PP's advantage is not specific to the default relation construction or corruption mechanism used in the main experiments.

Table 13: Clustering performance (%) under alternative relation and corruption constructions on CIFAR100-20, CIFAR10, FMNIST, and ImageNet10 in the pretrained `multi3` setting with corruption probability 0.3 and 9k constraints. Dot denotes the default inner-product relation, JSD denotes the normalized Jensen–Shannon relation, Uniform denotes the default uniform-replacement corruption mechanism, and Flip denotes complementary flipping. Results are mean±std over 5 runs. Best results are in **bold**, and second-best are underlined.

			External baselines				**Our progressive models**		
Dataset	Condition	Metric	Vanilla-DCC	VolMax-DCC	CIDEC	SpherePair	ProbPair	Weighted ProbPair	**ECI-PP** (Full)
CIFAR100-20	Dot+Flip	ACC	5.0±0.0	**43.8±2.6**	5.0±0.0	22.9±2.2	26.3±2.5	28.9±2.2	41.8±2.3
		NMI	0.0±0.0	**44.6±1.6**	0.2±0.3	17.5±1.5	22.0±1.4	24.7±1.4	41.0±1.0
		ARI	0.0±0.0	**27.9±1.8**	0.0±0.0	8.1±1.4	11.0±1.5	12.8±1.6	26.2±1.2
	JSD+Uniform	ACC	16.0±0.3	43.7±2.2	13.8±1.5	45.7±1.6	47.0±1.0	47.5±0.7	**50.3±1.2**
		NMI	15.3±0.7	44.9±1.3	12.8±1.9	43.9±0.9	44.5±0.8	45.5±0.6	**50.2±0.5**
		ARI	6.6±0.2	28.1±1.5	2.1±0.9	29.3±1.1	29.2±0.7	30.1±0.6	**35.3±0.6**
	JSD+Flip	ACC	5.0±0.0	**43.8±1.3**	7.1±1.9	23.3±0.7	27.2±1.4	29.1±1.0	41.8±1.7
		NMI	0.0±0.0	**44.7±1.0**	3.2±2.6	17.5±0.6	22.6±1.1	24.9±0.9	41.5±0.9
		ARI	0.0±0.0	**27.8±1.0**	0.2±0.2	8.1±0.6	11.1±1.0	13.1±0.9	26.2±1.2
CIFAR10	Dot+Flip	ACC	10.0±0.0	76.2±5.9	14.4±4.1	48.6±3.6	65.9±4.0	68.7±2.7	**79.4±3.5**
		NMI	0.0±0.0	67.7±5.2	4.8±4.4	35.9±3.5	48.0±3.0	50.4±3.1	**70.8±2.1**
		ARI	0.0±0.0	60.8±6.8	2.2±2.0	27.7±3.8	43.1±4.2	46.0±3.6	**65.3±3.3**
	JSD+Uniform	ACC	31.7±3.4	75.0±2.8	27.1±3.3	79.2±1.4	82.7±1.9	83.7±0.7	**85.7±1.0**
		NMI	30.7±5.1	72.3±2.5	24.7±4.5	69.0±0.5	73.4±1.2	74.2±0.6	**77.4±0.8**
		ARI	21.4±4.1	64.4±4.0	10.7±1.5	64.1±0.9	69.2±1.8	70.4±0.9	**73.7±1.3**
	JSD+Flip	ACC	11.7±3.7	73.4±6.4	14.4±6.2	51.2±4.0	66.8±3.5	64.4±4.4	**78.9±2.8**
		NMI	1.7±3.7	67.7±4.1	5.8±8.5	37.8±3.9	49.5±2.8	48.8±4.1	**70.4±2.0**
		ARI	0.8±1.8	59.8±6.7	1.3±1.9	29.9±5.1	45.0±3.4	43.1±4.2	**63.8±3.0**
FMNIST	Dot+Flip	ACC	10.0±0.0	53.6±4.0	16.7±4.1	33.9±3.4	52.7±3.2	50.7±3.5	**63.3±5.2**
		NMI	0.0±0.0	52.6±2.4	8.4±5.3	21.6±2.3	40.6±1.5	39.8±2.8	**59.0±2.1**
		ARI	0.0±0.0	37.2±4.0	1.8±1.5	13.8±2.6	31.3±1.5	30.8±3.0	**47.2±2.8**
	JSD+Uniform	ACC	33.4±0.9	59.2±2.8	28.3±2.0	61.6±2.0	70.2±1.9	**71.6±4.4**	64.4±3.9
		NMI	32.2±1.4	59.5±1.4	28.8±2.2	59.2±0.9	63.4±1.1	64.0±2.3	**65.8±1.1**
		ARI	20.5±0.9	45.8±2.2	14.6±1.0	48.8±1.3	54.0±2.1	**54.8±3.9**	53.1±2.2
	JSD+Flip	ACC	14.0±5.4	55.8±1.3	18.9±0.9	33.2±2.9	53.7±4.1	53.6±2.5	**63.7±1.1**
		NMI	5.1±6.1	55.1±1.3	12.0±1.8	22.5±0.9	41.2±2.5	41.9±2.2	**59.7±1.4**
		ARI	2.1±2.8	40.1±1.2	3.9±2.1	14.5±0.6	31.9±2.5	32.7±1.8	**47.0±1.7**
ImageNet10	Dot+Flip	ACC	21.1±0.6	**89.0±3.9**	10.0±0.0	41.1±2.0	67.7±5.8	69.1±7.3	88.9±1.7
		NMI	10.2±0.7	**84.5±3.0**	0.0±0.0	26.1±1.8	49.7±4.3	52.1±3.6	81.4±1.0
		ARI	4.9±0.6	**80.4±5.0**	0.0±0.0	17.8±1.4	45.6±5.6	47.3±5.8	78.7±1.6
	JSD+Uniform	ACC	32.6±1.7	91.1±4.9	28.8±2.9	87.9±1.7	90.5±1.3	91.8±0.7	**93.4±1.4**
		NMI	27.4±1.6	88.5±2.1	27.3±3.1	78.9±2.0	83.0±1.7	84.8±0.7	**88.7±0.8**
		ARI	18.2±1.4	85.4±3.8	9.6±2.1	76.6±2.3	81.2±2.0	83.4±1.0	**86.9±2.1**
	JSD+Flip	ACC	17.0±4.0	83.3±6.9	10.0±0.0	48.0±3.7	71.9±4.4	70.9±5.8	**90.2±0.7**
		NMI	7.2±4.1	81.5±5.7	0.0±0.0	32.2±3.2	55.8±2.5	53.8±4.1	**82.3±1.0**
		ARI	3.2±1.8	75.7±7.9	0.0±0.0	24.3±3.6	52.1±3.4	49.7±5.5	**80.3±1.0**

D.6 Held-out diagnostics

Diagnostic protocol. The held-out diagnostics complement the clustering results by tracking the training dynamics of ECI-PP's internal quantities. Following Appendix C.4, we construct a sample-disjoint diagnostic constraint set $\mathcal{C}^{\mathrm{diag}}$ from the test split, used only for diagnosis and never for optimization, early stopping, or hyperparameter selection. Benchmark labels are used only to define an external hard co-membership reference, rather than the canonical aleatoric relation $R^\star$. All diagnostics use `multi3` supervision with limited 3k training constraints in total (1k per expert) and corruption rate 0.3. To reduce the influence of a strong initial representation or a long warm-up phase, we use random initialization without unsupervised pretraining, give the Estimators and Integrator only one warm-up epoch, and then track the next 500 training iterations, each corresponding to one epoch. We report mean±std over 5 runs, recording at each iteration the Estimator out-of-fold relation $\hat{y}_i^{\mathrm{oof}}$, the corrected relation y_i^{cor}, and the Integrator relation y_i^{int} on $\mathcal{C}^{\mathrm{diag}}$.

Diagnostic metrics. For a held-out pair $i = (a_i, b_i)$ with benchmark labels t_{a_i} and t_{b_i}, define the external hard co-membership reference

$$o_i^\star := \mathbb{I}[t_{a_i} = t_{b_i}].$$

Table 14: Clustering performance (%) under alternative relation and corruption constructions on MNIST, Reuters, STL10, and RCV1-10 in the pretrained `multi3` setting with corruption probability 0.3 and 9k constraints. Dot denotes the default inner-product relation, JSD denotes the normalized Jensen–Shannon relation, Uniform denotes the default uniform-replacement corruption mechanism, and Flip denotes complementary flipping. Results are mean±std over 5 runs. Best results are in **bold**, and second-best are underlined.

Dataset	Condition	Metric	External baselines				Our progressive models		
			Vanilla-DCC	VolMax-DCC	CIDEC	SpherePair	ProbPair	Weighted ProbPair	**ECI-PP (Full)**
MNIST	Dot+Flip	ACC	11.3±0.0	53.3±5.3	14.4±6.8	53.0±6.8	65.1±4.1	65.0±7.9	**77.3±2.8**
		NMI	0.0±0.0	42.9±2.6	5.1±11.4	42.6±4.6	51.2±4.7	51.4±6.9	**70.7±1.5**
		ARI	0.0±0.0	34.3±4.3	0.5±1.2	32.3±4.8	45.5±4.9	45.5±7.8	**65.1±1.9**
	JSD+Uniform	ACC	30.2±3.0	68.3±3.0	31.6±1.1	77.2±1.8	86.4±3.0	89.4±1.6	**90.8±2.3**
		NMI	25.8±3.2	59.9±1.8	30.0±2.7	75.4±1.3	76.8±1.4	79.4±1.5	**82.2±1.7**
		ARI	17.1±2.3	53.1±2.9	12.6±1.2	69.6±2.0	75.0±2.8	78.7±2.5	**81.7±3.1**
	JSD+Flip	ACC	13.1±4.0	56.0±2.2	11.3±0.0	52.0±4.5	67.1±5.9	68.4±4.5	**79.3±4.4**
		NMI	1.9±4.2	46.1±1.3	0.0±0.0	43.9±3.0	54.1±2.4	54.4±2.3	**72.4±1.8**
		ARI	0.9±2.0	37.7±2.0	0.0±0.0	33.1±3.3	48.6±3.6	48.9±3.7	**67.5±3.1**
Reuters	Dot+Flip	ACC	42.4±4.6	59.7±7.9	38.7±0.0	44.6±3.7	53.3±5.5	57.0±4.8	**83.8±2.0**
		NMI	2.9±3.6	35.7±8.8	0.5±0.7	11.4±1.8	21.4±3.5	24.9±3.8	**58.7±3.1**
		ARI	3.3±4.1	33.8±11.9	0.0±0.0	11.4±2.1	22.0±4.2	25.3±4.6	**65.3±3.6**
	JSD+Uniform	ACC	58.4±0.9	68.6±6.1	47.7±5.5	65.5±4.9	78.2±8.6	80.9±5.1	**90.1±1.5**
		NMI	21.1±1.4	48.5±3.2	11.5±6.1	35.1±4.5	51.3±5.8	56.5±3.5	**70.0±3.2**
		ARI	24.1±1.3	49.3±6.2	7.8±4.9	37.2±5.4	57.1±8.0	61.3±5.9	**77.2±3.1**
	JSD+Flip	ACC	38.4±0.5	66.8±8.7	38.7±0.0	41.4±3.6	54.7±7.6	58.5±4.8	**83.0±4.1**
		NMI	0.3±0.5	42.0±7.8	0.0±0.0	9.8±3.1	22.5±4.9	27.6±3.6	**58.0±4.2**
		ARI	0.4±0.7	41.4±9.9	0.0±0.0	9.0±2.5	22.3±6.4	28.7±4.3	**64.6±4.8**
STL10	Dot+Flip	ACC	19.7±2.4	72.8±7.9	12.4±5.3	32.2±1.3	49.6±5.8	55.5±2.0	**79.6±3.4**
		NMI	10.1±2.3	**69.4±3.1**	3.8±7.7	18.7±1.5	34.3±4.0	37.3±2.6	68.1±2.1
		ARI	4.8±1.9	59.0±6.5	0.2±0.4	11.4±0.8	26.7±4.4	31.1±3.1	**63.0±4.1**
	JSD+Uniform	ACC	26.1±0.9	70.4±4.1	27.4±1.4	76.0±3.7	82.0±1.6	83.4±1.9	**86.9±0.9**
		NMI	22.5±1.9	70.2±1.8	28.0±1.5	64.9±1.7	70.8±1.5	72.2±1.7	**77.8±1.2**
		ARI	14.1±1.2	58.6±2.5	13.8±0.7	59.2±2.5	66.2±2.2	68.5±2.2	**74.5±1.8**
	JSD+Flip	ACC	17.8±1.5	69.1±7.1	10.0±0.0	37.2±2.9	60.8±4.7	58.1±4.7	**80.6±2.0**
		NMI	9.7±1.8	69.3±3.5	0.0±0.0	23.7±2.5	43.2±3.0	40.6±3.7	**69.6±1.7**
		ARI	3.9±0.9	57.6±6.2	0.0±0.0	15.9±2.8	37.0±3.9	34.0±4.5	**64.6±2.6**
RCV1-10	Dot+Flip	ACC	29.9±0.1	30.4±2.5	31.6±2.6	46.7±4.6	38.6±2.3	39.0±1.1	**49.1±1.7**
		NMI	0.3±0.4	16.7±1.1	3.2±3.9	36.6±2.4	30.8±1.3	34.1±1.5	**53.3±1.4**
		ARI	0.0±0.0	8.8±1.2	2.5±3.4	31.2±5.6	22.9±1.0	25.4±1.2	**39.5±2.4**
	JSD+Uniform	ACC	29.9±0.1	45.3±3.8	37.6±4.1	55.6±2.7	46.0±3.7	48.6±2.9	**56.4±3.8**
		NMI	0.8±1.4	29.0±3.5	13.3±1.2	53.8±0.8	47.6±0.7	51.4±1.3	**61.3±1.5**
		ARI	-0.1±0.1	25.0±6.3	10.2±1.9	44.8±1.4	35.7±2.0	38.4±2.0	**48.3±3.8**
	JSD+Flip	ACC	29.9±0.0	30.9±2.3	30.9±2.3	44.9±4.8	37.9±2.9	39.5±2.3	**49.8±1.9**
		NMI	0.0±0.0	17.4±1.4	1.7±3.4	37.2±1.7	31.5±1.8	35.3±1.8	**53.8±1.7**
		ARI	0.0±0.0	9.2±0.5	1.2±2.6	29.9±4.4	23.3±1.6	25.3±1.6	**39.9±2.6**

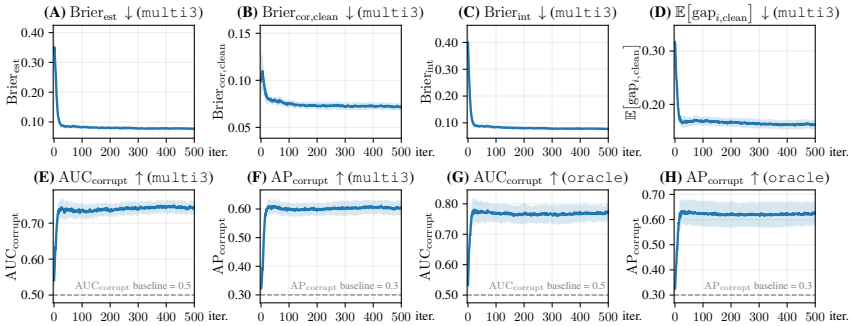


Figure 8: CIFAR100-20 held-out diagnostic evolution under `multi3` with corruption rate 0.3 (mean±std over 5 runs). Same panel definitions as in Fig. 3.

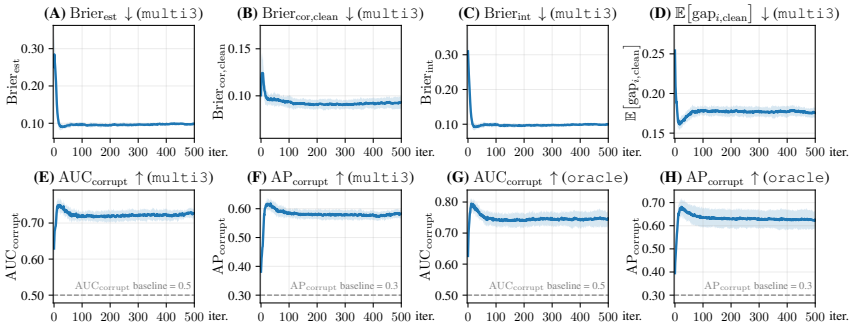


Figure 9: CIFAR10 held-out diagnostic evolution under `multi3` with corruption rate 0.3 (mean±std over 5 runs). Same panel definitions as in Fig. 3.

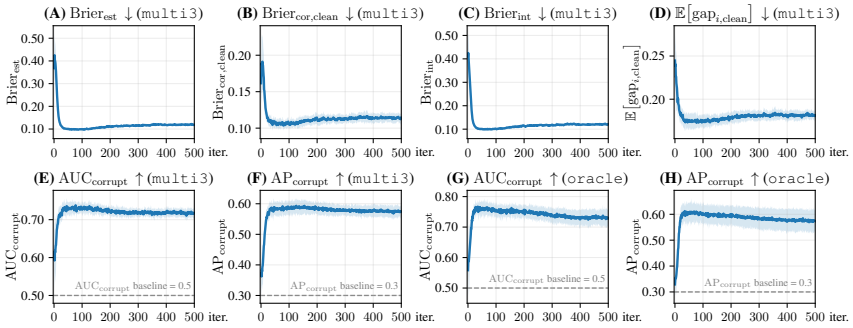


Figure 10: FMNIST held-out diagnostic evolution under `multi3` with corruption rate 0.3 (mean±std over 5 runs). Same panel definitions as in Fig. 3.

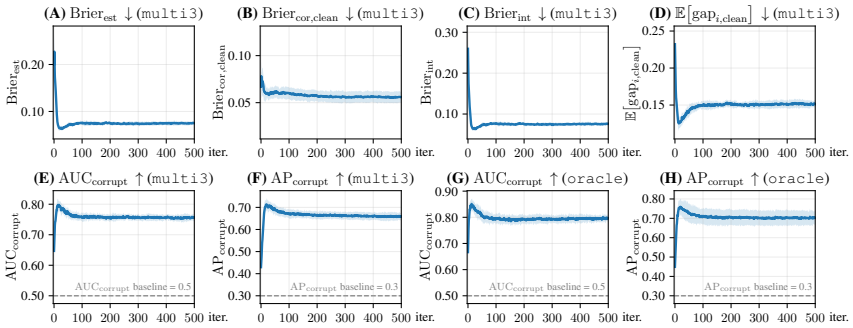


Figure 11: ImageNet10 held-out diagnostic evolution under `multi3` with corruption rate 0.3 (mean±std over 5 runs). Same panel definitions as in Fig. 3.

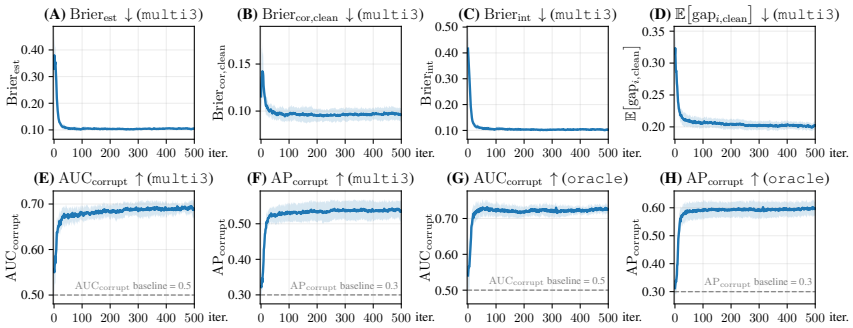


Figure 12: MNIST held-out diagnostic evolution under `multi3` with corruption rate 0.3 (mean±std over 5 runs). Same panel definitions as in Fig. 3.

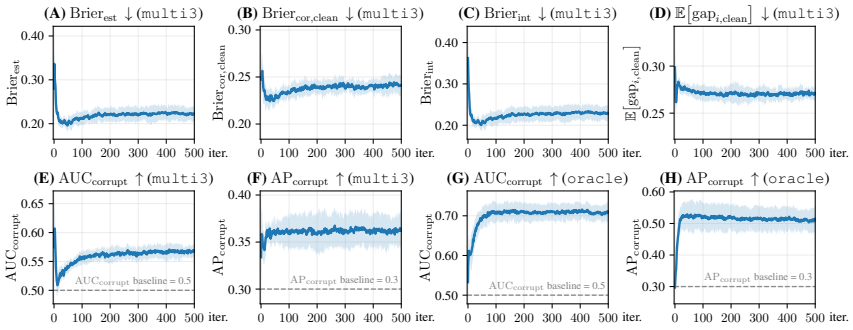


Figure 13: Reuters held-out diagnostic evolution under `multi3` with corruption rate 0.3 (mean±std over 5 runs). Same panel definitions as in Fig. 3.

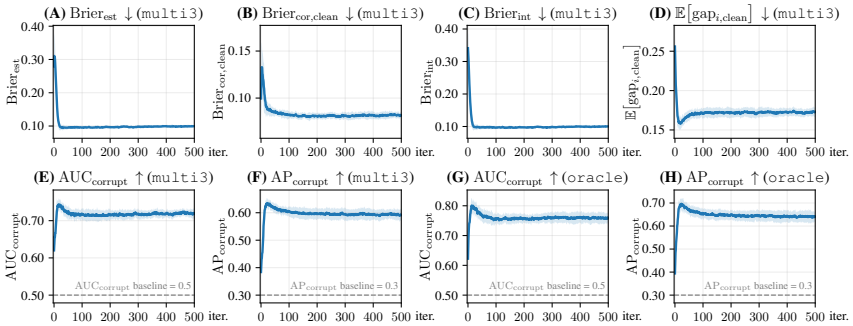


Figure 14: STL10 held-out diagnostic evolution under `multi3` with corruption rate 0.3 (mean±std over 5 runs). Same panel definitions as in Fig. 3.

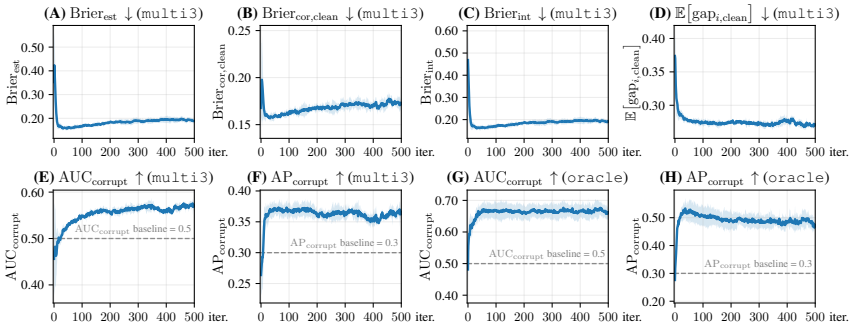


Figure 15: RCV1-10 held-out diagnostic evolution under `multi3` with corruption rate 0.3 (mean±std over 5 runs). Same panel definitions as in Fig. 3.

Let $\mathcal{I}_{\text{clean}} = \{i \in \mathcal{C}^{\text{diag}} : c_i = 0\}$ be the uncorrupted held-out subset, and define the residual discrepancy after correction as

$$\text{gap}_i = \left|\text{softclip}_\xi(\hat{y}_i^{\text{oof}}) - y_i^{\text{cor}}\right|.$$

For any relation score $v_i \in [0,1]$ and index set $\mathcal{I}$, we use

$$\text{Brier}(v;\mathcal{I}) = \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}(v_i - o_i^\star)^2.$$

We report $\text{Brier}_{\text{est}} = \text{Brier}(\hat{y}^{\text{oof}};\mathcal{C}^{\text{diag}})$, $\text{Brier}_{\text{cor,clean}} = \text{Brier}(y^{\text{cor}};\mathcal{I}_{\text{clean}})$, $\text{Brier}_{\text{int}} = \text{Brier}(y^{\text{int}};\mathcal{C}^{\text{diag}})$, and $\mathbb{E}[\text{gap}_{i,\text{clean}}] = |\mathcal{I}_{\text{clean}}|^{-1}\sum_{i\in\mathcal{I}_{\text{clean}}}\text{gap}_i$. For reliability-aware screening, gap_i is used to rank held-out constraints, and we report $\text{AUC}_{\text{corrupt}}$ and $\text{AP}_{\text{corrupt}}$ against the known corruption indicator c_i. These scores measure alignment with corruption rather than pure corruption estimation, since the reliability signal may also reflect difficult clean pairs, residual epistemic variation, and imperfect estimation/correction; nevertheless, corrupted records are expected to rank higher on average. As a reference, we also replace the `multi3` expert-generated judgments by $o_i^\star$ before applying the same 3k budget and corruption rate 0.3, thereby removing expert epistemic uncertainty from the clean supervision.

Evolution of internal estimates. The full diagnostic curves are shown in Figs. 8 to 15. Panels (A–C) track the Estimator, Corrector, and Integrator relation scores through their Brier errors against the external hard co-membership reference $o_i^\star$. Across datasets, these errors typically decrease quickly in the early stage and then become relatively stable, with some late-stage fluctuation. This pattern is clearer on CIFAR100-20, CIFAR10, ImageNet10, MNIST, and STL10, while FMNIST and Reuters show more rebound after the initial improvement, and RCV1-10 is the most unstable case. Because these Brier scores use $o_i^\star$ rather than the latent aleatoric relation $R^\star_{a_i b_i}$, small non-monotone changes should be read as diagnostic behavior rather than as direct evidence about latent-target recovery. Panel (D) gives a complementary correction-side view: the clean residual discrepancy usually drops early, indicating that corrected relations become more aligned with estimator beliefs on uncorrupted held-out constraints. Since the Estimator and Corrector co-evolve, this quantity should be interpreted together with (A–C), as evidence that the correction stage follows the improving estimator signal.

Screening against injected corruption. Panels (E,F) evaluate whether the reliability signal aligns with the injected corruption indicator. Under `multi3`, $\text{AUC}_{\text{corrupt}}$ and $\text{AP}_{\text{corrupt}}$ are generally above their chance baselines, so corrupted records tend to be ranked higher by the screening signal. The alignment is clearer on most image datasets, especially CIFAR100-20, CIFAR10, ImageNet10, MNIST, and STL10, but weaker or noisier on Reuters and particularly RCV1-10. This limitation reflects the heuristic nature of the screening signal: unreliability may arise not only from stochastic corruption, but also from expert-side epistemic variation, imperfect estimation/correction, and partial self-confirmation during training. The oracle-reference curves in panels (G,H) support this interpretation, as they usually track corruption more closely than the `multi3` curves when the clean supervision is generated from the hard oracle relation before corruption.

Summary. Overall, the diagnostics provide consistent qualitative evidence for the intended behavior of ECI-PP, although the curves are not uniformly monotone across datasets. The internal estimates improve mainly in the early stage, the correction discrepancy is reduced, and the screening signal remains meaningfully associated with injected corruption. RCV1-10 is the most challenging case, consistent with the broader experimental picture under severe class imbalance, where improvements in clustering structure are less uniformly reflected by the held-out diagnostic curves. Early stabilization further suggests that the ECI components often reach useful internal agreement before the fixed training horizon; however, we keep this schedule here and leave principled early stopping for future work. These diagnostics therefore provide practical held-out evidence for ECI-PP's intended behavior, while remaining proxy measurements rather than exact tests of latent aleatoric recovery or pure corruption detection.

D.7 Sensitivity analysis

We further study the sensitivity of the main ECI-PP settings around the default configuration specified in Section 5.1. All sensitivity runs use the default single-expert supervision condition, i.e., `lv0.01`

expert quality, corruption rate 0.3, and 9k constraints. To assess sensitivity to each factor, we use random initialization without unsupervised pretraining and vary one factor while keeping the others fixed. Figs. 16 to 23 report the corresponding test ACC, NMI, and ARI.

Estimator choices. Figs. 16 and 17 examine the Estimator-side choices. For the number of folds, increasing K from 2 to 20 gives mild gains on several datasets, such as MNIST, FMNIST, CIFAR10, and ImageNet10, but the curves are mostly flat and the improvement is not proportional to the extra cross-fitting cost. This supports using $K = 5$ as a practical default: it avoids the weakest cross-fitting setting while keeping the main Estimator cost moderate. The information-based warm-up weight κ_i provides a small but consistent benefit. Enabling it is usually better or comparable across datasets, with visible gains on MNIST, FMNIST, Reuters, and CIFAR10, indicating that informative constraints are useful for stabilizing the early Estimator signal. This agrees with the main comparison in Table 1, where Weighted ProbPair generally improves over ProbPair by using the same information-based weighting principle. At the same time, the gains remain moderate, suggesting that ECI-PP benefits from this design without being overly dependent on it.

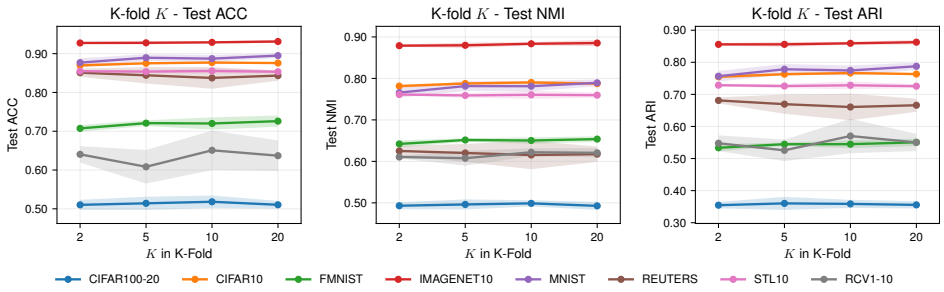


Figure 16: Sensitivity to the number of estimator folds K. Each panel reports test ACC, NMI, or ARI across datasets (mean±std over 5 runs).

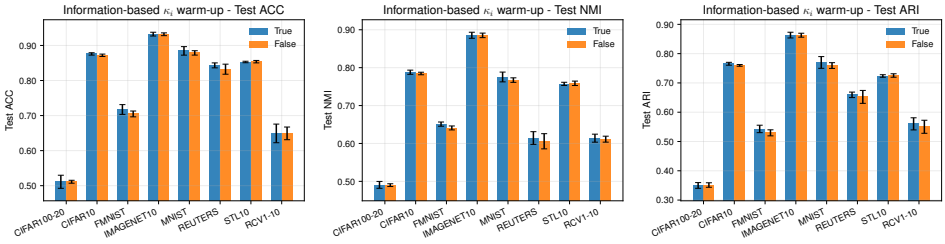


Figure 17: Effect of the information-based warm-up weight κ_i. Each panel reports test ACC, NMI, or ARI across datasets (mean±std over 5 runs).

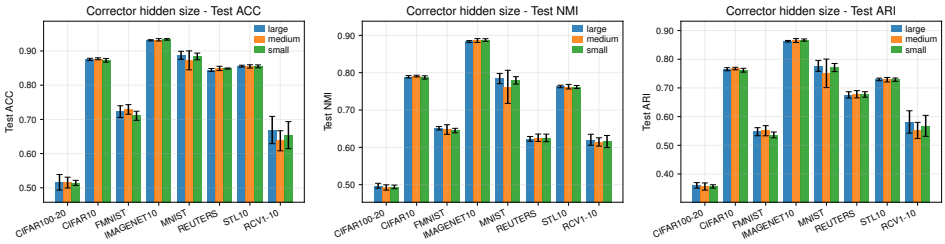


Figure 18: Sensitivity to Corrector capacity. Each panel reports test ACC, NMI, or ARI across datasets (mean±std over 5 runs).

Corrector choices. Figs. 18 and 19 study Corrector capacity and regularization. For capacity, we compare a *large* encoder-sized Corrector (500–500–2000), a *medium* classifier-sized Corrector

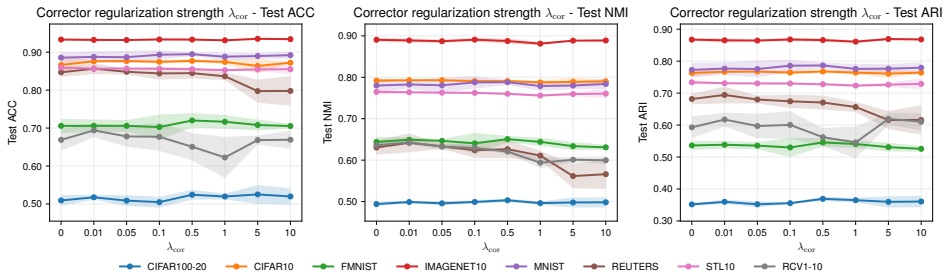


Figure 19: Sensitivity to the Corrector regularization strength λ_{cor}. Each panel reports test ACC, NMI, or ARI across datasets (mean$\pm$std over 5 runs).

(512–512), and the default *small* Corrector (64–16). The three sizes perform similarly on most datasets, and this supports the default use of a small Corrector: it is cheaper and also limits the risk of fitting arbitrary pair-specific deviations rather than structured expert-conditioned correction. The regularization strength λ_{cor} shows the same trade-off more directly. Compared with $\lambda_{\text{cor}} = 0$, mild to moderate regularization can help on datasets such as FMNIST, CIFAR10, CIFAR100-20, and RCV1-10, but overly strong regularization is harmful on some datasets, most clearly Reuters when λ_{cor} reaches 5 or 10. Thus, the Corrector should not be left completely unconstrained, but it also should not be forced too strongly toward zero correction. The default $\lambda_{\text{cor}} = 0.5$ lies in the stable middle range.

Reliability screening and boundary handling. Figs. 20 to 22 evaluate the parameters used after correction. The screening sharpness γ is robust over the tested range, especially for $\gamma \in [5, 20]$. Larger values are useful on some difficult datasets, notably RCV1-10 and also Reuters or MNIST in some metrics, indicating that stronger suppression of unreliable constraints can help when supervision is more fragile. The Bayesian-confidence scale n_0 is also stable overall. Larger n_0 improves RCV1-10 and slightly helps Reuters, suggesting that these datasets benefit from assigning more evidence to the refined supervision, whereas FMNIST does not benefit from the same increase. Thus, n_0 controls a real reliability trade-off, but the default value remains within a safe region. Finally, performance changes little as the soft-clipping sharpness ξ varies: most datasets change little from $\xi = 5$ to 50, with only mild fluctuations on RCV1-10 and a few small metric-specific changes. This supports soft clipping as a robust boundary-handling design for keeping corrected relations in a valid probability range without relying on a finely tuned clipping sharpness.

Reconstruction strength. Fig. 23 shows a noticeable dataset-dependent trade-off for λ_{rec}. The default value $\lambda_{\text{rec}} = 0.02$ follows SpherePair [3] and remains a reliable setting, but ECI-PP often benefits from moderately stronger reconstruction. For example, MNIST improves steadily as λ_{rec} increases, while Reuters and RCV1-10 prefer a moderate range around 0.05–0.2 before performance drops at larger values. In contrast, FMNIST and CIFAR10 do not benefit from overly large reconstruction weights, and ImageNet10 is almost unchanged. This suggests that reconstruction provides useful supervision-independent structure for out-of-fold estimation and final integration, but too much reconstruction can compete with clustering-oriented pairwise learning on some datasets. When validation-based tuning is available, λ_{rec} is therefore a meaningful parameter to adjust; without such tuning, the shared default remains a fair and stable compromise.

Overall, the sensitivity results support the use of the default ECI-PP configuration in the main experiments. Estimator, Corrector, screening, and soft-clipping choices are stable over broad ranges, while the more dataset-dependent reconstruction weight changes performance gradually rather than causing abrupt failure. The results therefore indicate that ECI-PP is not hyperparameter-fragile, even though dataset-specific validation could still improve individual settings when such tuning is allowed.

E Learning efficiency

Overall training time. Appendix B.2 analyzes the computational complexity of ECI-PP; here, based on the implementation in Appendix C.5, we report empirical wall-clock measurements for the

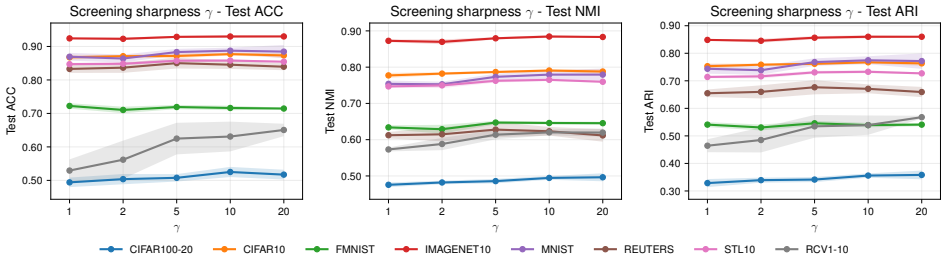


Figure 20: Sensitivity to the reliability-screening sharpness γ. Each panel reports test ACC, NMI, or ARI across datasets (mean$\pm$std over 5 runs).

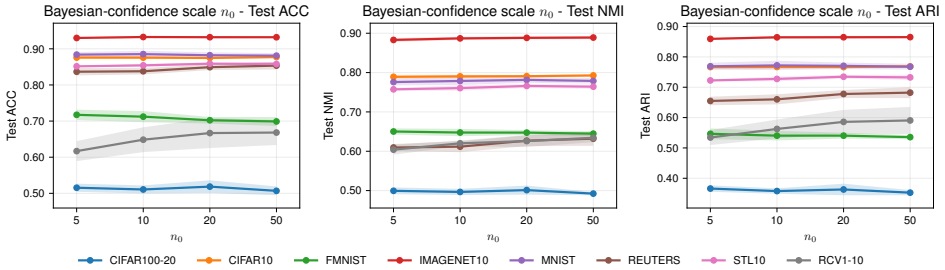


Figure 21: Sensitivity to the Bayesian-confidence scale n_0. Each panel reports test ACC, NMI, or ARI across datasets (mean$\pm$std over 5 runs).

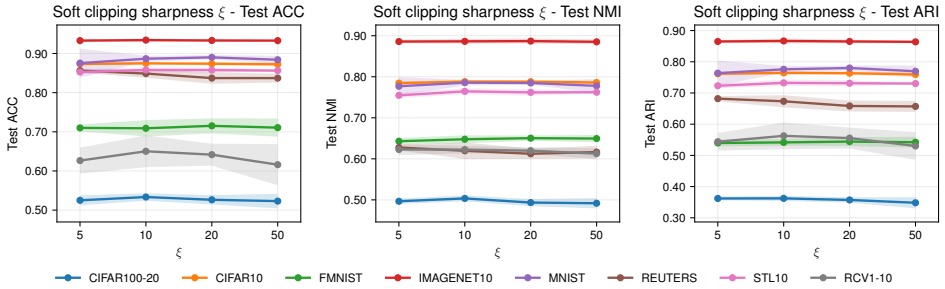


Figure 22: Sensitivity to the soft-clipping sharpness ξ. Each panel reports test ACC, NMI, or ARI across datasets (mean$\pm$std over 5 runs).

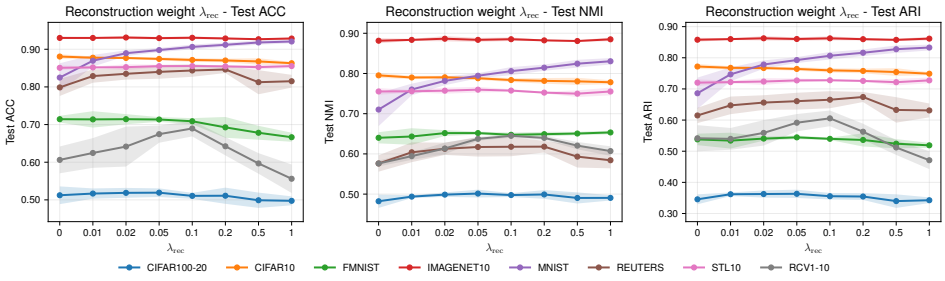


Figure 23: Sensitivity to the reconstruction weight λ_{rec}. Each panel reports test ACC, NMI, or ARI across datasets (mean$\pm$std over 5 runs).

Table 15: Overall wall-clock training time for different methods on eight datasets under the `multi3` multi-expert supervision setting with 9k constraints and corruption probability 0.3, measured on a single NVIDIA A100 40GB GPU. Methods marked with * require hyperparameter tuning, and their corresponding times are underlined.

	CIFAR100-20	CIFAR10	FMNIST	ImageNet10	MNIST	Reuters	STL10	RCV1-10
VanillaDCC	1m12s	1m10s	1m20s	0m45s	1m27s	0m37s	0m41s	1m51s
VolMaxDCC*	10m23s	45m21s	1h43m18s	6m20s	1h35m30s	5m00s	3m06s	2h39m25s
CIDEC	18m27s	19m26s	24m49s	5m03s	25m06s	5m53s	6m03s	1h07m26s
SpherePair	18m27s	17m58s	22m25s	5m09s	22m54s	6m05s	5m18s	1h01m54s
SpherePair (EA)*	3h53m10s	3h40m47s	4h30m35s	1h49m25s	4h33m48s	2h12m12s	1h51m31s	10h40m49s
ProbPair	18m19s	17m42s	22m16s	4m57s	22m41s	5m49s	5m02s	1h01m36s
Weighted ProbPair	18m24s	17m44s	22m23s	5m07s	22m56s	5m52s	5m06s	1h01m37s
Weighted ProbPair (EA)*	3h54m42s	3h34m20s	4h24m15s	1h44m36s	4h22m09s	2h01m15s	1h46m31s	10h32m39s
ECI-PP (Full)	29m48s	29m12s	34m01s	15m49s	34m39s	17m34s	15m59s	1h15m41s

complete training pipelines in Table 15. Each entry is measured under the `multi3` setting with 9k constraints and corruption probability 0.3, and covers the full practical training procedure: (i) for methods requiring unsupervised pretraining, the reported time includes the pretraining stage; (ii) for methods marked with *, it also includes three scans over the corresponding hyperparameter list and a final training run with the selected hyperparameter. When both pretraining and hyperparameter tuning are required, the timing includes pretraining on the training split excluding validation samples before tuning, followed by pretraining on the full training split before the final run.

Runtime comparison. The runtimes in Table 15 should be interpreted alongside the clustering results in Appendix D.1. VanillaDCC is fastest, but mainly because collapsed assignment learning triggers early stopping; its performance in Table 10 is far below the competitive methods. VolMaxDCC has variable cost, becoming expensive on FMNIST, MNIST, and RCV1-10 due to validation-based hyperparameter search. For methods with autoencoder pretraining, including CIDEC, SpherePair, ProbPair, Weighted ProbPair, and ECI-PP, overall costs remain on the same order, with pretraining as a shared pipeline component. ECI-PP adds moderate overhead from estimator cross-fitting and correction/integration, but this overhead is modest relative to its empirical gains; in more resource-limited settings, for example when unsupervised pretraining is unavailable, Appendix D.2 shows an even larger performance advantage. The expert-aware SpherePair and Weighted ProbPair extensions are most expensive because ensemble construction and hyperparameter tuning scale with the number of experts. In contrast, Appendix B.2 shows that ECI-PP incorporates expert identities through expert-conditioned correction without replicating the full training pipeline, so its dominant cost scales mainly with the constraint budget rather than the expert count.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main contributions and scope are stated in the abstract and Section 1. The formulation and identifiability analysis are given in Section 3, the method is developed in Section 4, and the experimental evidence is provided in Section 5 and Appendix D.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations and qualifications are discussed in Section 6, including the conditional population-level nature of the identifiability analysis and the surrogate nature of the Corrector and reliability signal.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The assumptions and theoretical results are stated in Section 3.2, and the derivations and complete proofs are provided in Appendix A.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The main experimental setup is described in Section 5.1, with full details on datasets, baselines, constraint generation, protocol, implementation, and hyperparameters in Appendix C. The algorithmic procedure is specified in Appendix B.1, and the public repository at https://github.com/ecipp/ECI-PP provides the ECI-PP implementation and usage instructions.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The datasets used in the experiments are publicly available and cited in Appendix C.1. Our source code and instructions for running ECI-PP are publicly available at https://github.com/ecipp/ECI-PP.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The key experimental settings are presented in Section 5.1, and full details are provided in Appendix C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Shaded regions in the robustness, held-out diagnostic, and sensitivity plots, as well as error bars in the corresponding bar charts, indicate mean$\pm$standard deviation over 5 runs. The main-text component ablation and CIFAR-10H table, as well as the

generator-misspecification tables also report mean±standard deviation over 5 runs. While the main comparison results in Table 1 omit standard deviations due to space constraints, the corresponding settings are covered by the robustness results in Appendix D.4, where run-to-run variability is reported.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The computing environment and GPU types are described in Appendix C.5, and training-time information is reported in Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We have strictly followed the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This paper aims to contribute to the advancement of the field of Machine Learning. While our work may have various societal implications, we do not believe any particular aspect requires explicit emphasis at this stage.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This work does not pose misuse risks. It focuses on fundamental research without deploying any models or assets that require safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Yes, we credited them in appropriate ways.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We release the ECI-PP source code at https://github.com/ecipp/ECI-PP, with documentation covering environment setup, data preparation, implementation structure, and runnable workflows.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The authors did not conduct crowdsourcing or collect new human-subject data. CIFAR-10H is an existing publicly released dataset of aggregate human annotations used only as secondary probabilistic supervision.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.