Semantic Homogeneity As Demonstration: Batch-Structured Semi-Supervised In-Context Learning for Natural Language Understanding

Cheng Chen[1,2], **YuanGang Pan**[1,2], **Ivor W. Tsang** [1,2,3]
[1] CFAR, Agency for Science, Technology and Research, Singapore
[2] IHPC, Agency for Science, Technology and Research, Singapore
[3] College of Computing and Data Science, Nanyang Technological University, Singapore
chengchen.martin@gmail.com, Pan_Yuangang@a-star.edu.sg, ivor_tsang@a-star.edu.sg

In-context learning (ICL) adapts large language models (LLMs) to downstream natural language understanding (NLU) tasks by prepending a small set of labeled demonstrations (input–label exemplars) to each query. While effective, this paradigm is costly and fragile: curating representative demonstrations and maintaining their relevance at scale is difficult, and inference cost grows with prompt length. This motivates a complementary question: *can LLMs benefit from in-context signals without using explicit exemplar pairs at all?* We propose **B**atch-Structured **I**mplicit **D**emonstration-Free **S**emi-supervised ICL (**BIDS**-ICL). Instead of providing exemplar pairs, we use a small labeled seed set only to induce *semantic structure*: we embed and cluster test-time inputs into *semantically homogeneous batches*, then prompt the LLM with the batch as context for predicting the labels of all items in that batch. In this non-exemplar regime, batch structure itself becomes an informative conditioning signal. We further consider a practical extension that arises naturally from the clustering pipeline: each item may be accompanied by a *pseudo-label hint* (e.g., an encoder-predicted intent), which can be noisy due to cluster misassignment and label propagation. Rather than asking whether pseudo-labels are universally good or bad, we ask a conditional question: *when is it useful to expose an LLM to pseudo-label hints under batch-structured prompting?* On the theory side, we provide a Bayesian aggregation perspective and draw on stagewise Plackett–Luce (PL) aggregation to explain why semantically homogeneous batches can improve prediction reliability. Empirically, across eight datasets and two LLMs, we observe a consistent competency–homogeneity interaction: semantic homogeneity acts as an orthogonal in-context signal that systematically modulates pseudo-label utility. When batches exhibit low homogeneity, pseudo-label hints often amplify clustering noise and may underperform unlabeled structured batching. When homogeneity is high, pseudo-label hints become more reliable, though their marginal benefit diminishes when structural coherence alone already induces strong label separation.

1. Introduction

Large language models (LLMs) are increasingly used for large-scale annotation, offering a scalable alternative to traditional crowdsourcing in NLU tasks such as intent classification, topic tagging, and domain detection [1]. However, LLM-based annotation is not a panacea: models remain vulnerable to hallucination, instability, and sampling-induced randomness [2–6]. To mitigate these issues, a broad range of prompting and decoding strategies have been proposed, including in-context (few-shot) prompting [7], instruction tuning [8], chain-of-thought prompting [9], and self-consistency prompting [10]. Among these, *in-context learning* (ICL) stands out for its simplicity and strong empirical performance. ICL is particularly attractive for annotation because it enables inference-time adaptation by conditioning on in-prompt instances without requiring parameter updates. In practice, ICL can steer an LLM toward a target labeling scheme or domain using

Third Conference on Parsimony and Learning (CPAL 2026).

lightweight prompting, thereby making large-scale annotation feasible. Yet this benefit comes at a cost: standard ICL relies on carefully curated labeled demonstrations, which are expensive to obtain and highly sensitive to selection. Moreover, demonstrations must be appended to each query, resulting in substantial inference overhead. Thus, the effectiveness of ICL depends critically on how demonstrations are constructed and organized at inference time. This dependence has motivated extensive efforts to understand why demonstrations help. Prior work examines (i) **how in-context examples shape latent task inference** and (ii) **whether explicit labels in demonstrations are necessary for performance gains**. Building on these questions, researchers have explored demonstration selection strategies, example ordering, hybrid sourcing, and robustness under noisy labels. Despite these advances, most studies implicitly assume that exemplar pairs—whether human-labeled or pseudo-generated—are the fundamental vehicle of supervision in ICL. However, emerging evidence complicates this view. Recent findings suggest that labels are not always essential for ICL performance [11, 12], indicating that models may rely more on contextual regularities than on explicit label tokens. If so, the supervisory signal in ICL may not reside solely in labeled exemplars, but also in the structure of the prompt itself. This observation raises a deeper question:

Can semantic coherence among unlabeled inputs act as an inference-time supervisory signal, enabling demonstration-free in-context learning through structural utility concentration rather than explicit exemplar pairs?

At the same time, prior experiments show that naïvely grouping samples by semantic similarity or maximizing in-batch diversity does not reliably outperform random grouping or single-sample prompting. This suggests that *structure alone is insufficient*: only certain forms of structural regularity may produce consistent gains. Consequently, the precise role of contextual organization in ICL remains unclear, particularly within batching-based frameworks. In parallel, a line of work on batching-based prompt packing studies how to process multiple inputs within a single LLM query, primarily to amortize inference costs. As deployment scales, this efficiency concern becomes acute: increasing demonstration size or retrieval scope inflates prompt length, latency, and token consumption, and may approach context limits. Batch prompting addresses this by jointly processing multiple inputs within a single query [13, 14]. Importantly, however, these approaches treat batching as a computational optimization rather than as a modeling mechanism. This distinction is crucial. Most batching-based approaches preserve the conventional exemplar-centric paradigm: exemplar blocks are shared or amortized across inputs, but the underlying supervision mechanism remains demonstration-based. More broadly, ICL research overwhelmingly studies explicit input–label or input–response pairs, even in so-called "demonstration-free" settings where pseudo-demonstrations are synthetically constructed [15, 16].Thus, batching has rarely been analyzed as a *structural conditioning regime* in its own right. As a result, there remains limited understanding of a distinct ICL mode—*non-exemplar, structure-driven conditioning*—in which the prompt consists solely of unlabeled inputs and supervision emerges from the internal coherence of the set itself.

Guided by this perspective, we treat *batch structure as an in-context signal* rather than merely a batching heuristic, and organize our investigation into three stages: Stage 1: Implicit demonstration-free prompting via clustered batches. We first study an implicit demonstration-free regime without explicit (input, label) exemplar pairs. Instead, we cluster test inputs into semantically homogeneous batches and present each batch as context. Here, structural coherence alone may prime a latent task mode and implicitly narrow the effective label space. Stage 2: When do pseudo-label hints help at a given coherence level? Having isolated structural effects, we next introduce pseudo-label hints (e.g., encoder-predicted intents). Because clustering and label propagation are imperfect, these hints are inherently noisy. We therefore ask: for a fixed level of batch coherence, when do pseudo-labels enhance prediction, and when do they amplify noise? Empirically, pseudo-labels often degrade performance under low homogeneity but become more beneficial as structural coherence increases, albeit with diminishing returns once structure already provides strong guidance. Stage 3: How LLM competency modulates label necessity across regimes. Finally, we examine how model competency interacts with structural coherence. Stronger LLMs extract richer signal from homogeneous batches and rely less on pseudo-label anchoring, whereas weaker LLMs depend more heavily

on clean hints and are more vulnerable to noise under low homogeneity. On the theoretical side, we provide a Bayesian aggregation perspective [17] using stagewise Plackett–Luce (PL) aggregation [18, 19] to explain *why* semantically homogeneous batches influence LLM behavior and how this effect interacts with label noise and model competency. In stagewise PL aggregation, two key parameter families govern performance: object-level scores (which determine ranking preferences) and worker uncertainty (which captures annotator reliability). We show how cluster-level statistics induced by our clustering pipeline can be mapped onto these latent factors. This mapping provides a principled account of when clustered batch prompts approximate robust k-ary aggregation and when pseudo-label hints instead amplify noise.

In summary, our main contributions are as follows:

- **Implicit demonstration-free batching.** We formalize a non-exemplar, batch-structured prompting regime for LLM-based annotation, in which semantically homogeneous batches—without explicit input–label exemplar pairs—serve as in-context signals for closed-set prediction.
- **Coherence-conditioned label necessity.** We characterize how pseudo-label hints help, plateau, or harm performance as a function of batch semantic homogeneity, yielding a two-dimensional regime view of pseudo-label utility.
- **Competency-modulated regimes.** We show that model competency shifts the homogeneity threshold at which pseudo-label hints become beneficial, explaining why label necessity varies across LLMs under identical batch structures.
- **Theoretical analysis via stagewise PL aggregation.** We map cluster-level statistics to latent parameters in stagewise PL aggregation, providing a principled explanation of when clustered batches approximate robust k-ary aggregation and how homogeneity governs robustness to noisy hints.
- **Empirical validation.** Extensive experiments across multiple LLMs and eight real-world NLU datasets validate the joint role of (i) batch homogeneity, (ii) pseudo-label reliability, and (iii) model competency in determining annotation quality.

2. Related Work

The Role of Labels in In-Context Learning. Existing literature presents conflicting views on the utility of labels in ICL. While standard few-shot prompting assumes that correct in-context labels are critical for performance, Min et al. [11] argue that models primarily exploit input distributional patterns, showing that replacing labels with random ones often leads to only minor degradation. In contrast, Kossen et al. [12] report high sensitivity to label correctness, suggesting that models do not treat all contextual information equivalently. These seemingly contradictory findings indicate that label utility may not be universal, but instead depends on previously unidentified moderating factors. In this work, we investigate whether the *semantic structure of the batch* governs when labels dominate prediction and when input patterns alone suffice.

Example Selection and Semi-Supervised ICL. A large body of work studies how to optimize demonstration selection through similarity-based retrieval, diversity maximization, or ordering strategies. Related approaches explore semi-supervised and pseudo-demonstration methods, such as Z-ICL [16, 20], which retrieve unlabeled neighbors and assign synthetic labels to construct in-context examples. Self-adaptive prompting methods similarly generate pseudo-demonstrations from model outputs and prepend them to improve zero-shot performance [21, 22]. While these methods leverage unlabeled data effectively, they generally assume that improved retrieval quality or more accurate pseudo-labels are uniformly beneficial. In contrast, our work does not treat selection metrics as universally helpful; instead, we analyze how the *semantic coherence* of a batch conditions whether pseudo-label exposure helps, becomes redundant, or degrades performance.

Batched Inference and Prompt Packing. From a systems perspective, batching has primarily been studied as a computational optimization technique. Methods such as Batch Prompting [13, 23], CliqueParcel [24], and BatchLLM [25] aim to improve throughput, reduce token cost, or enhance cache reuse. However, these approaches retain the standard few-shot format based on explicit input–label pairs and use batching mainly to amortize exemplar overhead. Our approach differs fundamentally: we discard formatted demonstrations entirely and treat batch composition—constructed via clustering—as a *semantic variable*.

Robustness to Noisy Labels. Recent studies analyze the impact of label noise within explicit demonstrations [26–28], proposing mitigation strategies such as Local Perplexity Ranking. These works characterize noise as annotation error within labeled exemplars. In contrast, we identify a distinct noise pathway. Because our setting eliminates explicit exemplars, pseudo-labels function only as *hints* derived from clustering and label propagation. Noise therefore arises from *batch heterogeneity* rather than mislabeled annotations. This distinction enables us to characterize when clustering-induced structure amplifies error and when it provides reliable implicit supervision.

3. Problem Setting

3.1. Preliminaries

We denote $\mathcal{X}$ and $\mathcal{Y}$ as the input and label spaces, respectively. We assume that the unsupervised text corpus distributions for training and testing are defined as $\mathcal{D}_U = \{x_1, \ldots, x_N\}$ and $\mathcal{D}_T = \{x_1, \ldots, x_L\}$, respectively, where samples $x \in \mathcal{X} \subseteq \mathbb{R}^d$ are drawn independently and identically distributed (i.i.d.) from an unknown distribution P over $\mathcal{X} \times \mathcal{Y}$. Moreover, the distribution $\mathcal{D}_U$ can be clustered into k disjoint clusters $\{C_k \mid k = 1, \ldots, k\}$, such that each cluster is disjoint, satisfying the condition: $C_{i'} \cap C_i = \emptyset, \forall i' \neq i$, and the union of all clusters covers the full unsupervised set: $\mathcal{D}_U = \bigcup_{i=1}^{k} C_i$. Correspondingly, we use $\lambda_j \in \{1, 2, \ldots, k\}$ to denote the cluster annotation of sample x_j, where $x_j \in C_{\lambda_j}$. The complete cluster assignment is denoted by the vector $\boldsymbol{\lambda} = (\lambda_1; \lambda_2; \ldots; \lambda_m)$. In addition, a set of preference annotations is provided in our experimental study, defined as $\mathcal{Y} = \{\bar{y}_1, \bar{y}_2, \ldots, \bar{y}_k\}$. Each cluster C_j is associated with preference-labeled samples of the form: $C_j = \{(x_i, \bar{y}_j) \mid x_i \in H_j\}$, where $H_j \subseteq H$, and $H = \{(x_i, \bar{y}_i)\}_{i=1}^{s}$ is a small user-preference-aligned subset with $s = 5\% \cdot |\mathcal{D}_U|$. These user-preference clusters are also disjoint, satisfying: $C_{i'} \cap C_i = \emptyset, \quad \forall i' \neq i$, and cover the aligned preference set: $H = \bigcup_{i=1}^{k} C_i$. These user-preference samples serve as reference signals for user response alignment. **The difference is that the "context" is not exemplar pairs but a structured set of instances (plus optional hints).**

Definition 1 (Batch-Structured ICL (BS-ICL))**.** *Given a label set* $(\mathcal{Y})$ *and test inputs* $(x_i{}_{i=1}^{N})$*, BS-ICL groups inputs into semantically homogeneous batches* (B_b) *via encoding and clustering. An LLM then predicts labels for all items in a batch using a single prompt containing either* (i) *only the batch inputs* (*unlabeled BS-ICL*) *or* (ii) *the inputs augmented with pseudo-label hints* $(\tilde{y}_i)$ (*pseudo-labeled BS-ICL*)*. No explicit* (*input, label*) *exemplar pairs are provided.*

4. Batch-Structured Semi-Supervised In-Context Learning under a Plackett–Luce (PL) Perspective

4.1. Closed-set prediction as a PL-inspired choice model

We study closed-set prediction over a label set $\mathcal{Y}$ with $|\mathcal{Y}| = K$. For an input query x, prediction selects a single label $y \in \mathcal{Y}$ under prompt context $\mathcal{C}$. Under a Plackett–Luce (PL) perspective [29, 30], single-label prediction can be viewed as the top-1 (degenerate) case of a ranking model. In this case, the PL likelihood reduces to the multinomial logit form:

$$P(y = r \mid x, \mathcal{C}) = \frac{\exp(\lambda_r(x; \mathcal{C}))}{\sum_{t=1}^{K} \exp(\lambda_t(x; \mathcal{C}))}, \tag{1}$$

where $\lambda_r(x;\mathcal{C})$ denotes the *context-modulated latent utility* of label r for query x. We interpret $\lambda_r(x;\mathcal{C})$ as capturing both intrinsic task relevance and context-induced utility shifts. In particular, batch semantic homogeneity affects the relative concentration of the utility vector $\{\lambda_r(x;\mathcal{C})\}_{r=1}^{K}$. When $\mathcal{C}$ is semantically coherent, utilities tend to concentrate toward a dominant latent mode, thereby sharpening the softmax distribution in (1).

4.1.1. Noisy pseudo-label hints and model competency

In batch-structured prompting, each query x_i may optionally be accompanied by a pseudo-label hint $\tilde{y}_i \in \mathcal{Y}$ produced by a clustering-and-propagation pipeline. Because both cluster assignment and label propagation are imperfect, $\tilde{y}_i$ is treated as a noisy auxiliary signal rather than a trusted label. We hypothesize that the utility of pseudo-label hints is jointly governed by two interacting factors:

1. **Batch semantic homogeneity**, which determines how informative the prompt context $\mathcal{C}$ is and how sharply it induces separation among effective label utilities $\{\lambda_r(x_i;\mathcal{C})\}_{r=1}^{K}$;
2. **Model competency**, which determines how reliably the LLM extracts signal from $\mathcal{C}$ and resists corruption induced by noisy pseudo-label hints.

Virtual-worker uncertainty (stagewise PL as an interpretive noise model). Following the intuition of stagewise noisy Plackett–Luce (PL) models [18], we model each proxy LLM as a *virtual worker* w equipped with a latent uncertainty profile $\boldsymbol{\eta}_w$. Intuitively, stronger models correspond to more concentrated uncertainty profiles (expert-like behavior), whereas weaker models correspond to more diffuse profiles (greater susceptibility to ambiguity and noise). We use multiple proxy LLMs of increasing capability to study how model competency shifts the utility of pseudo-label hints across coherence regimes.

Worker competency tiers (conceptual regimes). To provide a concrete interpretation of $\boldsymbol{\eta}_w$, we describe the following conceptual regimes:

- **Expert-like worker:** $\eta_w^{(1)} \approx 1$ and $\eta_w^{(r)} \approx 0$ for $r \geq 2$, i.e., the intended choice is selected with high probability.
- **Amateur-like worker:** probability mass concentrates on the top few alternatives. Let m denote the number of near-indistinguishable candidates (empirically often small, e.g., $m \approx 2$); then $\sum_{r=1}^{m} \eta_w^{(r)} \approx 1$.
- **Spammer-like worker:** near-uniform uncertainty, $\eta_w^{(1)} \approx \eta_w^{(2)} \approx \cdots \approx \eta_w^{(L)}$, carrying little discriminative information.
- **Malicious-like worker:** systematically biased away from the intended choice, e.g., $\eta_w^{(r)} > \eta_w^{(1)}$ for some $r > 1$.

Here L denotes the ranking depth used in the stagewise PL analogy (to avoid overloading $K = |\mathcal{Y}|$).

Stagewise noisy likelihood (interpretive form). To formalize the role of uncertainty, consider a latent top-L ranking $\rho = (\rho^{(1)}, \ldots, \rho^{(L)})$, where $\rho^{(i)}$ is the item at rank i. Let Y_i denote the latent (noise-free) stage-i choice and $\tilde{Y}_i$ the observed (possibly corrupted) stage-i choice under worker w.

Under the PL without-replacement assumption, the remaining candidate set at stage i is $\{\rho^{(i)}, \ldots, \rho^{(L)}\}$. Marginalizing over latent stage outcomes yields

$$P(\tilde{Y}_i = \rho^{(i)} \mid \lambda, \boldsymbol{\eta}_w) = \sum_{t=i}^{L} P(\tilde{Y}_i = \rho^{(i)} \mid Y_i = \rho^{(t)}, \boldsymbol{\eta}_w)\, P(Y_i = \rho^{(t)} \mid \lambda). \tag{2}$$

Under the stagewise PL model,

$$P(Y_i = \rho^{(t)} \mid \lambda) = \frac{\exp(\lambda_{\rho^{(t)}})}{\sum_{v=i}^{L} \exp(\lambda_{\rho^{(v)}})} \triangleq \Delta_{i,t}(\lambda).$$

To model worker-dependent corruption, we introduce a normalized stagewise uncertainty profile

$$\bar{\eta}_w^{(r;i)} = \frac{\eta_w^{(r)}}{\sum_{v=1}^{L-i+1} \eta_w^{(v)}}, \qquad r = 1, \ldots, L-i+1,$$

where $\eta_w^{(r)}$ represents the worker's relative reliability at displacement r. We assume stagewise displacement noise that depends only on rank offset and not item identity. Thus,

$$P(\tilde{Y}_i = \rho^{(i)} \mid Y_i = \rho^{(t)}, \boldsymbol{\eta}_w) = \bar{\eta}_w^{(t-i+1;i)}.$$

This yields

$$P(\tilde{Y}_i = \rho^{(i)} \mid \lambda, \boldsymbol{\eta}_w) = \sum_{t=i}^{L} \bar{\eta}_w^{(t-i+1;i)} \Delta_{i,t}(\lambda). \tag{3}$$

Scope relative to our method. Equations 2–3 serve as an interpretive analogy for how model competency and pseudo-label noise interact. Our task is closed-set top-1 prediction, and we do not optimize a stagewise PL likelihood for LLM decoding. Instead, this formulation motivates the hypothesis that pseudo-label utility depends non-linearly on both batch coherence and model competency, reflecting the interaction between likelihood concentration and worker-induced corruption.

4.1.2. Utility surrogates from clustering (no re-training)

We do not fit a contextual PL model, nor do we retrain the LLM. Instead, we use a non-parametric *cluster-coherence surrogate* [31, 32] to quantify how strongly batch structure may sharpen the effective label-choice distribution. Let $\mathcal{S} : \mathcal{X} \to \mathbb{R}^d$ be a sentence encoder, and suppose x_i is assigned to cluster C_j. Define its embedding as

$$e_i = \mathcal{S}(x_i).$$

We then define the within-cluster neighborhood coherence score

$$\hat{c}(x_i, C_j) = \frac{1}{k_{\mathrm{NN}}} \sum_{e \in \text{Top-}k_{\mathrm{NN}}(C_j, e_i)} \cos(e_i, e), \tag{4}$$

where Top-$k_{\mathrm{NN}}(C_j, e_i)$ denotes the k_{NN} nearest neighbors of e_i within cluster C_j under cosine similarity. A larger value of $\hat{c}(x_i, C_j)$ indicates tighter local semantic structure around x_i. Semantically coherent batches reduce feature dispersion in prompt conditioning, thereby stabilizing the induced context $\mathcal{C}$ and concentrating the effective utility vector $\{\lambda_r(x_i; \mathcal{C})\}_{r=1}^K$. Under our PL-inspired interpretive lens, higher cluster coherence is hypothesized to increase utility margins

$$\lambda_{r^\star}(x_i; \mathcal{C}) - \lambda_r(x_i; \mathcal{C}), \qquad r \neq r^\star,$$

which corresponds to a more concentrated top-1 predictive distribution in (1). Equivalently, in the Bayesian view, stronger coherence induces a sharper worker-dependent prior over latent utilities $P(\lambda \mid \boldsymbol{\eta}_w)$, reducing ambiguity without explicit model re-training. This provides a practical surrogate linking clustering quality to reduced predictive uncertainty under batch-structured prompting.

Preference-Aligned Clustered Batch Prompting. To translate non-parametric cluster structure into a parametric LLM, we group inputs into class-pure batches $B_j \subseteq D_s$ (one batch per latent class C_j). Each batch defines a prompt context $\mathcal{C}_j$. A batch embedding $\phi(B_j)$ (e.g. mean pooling over sentence embeddings) summarizes its semantic structure. We interpret the batch context as inducing context-modulated utilities

$$\lambda_r(x; \mathcal{C}_j),$$

where $\mathcal{C}_j$ is constructed from B_j. Operationally, $\phi(B_j)$ influences the LLM's internal scoring, shifting the effective utility vector $\{\lambda_r(x; \mathcal{C}_j)\}_{r=1}^K$. Because each batch is semantically coherent, utilities tend to concentrate toward the class-consistent label, thereby sharpening the softmax distribution in (1). This reduces predictive variance and implicitly regularizes the effective likelihood.

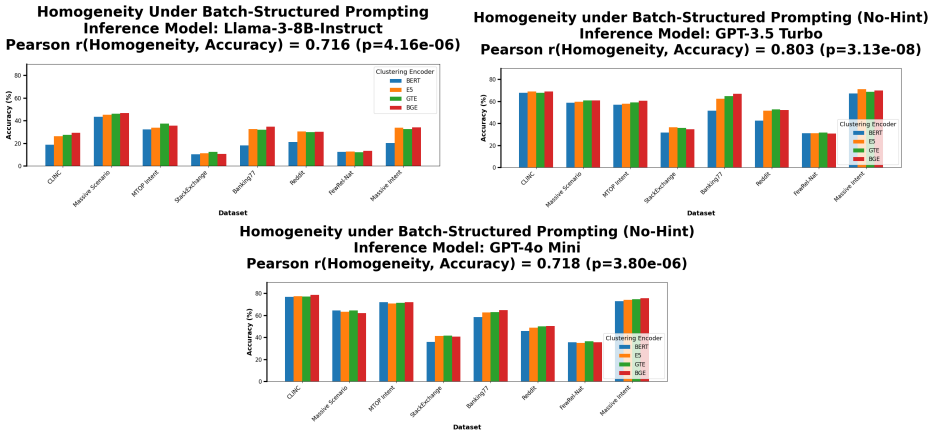

Figure 1: **Backbone ablation under batch-structured prompting.** Bars report No-Hint (unlabeled BS-ICL) accuracy across clustering encoders. Measured homogeneity (batch coherence) shows a strong positive correlation with performance across models: Llama-3-8B-Instruct ($r = 0.716$, $p = 4.16 \times 10^{-6}$), GPT-3.5 Turbo ($r = 0.803$, $p = 3.13 \times 10^{-8}$), and GPT-4o Mini ($r = 0.718$, $p = 3.80 \times 10^{-6}$, pooled). These results support the homogeneous batch hypothesis under structured prompting. **Detailed analysis is provided in Section E.**

Why clustering matters. If batches are noisy, heterogeneous, or class-imbalanced, the induced utility vector becomes diffuse, reducing concentration in (1) and degrading top-1 accuracy. Empirically, we observe that: 1. higher-quality embeddings produce tighter clusters; 2. tighter clusters induce more concentrated utility vectors; 3. more concentrated utilities improve prediction accuracy. Random or poorly clustered batches, as in conventional prompting, fail to provide this amplification effect.

From the Plackett–Luce model to LLM competency. Replacing a crowdsourced worker with a specialised proxy LLM allows us to treat the model as a virtual worker with uncertainty profile $\boldsymbol{\eta}_w$. For a highly competent worker w, $\eta_w^{(1)} \approx 1$ and $\eta_w^{(r)} \approx 0$ for $r \geq 2$. Under the stagewise noisy PL analogy,

$$P(\tilde{Y}_i = \rho^{(i)} \mid \lambda, \boldsymbol{\eta}_w) = \sum_{t=i}^{L} \bar{\eta}_w^{(t-i+1;i)} P(Y_i = \rho^{(t)} \mid \lambda).$$

When $\bar{\eta}_w^{(1;i)} \to 1$, the corruption term vanishes and

$$P(\tilde{Y}_i = \rho^{(i)} \mid \lambda, \boldsymbol{\eta}_w) \approx P(Y_i = \rho^{(i)} \mid \lambda).$$

In the top-1 degenerate case, this reduces to the softmax form in (1).

4.2. Posterior Interpretation

We use Bayes-style language to interpret the effect of context: a more homogeneous batch provides stronger evidence for a single latent mode, which corresponds to a more concentrated posterior over labels. We do not explicitly estimate $P(\lambda)$ or $P(\boldsymbol{\eta}_w)$. Instead, we use practical surrogates: (i) within-cluster similarity as a proxy for utility concentration ($\hat{\lambda}$), and (ii) LLM choice as a proxy for worker reliability (competency).

Posterior formulation. Given an observation $\tilde{Y} = \rho^{(i)}$ and worker w, Bayes' rule yields

$$P\big(\lambda, \boldsymbol{\eta}_w \mid \tilde{Y} = \rho^{(i)}\big) = \frac{P\big(\tilde{Y} = \rho^{(i)} \mid \lambda, \boldsymbol{\eta}_w\big)\, P(\lambda \mid \boldsymbol{\eta}_w)\, P(\boldsymbol{\eta}_w)}{P\big(\tilde{Y} = \rho^{(i)}\big)}. \tag{5}$$

Equivalently, the posterior is proportional to

$$P(\lambda, \boldsymbol{\eta}_w \mid \tilde{Y}) \propto \underbrace{P(\tilde{Y} \mid \lambda, \boldsymbol{\eta}_w)}_{\text{likelihood}} \quad \underbrace{P(\lambda \mid \boldsymbol{\eta}_w)}_{\text{worker-induced cluster prior}} \quad \underbrace{P(\boldsymbol{\eta}_w)}_{\text{worker prior}}. \tag{6}$$

Maximising the posterior is equivalent to maximising the product of the likelihood and the two priors. Under the worker-dependent formulation,

$$P(\lambda, \boldsymbol{\eta}_w \mid \tilde{Y}) \propto P(\tilde{Y} \mid \lambda, \boldsymbol{\eta}_w)\, P(\lambda \mid \boldsymbol{\eta}_w)\, P(\boldsymbol{\eta}_w).$$

In our interpretive setting:

- a *higher-quality clustering mechanism*, when induced by a competent worker, corresponds to a more concentrated worker-dependent prior over utility structures $P(\lambda \mid \boldsymbol{\eta}_w)$, reflecting stronger evidence for a single latent label mode;
- a *higher-competency worker* (the proxy LLM) corresponds to a more concentrated reliability prior $P(\boldsymbol{\eta}_w)$, reflecting reduced uncertainty in stage-wise selection.

Practical surrogates. Directly estimating $P(\tilde{Y} = \rho^{(i)} \mid \lambda, \boldsymbol{\eta}_w)$, $P(\lambda \mid \boldsymbol{\eta}_w)$, and $P(\boldsymbol{\eta}_w)$ is intractable at LLM scale. Instead, we introduce two operational surrogates:

- **Specialised proxy LLM (Worker).** A domain-adapted LLM acts as a virtual annotator whose uncertainty vector $\boldsymbol{\eta}_w$ is empirically concentrated on its top-ranked choice (i.e., $\bar{\eta}_w^1 \approx 1$), corresponding to an expert-like regime.
- **Cluster coherence.** The average top-k within-cluster cosine similarity serves as an empirical proxy for concentration in the worker-induced latent utility structure $P(\lambda \mid \boldsymbol{\eta}_w)$, without explicitly fitting a PL model.

Under this interpretation, improved clustering sharpens the worker-dependent prior over utilities, and higher worker competency reduces corruption in stage-wise selection. Together, these factors increase the alignment between induced context structure and true label ranking, leading to higher annotation accuracy without explicit Bayesian optimisation.

5. Experimental Design and Results

Our experimental design targets two questions that follow directly from our novelty claims: (i) **whether *batch structure alone* (without exemplar pairs) can serve as an effective in-context signal**, and (ii) **when *pseudo-label hints* help or harm under different levels of batch semantic homogeneity and model competency.** The details of baselines , batch configurations and dataset are in the Appendix Page. We use all-**MiniLM-L6-v2** as expert worker, a fine-tuned variant of MiniLM-L6-H384-uncased. The model is trained specifically to generate high-quality sentence embeddings, To empirically ground the virtual-worker interpretation, we compare proxy LLMs in amateur-like and expert-like competency regimes across eight NLU datasets. As shown in Table 1, the expert worker outperforms amateur workers on most datasets, with particularly large gains on **Banking77**, **Clinc**, and **Massive Intent**. This pattern is consistent with our stagewise PL-inspired view in which higher competency corresponds to a more concentrated latent uncertainty profile $\boldsymbol{\eta}_w$, enabling more reliable use of coherent batch structure and greater robustness to noisy pseudo-label hints. The weaker result on **Few Rel Nat** further suggests that competency effects are regime-dependent rather than uniformly monotone, making it well suited to tasks such as text clustering and sentence-similarity measurement. Amateur Worker. For the lower-quality baseline we use **bert-base-uncased**, a general-purpose English transformer that is pretrained but not task-specific. The expert worker shows higher accuracy.

Dataset/Homogeneity	Amateur	Expert
Clinc	$56.01_{\pm 0.05}\%$	$79.01_{\pm 1.08}\%$
Massive Scenario	$59.31_{\pm 2.90}\%$	$75.55_{\pm 1.76}\%$
Mtop Intent	$50.11_{\pm 2.16}\%$	$52.49_{\pm 2.52}\%$
StackExchange	$28.02_{\pm 0.53}\%$	$32.27_{\pm 0.65}\%$
Banking77	$38.93_{\pm 1.70}\%$	$73.93_{\pm 0.81}\%$
Reddit	$31.44_{\pm 0.55}\%$	$51.73_{\pm 0.62}\%$
Few Rel Nat	$42.04_{\pm 0.99}\%$	$35.35_{\pm 0.02}\%$
Massive Intent	$42.08_{\pm 0.50}\%$	$61.80_{\pm 1.04}\%$

Table 1: **Comparison of Amateur vs. Expert Workers.** Mean $\pm$ standard deviation (%) over two seeds.

Datasets / GPT-3.5	Batch Prompting Strategies			High-Quality Clustering	Advanced Single-Prompt Methods			
	Random Batch(Spammer)	Low-Fidelity Clustered Batch(Amateur)	High-Fidelity Clustered Batch(Expert)	High-Fidelity Clustered Batch(Expert) (Annotated)	CoT	FoT	Self-Consistency	Self-Refine
Clinc	$58.70_{\pm 0.23}\%$	$63.26_{\pm 0.77}\%$	$66.58_{\pm 3.36}\%$	$\mathbf{76.82}_{\pm 1.51}\%$	$45.46_{\pm 0.55}\%$	$46.66_{\pm 0.46}\%$	$76.56_{\pm 0.03}\%$	$64.39_{\pm 0.14}\%$
Massive_Scenario	$59.63_{\pm 0.31}\%$	$57.99_{\pm 0.83}\%$	$\mathbf{70.23}_{\pm 1.64}\%$	$60.85_{\pm 4.33}\%$	$52.01_{\pm 0.28}\%$	$56.06_{\pm 0.36}\%$	$63.44_{\pm 0.12}\%$	$47.70_{\pm 0.26}\%$
Mtop Intent	$59.12_{\pm 1.23}\%$	$62.96_{\pm 0.50}\%$	$64.95_{\pm 0.21}\%$	$55.12_{\pm 3.08}\%$	$58.41_{\pm 0.90}\%$	$\mathbf{59.64}_{\pm 0.73}\%$	$68.00_{\pm 0.26}\%$	$39.99_{\pm 0.08}\%$
StackExchange	$25.30_{\pm 0.32}\%$	$28.76_{\pm 0.52}\%$	$30.10_{\pm 0.10}\%$	$30.92_{\pm 2.21}\%$	$9.71_{\pm 0.34}\%$	$13.50_{\pm 0.19}\%$	$\mathbf{37.18}_{\pm 0.70}\%$	$21.21_{\pm 0.76}\%$
Banking77	$45.93_{\pm 0.16}\%$	$46.83_{\pm 1.17}\%$	$65.12_{\pm 0.30}\%$	$\mathbf{75.39}_{\pm 0.32}\%$	$27.24_{\pm 0.05}\%$	$32.34_{\pm 0.28}\%$	$56.10_{\pm 0.05}\%$	$36.74_{\pm 0.07}\%$
Reddit	$29.82_{\pm 1.11}\%$	$47.11_{\pm 1.52}\%$	$51.12_{\pm 1.27}\%$	$\mathbf{51.64}_{\pm 0.18}\%$	$22.69_{\pm 0.51}\%$	$27.52_{\pm 0.75}\%$	$41.15_{\pm 0.26}\%$	$26.88_{\pm 0.51}\%$
Few Rel Nat	$31.72_{\pm 1.77}\%$	$33.30_{\pm 2.02}\%$	$32.87_{\pm 1.72}\%$	$\mathbf{37.37}_{\pm 0.13}\%$	$18.36_{\pm 0.14}\%$	$17.34_{\pm 0.41}\%$	$27.52_{\pm 0.03}\%$	$15.68_{\pm 0.52}\%$
Massive Intent	$68.39_{\pm 1.38}\%$	$71.52_{\pm 2.00}\%$	$71.52_{\pm 0.95}\%$	$64.54_{\pm 0.02}\%$	$52.52_{\pm 1.33}\%$	$55.89_{\pm 0.36}\%$	$\mathbf{74.88}_{\pm 0.36}\%$	$55.12_{\pm 0.26}\%$

Table 2: **Batch prompting versus single prompting on GPT-3.5-Turbo .** Columns 2–4 demonstrate that *clustered* batches consistently improve or at least do not degrade accuracy across datasets, highlighting the value of incorporating preference-aligned clustering even with noisy preferences. Column 5 shows that when the same high-quality clusters are annotated, performance surpasses every single-prompt baseline (Columns 6–9), including CoT, FoT, Self-Consistency, and Self-Refine.

Datasets (Llama 3-8 Instruct)	Batch Prompting Strategies			High-Quality Clustering	Advanced Single-Prompt Methods			
	Random Batch(Spammer)	Low-Fidelity Clustered Batch(Amateur)	High-Fidelity Clustered Batch(Expert)	High-Fidelity Clustered Batch(Expert) (Annotated)	CoT	FoT	Self-Consistency	Self-Refine
Clinc	$25.57_{\pm 1.43}\%$	$33.96_{\pm 0.67}\%$	$32.49_{\pm 6.73}\%$	$\mathbf{69.40}_{\pm 7.28}\%$	$31.07_{\pm 0.21}\%$	$38.08_{\pm 0.99}\%$	$52.53_{\pm 0.27}\%$	$48.02_{\pm 1.07}\%$
Massive_Scenario	$41.56_{\pm 0.13}\%$	$44.42_{\pm 0.03}\%$	$43.52_{\pm 1.85}\%$	$\mathbf{66.74}_{\pm 0.98}\%$	$44.29_{\pm 1.26}\%$	$43.10_{\pm 1.10}\%$	$58.11_{\pm 0.15}\%$	$54.65_{\pm 1.29}\%$
Mtop Intent	$27.59_{\pm 0.48}\%$	$34.67_{\pm 1.27}\%$	$34.17_{\pm 6.70}\%$	$48.23_{\pm 0.25}\%$	$53.66_{\pm 0.03}\%$	$61.19_{\pm 0.07}\%$	$\mathbf{68.18}_{\pm 0.20}\%$	$39.93_{\pm 0.26}\%$
StackExchange	$10.46_{\pm 0.44}\%$	$13.84_{\pm 0.55}\%$	$11.02_{\pm 2.78}\%$	$\mathbf{26.26}_{\pm 2.16}\%$	$15.05_{\pm 1.58}\%$	$16.04_{\pm 1.38}\%$	$5.04_{\pm 0.21}\%$	$21.26_{\pm 0.76}\%$
Banking77	$12.34_{\pm 0.75}\%$	$21.27_{\pm 0.55}\%$	$33.06_{\pm 1.92}\%$	$\mathbf{69.66}_{\pm 1.74}\%$	$27.24_{\pm 0.05}\%$	$32.53_{\pm 0.49}\%$	$56.07_{\pm 0.05}\%$	$36.69_{\pm 0.07}\%$
Reddit	$16.65_{\pm 0.29}\%$	$26.29_{\pm 1.45}\%$	$36.31_{\pm 0.97}\%$	$\mathbf{46.00}_{\pm 2.51}\%$	$16.65_{\pm 0.29}\%$	$26.29_{\pm 1.45}\%$	$40.34_{\pm 0.92}\%$	$40.30_{\pm 2.09}\%$
Few Rel Nat	$8.56_{\pm 0.35}\%$	$13.17_{\pm 0.11}\%$	$14.25_{\pm 0.36}\%$	$\mathbf{31.80}_{\pm 0.34}\%$	$15.13_{\pm 0.14}\%$	$18.66_{\pm 0.26}\%$	$19.84_{\pm 0.17}\%$	$19.41_{\pm 0.24}\%$
Massive Intent	$17.57_{\pm 0.35}\%$	$31.05_{\pm 0.49}\%$	$45.41_{\pm 0.06}\%$	$56.03_{\pm 0.08}\%$	$35.02_{\pm 0.76}\%$	$43.05_{\pm 0.40}\%$	$\mathbf{74.63}_{\pm 0.19}\%$	$54.93_{\pm 0.36}\%$

Table 3: **Batch prompting versus single prompting on Lama 3-8 Instruct** Columns 2–4 demonstrate that *clustered* batches consistently improve or at least do not degrade accuracy across datasets, highlighting the value of incorporating preference-aligned clustering even with noisy preferences. Column 5 shows that when the same high-quality clusters are annotated, performance surpasses every single-prompt baseline (Columns 6–9), including CoT, FoT, Self-Consistency, and Self-Refine. Preference-Aligned Clustered Batch from random to high.

5.0.1. Large Language Model and Benchmark Datasets Setup

We conducted our experimental on LLMs—GPT-3.5 Turbo, and Llama-8B Instruct and evaluated these across ten textual datasets. These datasets include Bank77 [33], CLINC, Go Emotion, MTOP, Massive (Intent) [34–36], StackExchange, Reddit [37], FewRel Nat, and FewNerd Nat [38]. Covering domains such as intent classification, topic modelling, and unsupervised intent discovery [39, 40], their annotation practices follow [41].Detailed statistics of the datasets are shown in Table 6.

5.1. Empirical Findings.

(1) **Structure is the dominant driver in the batched regime (even without exemplar pairs).** Across both Llama-3-8B and GPT-3.5, moving from random batches to preference-aligned clustered batches yields large and consistent gains, showing that batching is not merely an efficiency trick: the *composition of the set* provides a usable in-context signal. For example, on Llama-3-8B, CLINC improves $25.57\% \rightarrow 32.49\%$, Banking77 $12.34\% \rightarrow 33.06\%$, and Massive Intent $17.57\% \rightarrow 45.41\%$; on GPT-3.5, Reddit improves $29.82\% \rightarrow 51.12\%$ and Banking77 $45.93\% \rightarrow 65.12\%$ when switching to expert clustered batching. This directly supports our Stage 1 claim of *non-exemplar, structure-driven conditioning*.

(2) **Annotating high-quality clusters produces a step-change that can match or exceed stronger decoding.** When the same high-quality clusters are annotated, accuracy jumps sharply and becomes competitive with, and often higher than, advanced single-prompt methods. On Llama-3-8B, annotated clustering reaches 69.40% on CLINC (vs. 52.53% Self-Consistency), 66.74% on Massive_Scenario (vs. 58.11%), and 69.66% on Banking77 (vs. 56.07%). On GPT-3.5, annotated clustering matches Self-Consistency on CLINC (76.82% vs. 76.56%) and substantially exceeds it on Bank-

Table 4: **Homogeneity–Label-Utility Trade-off under Strong and Weak LLMs.** The table illustrates the interaction between semantic homogeneity and model capability. Detailed analysis is provided in Section D, with additional results for the strong LLM (Table 15) and weak LLM (Table 9) reported in Appendix A.

Low Homogeneity (*BERT-like structure*)	**High Homogeneity** (*MiniLM-like structure*)
Strong LLM (DeepSeek-chat)	
Structure-Insensitive Regime	**Structure-Sufficient Regime**
• Batch structure contains noise	• Clean structure provides implicit supervision
• Pseudo-label hints amplify errors	• Pseudo-label hints unnecessary/harmful
• Random batching remains competitive	• Demonstration-free batching sufficient
Best: Random or Structured (No Hints)	**Best:** Structured (No Hints, MiniLM)
Examples: FewRel-Nat, StackExchange	**Examples:** Banking77, CLINC, Massive-Intent, Reddit
Weak LLM (GPT-4o Mini)	
Capability-Limited Regime	**Hint-Dependent Regime**
• Model cannot reliably exploit structure	• Structure alone insufficient
• Neither structure nor hints help consistently	• Pseudo-label hints provide external supervision
• Defaults to zero-shot/random batching	• Hints compensate for limited reasoning
Best: Zero-shot or Random	**Best:** Structured (With Hints)
Examples: FewRel-Nat, CLINC, MTOP	**Examples:** Reddit, Banking77, Massive-Scenario

ing77 (75.39% vs. 56.10%) and Reddit (51.64% vs. 41.15%). These results show that *improving in-context structure* can be as important as (or more important than) heavier decoding, consistent with "homogeneity is supervision or demonstration.

(3) Gains are regime-dependent and reflect coherence $\times$ competency effects. These tables also expose systematic boundaries: on some datasets, strong decoding remains dominant (e.g., Massive Intent where Self-Consistency is highest for both models), while on heterogeneous/low-clusterability datasets (e.g., StackExchange, FewRel-Nat) annotated clustering helps but may not fully close the gap to the best single-prompt baseline. Moreover, MTOP Intent illustrates that annotation can *hurt* in misaligned regimes (e.g., GPT-3.5: $64.95\% \rightarrow 55.12\%$), directly supporting our coherence-conditioned view that pseudo-label hints can amplify noise when structure is weak or mis-specified.

6. Conclusion

This paper studies *Batch-Structured Semi-Supervised In-Context Learning* for natural language understanding, focusing on an implicit, demonstration-free batching regime. We first examine whether semantically structured batches can serve as an in-context signal without explicit exemplar (input,label) pairs. We then analyze *coherence-conditioned label necessity*, characterizing when pseudo-label hints help, harm, or become redundant, and how these effects vary across LLMs with different competency levels. Most importantly, we connect the empirical gains of batch-structured prompting to a stage-wise Plackett–Luce (PL) aggregation perspective. We instantiate each PL utility score using the average top-k cosine similarity between an instance embedding and its nearest neighbors within the assigned cluster. Higher-purity clusters induce larger and more separable utilities, yielding a more concentrated stage-wise likelihood and, consequently, more accurate LLM predictions. In this formulation, the LLM acts as a surrogate annotator whose reliability is summarized by an uncertainty vector. When high-quality clusters are paired with a reliable model, the induced stage-wise likelihood becomes sharply peaked, improving robustness and accuracy. Extensive experiments support this interpretation and highlight a joint dependency: cluster purity and model competence interact multiplicatively. A strong model cannot fully compensate for heterogeneous or misclustered batches, and pure clusters alone are insufficient if the model behaves like a low-reliability worker. High annotation quality emerges most reliably when both batch structure and model competence are aligned.

References

[1] Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. Large language models for data annotation and synthesis: A survey, 2024. URL https://arxiv.org/abs/2402.13446.

[2] Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity. In Jong C. Park, Yuki Arase, Baotian Hu, Wei Lu, Derry Wijaya, Ayu Purwarianti, and Adila Alfa Krisnadhi, editors, *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics* (*Volume 1: Long Papers*), pages 675–718, Nusa Dua, Bali, November 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.ijcnlp-main.45. URL https://aclanthology.org/2023.ijcnlp-main.45/.

[3] Hussam Alkaissi and Samy I McFarlane. Artificial hallucinations in chatgpt: implications in scientific writing. *Cureus*, 15(2), 2023.

[4] Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=IkmD3fKBPQ.

[5] Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR, 2023.

[6] Aojun Zhou, Ke Wang, Zimu Lu, Weikang Shi, Sichun Luo, Zipeng Qin, Shaoqing Lu, Anya Jia, Linqi Song, Mingjie Zhan, et al. Solving challenging math word problems using gpt-4 code interpreter with code-based self-verification. In *The Twelfth International Conference on Learning Representations*, 2023.

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[8] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

[9] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

[10] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

[11] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022.

[12] Jannik Kossen, Neil Band, Clare Lyle, Aidan N. Gomez, Tom Rainforth, and Yarin Gal. In-context learning learns label relationships but is not conventional learning. In *Proceedings of the Twelfth International Conference on Learning Representations*, 2024.

[13] Zhoujun Cheng, Jungo Kasai, and Tao Yu. Batch prompting: Efficient inference with large language model apis. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 792–810, 2023.

[14] Jianzhe Lin, Maurice Diesendruck, Liang Du, and Robin Abraham. Batchprompt: Accomplish more with less. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Agyicd577r.

[15] Wei-Lin Chen, Cheng-Kuang Wu, Yun-Nung Chen, and Hsin-Hsi Chen. Self-icl: Zero-shot in-context learning with self-generated demonstrations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15651–15662, 2023.

[16] Xinxi Lyu, Sewon Min, Iz Beltagy, Luke Zettlemoyer, and Hannaneh Hajishirzi. Z-icl: Zero-shot in-context learning with pseudo-demonstrations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 2345–2360. Association for Computational Linguistics, 2023.

[17] Ke Deng, Simeng Han, Kate J Li, and Jun S Liu. Bayesian aggregation of order-based rank data. *Journal of the American Statistical Association*, 109(507):1023–1039, 2014.

[18] Yuangang Pan, Bo Han, and Ivor W Tsang. Stagewise learning for noisy k-ary preferences. *Machine Learning*, 107:1333–1361, 2018.

[19] Bo Han, Yuangang Pan, and Ivor W Tsang. Robust plackett–luce model for k-ary crowdsourced preferences. *Machine Learning*, 107:675–702, 2018.

[20] Cheng Chen and Ivor Tsang. Self-teaching prompting for multi-intent learning with limited supervision. In *The Second Tiny Papers Track at ICLR 2024*, 2024. URL https://openreview.net/forum?id=DeoamI1BFh.

[21] Xingchen Wan, Ruoxi Sun, Hanjun Dai, Sercan Arik, and Tomas Pfister. Better zero-shot reasoning with self-adaptive prompting. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3493–3514, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.216. URL https://aclanthology.org/2023.findings-acl.216/.

[22] Xingchen Wan, Ruoxi Sun, Hootan Nakhost, Hanjun Dai, Julian Eisenschlos, Sercan Arik, and Tomas Pfister. Universal self-adaptive prompting. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7437–7462, 2023.

[23] Cheng Chen, Bowen Xing, and Ivor W Tsang. Low-hanging fruit: Knowledge distillation from noisy teachers for open domain spoken language understanding. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 107–125, 2024.

[24] Jiayi Liu, Tinghan Yang, and Jennifer Neville. Cliqueparcel: An approach for batching LLM prompts that jointly optimizes efficiency and faithfulness. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2024.

[25] Zhen Zheng, Xin Ji, Taosong Fang, Fanghao Zhou, Chuanjie Liu, and Gang Peng. Batchllm: Optimizing large batched llm inference with global prefix sharing and throughput-oriented token batching. *arXiv preprint arXiv:2412.03594*, 2024.

[26] Chen Cheng, Xinzhi Yu, Haodong Wen, Jingsong Sun, Guanzhang Yue, Yihao Zhang, and Zeming Wei. Exploring the robustness of in-context learning with noisy labels. *arXiv preprint arXiv:2404.18191*, 2024.

[27] Hongfu Gao, Feipeng Zhang, Wenyu Jiang, Jun Shu, Feng Zheng, and Hongxin Wei. On the noise robustness of in-context learning for text generation. *arXiv preprint arXiv:2405.17264*, 2024.

[28] Cheng Chen, Atsushi Nitanda, and Ivor Tsang. Unlearning misalignment for personalized LLM adaptation via instance-response-dependent discrepancies. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=njE3swFBMc.

[29] Gerard Debreu. Individual choice behavior: A theoretical analysis, 1960.

[30] Robin L Plackett. The analysis of permutations. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 24(2):193–202, 1975.

[31] Cheng Chen, Haiyan Yin, and Ivor Tsang. Verification and co-alignment via heterogeneous consistency for preference-aligned LLM annotations. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=jugY302BAh.

[32] Cheng Chen, Haiyan Yin, and Ivor Tsang. Evaluating llms without oracle feedback: Agentic annotation evaluation through unsupervised consistency signals. *arXiv preprint arXiv:2509.08809*, 2025.

[33] Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. Efficient intent detection with dual sentence encoders. *arXiv preprint arXiv:2003.04807*, 2020.

[34] Stefan Larson, Anish Mahendran, Joseph J Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K Kummerfeld, Kevin Leach, Michael A Laurenzano, Lingjia Tang, et al. An evaluation dataset for intent classification and out-of-scope prediction. *arXiv preprint arXiv:1909.02027*, 2019.

[35] Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, Swetha Ranganath, Laurie Crist, Misha Britan, Wouter Leeuwis, Gokhan Tur, and Prem Natarajan. Massive: A 1m-example multilingual natural language understanding dataset with 51 typologically-diverse languages, 2022.

[36] Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. Mtop: A comprehensive multilingual task-oriented semantic parsing benchmark. *arXiv preprint arXiv:2008.09335*, 2020.

[37] Gregor Geigle, Nils Reimers, Andreas Rücklé, and Iryna Gurevych. Tweac: transformer with extendable qa agent classifiers. *arXiv preprint arXiv:2104.07081*, 2021.

[38] Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. *arXiv preprint arXiv:1810.10147*, 2018.

[39] Hanlei Zhang, Hua Xu, Ting-En Lin, and Rui Lyu. Discovering new intents with deep aligned clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14365–14373, 2021.

[40] Yuwei Zhang, Haode Zhang, Li-Ming Zhan, Xiao-Ming Wu, and Albert Lam. New intent discovery with pre-training and contrastive learning. *arXiv preprint arXiv:2205.12914*, 2022.

[41] Yuwei Zhang, Zihan Wang, and Jingbo Shang. Clusterllm: Large language models as a guide for text clustering. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

A. Appendix

A.1. Baselines

Random Batch (baseline). We prompt the LLMs—`GPT-3.5-Turbo` and `GPT-4o-mini`—with *random batches*, i.e. groups of samples drawn uniformly from the entire pool without regard to latent category membership. As a result, each batch may contain a mixture of categories.

Preference-Aligned Clustered Batch (proposed). We prompt the same LLMs with *preference-aligned clustered batches*, each of which contains only samples from a single latent category. These categories are obtained by clustering embeddings produced by *all-MiniLM-L6-v2* and *bert-base-uncased*, our high-competence proxy-worker model.

Preference-Aligned Clustered Batch (Annotated). To demonstrate that clustered samples can enhance LLM responses, we incorporate worker-assigned annotations into the prompt to guide the LLM toward better final outputs. This can be viewed as a finer-grained approximation of the utility score λ.

Prompt-based approaches with random batch on ChatGPT 3.5 turbo and 4o mini and single query on Llama 8B instruct

- **Self-Consistency** [**?**] aims to improve the response accuracy of large language models (LLMs) by selecting outputs that are consistent across multiple diverse reasoning paths.
- **Chain-of-Thought (CoT) prompting** [9] is step-by-step reasoning by demonstrating steps by step for a given query, helping LLMs generate more accurate response.
- **Few-Shot Thought Prompting** [7] adding a few examples into the prompt to help the model generate better responses.
- **Self-Refine** [**?**] is designed to enhance initial response through iteration of self-correction.

A.2. Input configurations (implicit, non-exemplar prompting).

We compare the following inference configurations, all of which contain *no* explicit (input,label) demonstrations:

1. **High-homogeneity clustered batches (Structure-only).** Test inputs are grouped into semantically homogeneous batches via encoder-based clustering, and the LLM predicts labels for all items in the batch from a single prompt that contains only the batch inputs.
2. **High-homogeneity clustered batches + pseudo-label hints (Structure + Hints).** The same clustered batches as above, but each item is accompanied by a pseudo-label hint produced by the clustering/propagation pipeline (and therefore subject to misclustering and propagation noise). This isolates *coherence-conditioned pseudo-label utility*.
3. **Low-homogeneity random batches.** Inputs are randomly grouped into batches of the same size, yielding low semantic homogeneity. This serves as a counterfactual to test whether gains stem from *structure* rather than batching itself.
4. **Single-instance prompting.** Each query is prompted independently (batch size $= 1$), providing a non-batched baseline for accuracy and variance.
5. **Imbalanced-cluster batches.** Batches are clustered but constructed from imbalanced seed-induced assignments (skewed class frequencies), to test robustness of structure-driven prompting under realistic distribution shift and cluster impurity.

A.3. Ablation: Preference-Aligned Clustered Under Imbalanced with different label budgets

To test the robustness of our clustering scheme under label imbalance, we performed an additional ablation using *Llama–3–8B–Instruct*. We varied the *label budget*—the proportion of preference sam-

Table 5: **Lama 3-8B Instruct**: Accuracy Comparison Across Datasets with 1%, 5%, and 10% Label Budgets (Mean ± Std, in %) under **Imbalanced Preference Samples**.

Dataset	Label Budget	Accuracy Metrics		
		Preference-Aligned Clustered Batch (%)	Preference-Aligned Clustered Batch(Annotated) (%)	Proxy LLM (Expert Worker) (%)
Clinc	1%	27.28 ± 0.06	54.39 ± 0.06	62.86 ± 0.94
	5%	34.01 ± 2.43	68.28 ± 1.34	78.98 ± 0.24
	10%	36.54 ± 1.46	71.79 ± 1.48	83.98 ± 1.49
Massive_Scenario	1%	45.15 ± 0.49	57.30 ± 2.10	60.73 ± 2.76
	5%	46.22 ± 0.29	66.80 ± 1.02	75.94 ± 1.16
	10%	46.36 ± 0.86	70.10 ± 0.57	78.50 ± 0.08
Mtop_Intent	1%	27.82 ± 1.31	44.93 ± 1.90	39.47 ± 3.83
	5%	30.11 ± 0.50	51.64 ± 1.85	49.66 ± 1.09
	10%	32.03 ± 0.57	57.37 ± 0.83	55.97 ± 2.30
StackExchange	1%	11.49 ± 0.70	21.11 ± 0.96	21.03 ± 1.49
	5%	12.27 ± 0.37	29.86 ± 0.48	31.44 ± 0.06
	10%	13.16 ± 0.28	33.20 ± 0.67	35.73 ± 0.38
Banking77	1%	31.49 ± 0.19	58.72 ± 0.83	62.03 ± 0.41
	5%	37.08 ± 0.06	69.71 ± 1.43	74.17 ± 0.76
	10%	40.39 ± 1.07	77.55 ± 1.25	81.38 ± 1.02
Reddit Reddit	1%	22.58 ± 0.22	38.25 ± 0.89	38.67 ± 1.03
	5%	28.88 ± 0.12	49.19 ± 0.84	50.23 ± 0.81
	10%	29.49 ± 1.60	52.34 ± 0.36	53.48 ± 0.39
Few_Rel_Nat	1%	11.45 ± 0.27	23.36 ± 0.12	24.70 ± 0.23
	5%	12.46 ± 0.25	31.46 ± 1.22	33.69 ± 0.84
	10%	12.91 ± 0.46	33.34 ± 0.35	36.10 ± 0.23
MASSIVE_INTENT	1%	31.61 ± 1.01	45.76 ± 0.39	49.34 ± 1.43
	5%	34.56 ± 0.44	54.14 ± 0.29	58.15 ± 0.22
	10%	35.92 ± 0.08	59.18 ± 0.49	63.23 ± 0.18

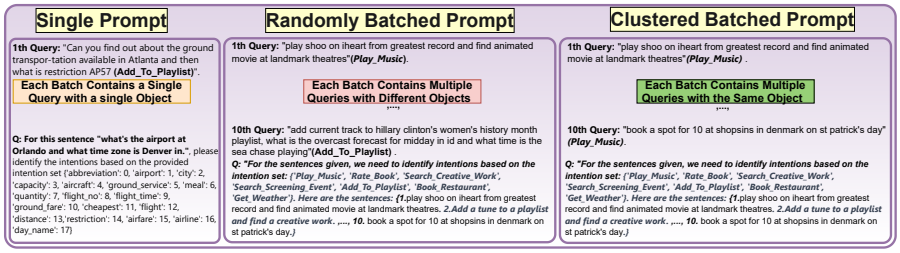

Figure 2: shows the format of Single Prompt, Random Batch Prompt, and Clustered Prompt

ples available for each class, while keeping the total number of samples fixed. As the label budget increases from 1% to 5% and 10%, clustering quality improves, and the specialised prox LLM accuracy increases accordingly on every dataset (e.g., Clinc 62.9% → 84.0%). These improvements show that cosine-based clustering remains robust even under imbalanced preference samples and clustering.

A.4. Dataset

We have show all the datasets in our study in the following Table 6.

Task	Name	# data (Testing)	# data (Training)	# classes
Intent	Bank77	3,080	10,003	77
	CLINC (I)	4,500	15,000	150
	MTOP (I)	4,386	15,638	102
	Massive (I)	2,974	11,510	59
Type	FewRel	4,480	40,320	64
Topic	StackEx	4,156	50,000	121
	Reddit	3,217	50,000	50
Domain	Massive Scenario	2,974	11,514	18

Table 6: Summary of Benchmark Datasets.

A.5. Prompting Format

The Prompting Format format is shown in Figure 2.

B. Additional Experiments

B.1. Llama3-8 Instruct

Batch-structured prompting vs. single-prompt baselines. Table 7 shows that our batch-structured prompting can outperform strong single-prompt baselines (CoT, FoT, Self-Consistency, Self-Refine), and that the gap depends on batch structure. In particular, under *high-quality clustering* (Cluster Quality column), the **With-Hint** variant consistently achieves large gains on several intent-heavy datasets (e.g., CLINC, Massive Scenario, Banking77, and Reddit), often exceeding the best single-prompt baseline by a wide margin. This supports our core thesis that *batch structure can serve as an effective in-context signal*: grouping semantically aligned items reduces ambiguity and stabilizes prediction, yielding improvements that cannot be obtained by stronger decoding alone. At the same time, the comparison also exposes regime boundaries: for datasets where clustering purity is intrinsically low or label space is highly heterogeneous (e.g., StackExchange and FewRel-Nat), the best single-prompt baseline may remain competitive, indicating that batch gains are not unconditional but arise when the induced batches form genuine latent clusters. Overall, these results show that **structured batch prompting can be a competitive (and often superior) alternative to single-prompt inference**, especially when batch homogeneity is high and pseudo-label hints are reliable.

B.2. ChatGPT 3.5 Turbo

Batch-structured prompting vs. single-prompt baselines. Table 8 compares our batch-structured ICL variants (No-Hint / With-Hint) against strong single-prompt inference baselines (CoT, FoT, Self-Consistency, Self-Refine). Two patterns are clear.

First, structured batches can match or surpass stronger decoding. On multiple intent-centric datasets where clustering yields *high semantic coherence* (high values in the Cluster Quality column; typically with E5/GTE/BGE), the **With-Hint** variant achieves the strongest overall performance and often exceeds the best single-prompt baseline. For example, on **CLINC**, **Banking77**, and **Reddit**, the With-Hint scores under modern backbones are substantially higher than CoT/FoT and are competitive with or higher than Self-Consistency/Self-Refine. This supports our central claim that *batch structure itself constitutes a usable in-context signal*: by presenting semantically aligned inputs together, the model can infer a shared latent task mode and reduce label-space ambiguity, producing gains that are not merely attributable to decoding heuristics.

Second, the advantage is regime-dependent and tracks coherence. When clustering purity is intrinsically limited or the dataset is highly heterogeneous (e.g., **StackExchange** and parts of **FewRel-Nat**), the best single-prompt baseline—often Self-Consistency—remains competitive. In these regimes, the batch context is less informative, and pseudo-label hints are more prone to misguidance; consequently, the marginal value of batching shrinks and the best results may still come from stronger single-prompt decoding. This boundary case is consistent with our broader thesis: *batch prompting is not universally beneficial*—its gains arise primarily when the induced batches form *genuine latent clusters*, i.e., high-homogeneity sets aligned with the underlying label distribution.

Overall, the table supports a coherent conclusion: **structured batch prompting is a competitive—and often superior—alternative to single-prompt inference**, particularly in the high-coherence regime, while single-prompt methods remain strong baselines when coherence is low.

Dataset	Backbone	Batch-Structured Prompting			Single-Prompt Baselines			
		No-Hint	With-Hint	Homogeneity Given Backbone	CoT	FoT	Self-Consistency	Self-Refine
CLINC	BERT	18.58	42.80	55.40	31.07	38.08	**52.53**	48.02
	E5	26.18	61.51	80.71				
	GTE	27.31	60.18	81.22				
	BGE	29.27	60.27	**83.13**				
Massive Scenario	BERT	43.31	53.67	58.44	44.29	43.10	**58.11**	54.65
	E5	45.16	57.73	69.91				
	GTE	45.93	62.78	73.60				
	BGE	46.57	62.58	**76.40**				
MTOP Intent	BERT	32.15	45.55	51.39	53.66	61.19	**68.18**	39.93
	E5	33.58	47.04	51.48				
	GTE	37.28	46.58	50.98				
	BGE	35.59	47.61	**56.57**				
StackExchange	BERT	10.18	26.59	27.41	15.05	16.04	5.04	**21.26**
	E5	11.21	36.93	38.67				
	GTE	12.22	38.93	**38.98**				
	BGE	10.68	36.55	37.13				
Banking77	BERT	17.99	34.68	40.13	27.24	32.53	**56.07**	36.69
	E5	32.53	63.08	71.56				
	GTE	31.95	67.18	75.62				
	BGE	34.55	68.70	**79.48**				
Reddit	BERT	21.17	32.36	32.05	16.65	26.29	**40.34**	40.30
	E5	30.34	50.23	51.63				
	GTE	29.87	52.60	**53.34**				
	BGE	30.15	50.73	51.51				
FewRel-Nat	BERT	12.21	34.96	**42.50**	15.13	18.66	**19.84**	19.41
	E5	12.77	35.31	41.23				
	GTE	11.99	31.23	38.37				
	BGE	13.33	32.99	37.68				
Massive Intent	BERT	20.07	40.85	42.80	35.02	43.05	**74.63**	54.93
	E5	33.62	58.14	61.26				
	GTE	32.41	60.05	63.21				
	BGE	34.10	67.05	**68.83**				

Table 7: **Backbone ablation under batch-structured semi-supervised prompting (Llama-3-8B Instruct). No-Hint/With-Hint** are clustered batches without/with pseudo-label hints. **Cluster Quality** is the batch homogeneity score. CoT/FoT/Self-Consistency/Self-Refine are single-prompt baselines reported per dataset (independent of backbone).

B.3. GPT-4o Mini

Backbone ablation with single-prompt baselines (GPT-4o Mini). Table 9 highlights three consistent patterns.

(1) Better backbones yield higher cluster quality and higher batch-prompt accuracy. Moving from BERT to modern encoders (E5/GTE/BGE) substantially increases the batch homogeneity score (Cluster Quality), and this is accompanied by large gains under both *No-Hint* and *With-Hint*. The strongest backbone is usually BGE, which achieves the best Cluster Quality on every dataset (e.g., CLINC: 83.13, Massive Scenario: 76.40, Banking77: 79.48).

(2) Pseudo-label hints help mainly when cluster quality is high. With weak clustering (BERT), adding hints often hurts (e.g., CLINC: $18.58 \rightarrow 42.80$ improves but remains far below strong baselines; MTOP: $32.15 \rightarrow 45.55$ is still low; Massive Intent: $20.07 \rightarrow 40.85$). In contrast, under higher-quality clustering (E5/GTE/BGE), hints become reliably beneficial and the best results typically come from either *With-Hint* or the high-quality Cluster Quality column (e.g., Banking77 reaches 79.48; Massive Scenario reaches 76.40; CLINC reaches 83.13).

(3) Batch-structured prompting can match or exceed strong single-prompt decoding on several datasets, but not universally. On intent-heavy datasets with strong clusters (CLINC, Massive Scenario, Banking77), the best batch-structured configuration approaches or surpasses competitive single-prompt baselines; for example, Banking77 with BGE reaches 79.48, exceeding Self-Consistency (66.82), and CLINC reaches 83.13, close to Self-Consistency (84.22). However, on datasets where the single-prompt baseline is particularly strong or the task is harder to cluster (e.g., StackExchange and Massive Intent), Self-Consistency remains higher (StackExchange: 48.06 vs. best

		Batch-Structured Prompting			Single-Prompt Baselines (GPT-3.5 Turbo)			
Dataset	**Backbone**	**No-Hint** (Unlabeled BS-ICL)	**With-Hint** (Pseudo-Label BS-ICL)	**Homogeneity Given Backbone**	**CoT**	**FoT**	**Self-Consistency**	**Self-Refine**
CLINC	BERT	67.78	56.22	55.40	45.46	46.66	76.56	64.39
	E5	68.89	80.04	80.71				
	GTE	67.71	80.49	81.22				
	BGE	68.80	82.00	**83.13**				
Massive Scenario	BERT	58.81	60.52	58.44	52.01	56.06	63.44	47.70
	E5	59.65	67.25	69.91				
	GTE	60.86	71.32	73.60				
	BGE	60.66	73.97	**76.40**				
MTOP Intent	BERT	56.79	53.67	51.39	58.41	59.64	68.00	39.99
	E5	57.84	52.49	51.48				
	GTE	58.96	53.90	50.98				
	BGE	60.37	57.73	**56.57**				
StackExchange	BERT	31.47	27.45	27.41	9.71	13.50	37.18	21.21
	E5	36.33	38.57	**38.67**				
	GTE	35.68	38.76	**38.98**				
	BGE	34.53	37.05	**37.13**				
Banking77	BERT	51.46	42.08	40.13	27.24	32.34	56.10	36.74
	E5	62.31	71.56	71.56				
	GTE	64.81	75.49	**75.62**				
	BGE	66.72	79.41	**79.48**				
Reddit	BERT	42.56	34.19	32.05	22.69	27.52	41.15	26.88
	E5	51.48	**51.91**	51.63				
	GTE	52.75	**53.93**	53.34				
	BGE	52.10	51.79	51.51				
FewRel-Nat	BERT	31.14	43.46	**42.50**	18.36	17.34	27.52	15.68
	E5	31.00	40.49	41.23				
	GTE	31.65	38.77	38.37				
	BGE	30.76	38.68	37.68				
Massive Intent	BERT	**67.22**	48.15	42.80	52.52	55.89	74.88	55.12
	E5	**71.08**	63.25	61.26				
	GTE	**68.59**	64.93	63.21				
	BGE	**69.94**	69.33	68.83				

Table 8: **Backbone ablation with single-prompt baselines (GPT-3.5 Turbo). No-Hint/With-Hint** are batch-structured prompting variants (BS-ICL); **Cluster Quality** is the specialized coherence proxy induced by the backbone. **CoT/FoT/Self-Consistency/Self-Refine** are *single-prompt* methods and therefore reported once per dataset (shared across backbones). Overall, stronger backbones (E5/GTE/BGE) yield higher cluster quality and larger gains from BS-ICL, and high-quality clustered batches with hints are competitive with—or exceed—strong single-prompt baselines on multiple datasets.

Cluster Quality 38.98; Massive Intent: 74.43 vs. best Cluster Quality 68.83), indicating that batch gains are coherence-dependent rather than unconditional.

Overall, the table supports our central claim: *batch semantic homogeneity is the key moderator*. Stronger encoders produce cleaner batches, which makes pseudo-label hints more reliable and allows batch-structured prompting to become competitive with (and sometimes outperform) advanced single-prompt decoding.

Random batching is not sufficient. To isolate whether accuracy gains come from batching *per se* rather than from semantic structure, we also evaluate *random batch prompting* (no hints) on GPT-4o Mini. Random batching is generally *not* a reliable improvement over single-instance zero-shot prompting: it underperforms on $5/8$ datasets, often by a non-trivial margin (e.g., Reddit: 32.7% vs. 50.2%, MTOP: 65.9% vs. 71.8%, Banking77: 60.3% vs. 64.7%), while the few improvements are small and inconsistent (e.g., Massive Scenario: 64.8% vs. 61.9%; CLINC: 78.8% vs. 78.4%). These results confirm that batching alone is not the source of our improvements. Instead, performance gains require *genuine semantic clusters*: only when the in-prompt set is homogeneous and label-aligned does batch context provide a useful in-context signal.

Dataset	Backbone	Batch-Structured Prompting			Single-Prompt Baselines (GPT-4o Mini)			
		No-Hint	With-Hint	Homogeneity Given Backbone	CoT	FoT	Self-Consistency	Self-Refine
CLINC	BERT	76.82	56.89	55.40	$74.93_{\pm 0.40}\%$	$77.17_{\pm 0.19}\%$	$\mathbf{84.22}_{\pm 0.88}\%$	$68.06_{\pm 0.65}\%$
	E5	77.33	71.87	80.71				
	GTE	77.16	70.93	81.22				
	BGE	78.38	72.64	**83.13**				
Massive Scenario	BERT	64.29	55.78	58.44	$62.96_{\pm 0.28}\%$	$70.17_{\pm 0.24}\%$	$68.99_{\pm 0.76}\%$	$50.27_{\pm 0.95}\%$
	E5	63.21	63.52	69.91				
	GTE	64.36	67.72	73.60				
	BGE	61.87	69.87	**76.40**				
MTOP Intent	BERT	71.82	54.77	51.39	$74.48_{\pm 0.29}\%$	$\mathbf{78.65}_{\pm 0.26}\%$	$73.32_{\pm 1.10}\%$	$39.98_{\pm 0.22}\%$
	E5	70.75	55.77	51.48				
	GTE	71.32	54.65	50.98				
	BGE	71.77	58.50	**56.57**				
StackExchange	BERT	35.90	39.03	27.41	$39.42_{\pm 0.20}\%$	$40.99_{\pm 0.17}\%$	$\mathbf{48.06}_{\pm 0.12}\%$	$25.98_{\pm 0.10}\%$
	E5	41.34	41.03	38.67				
	GTE	41.46	42.11	**38.98**				
	BGE	40.59	41.24	37.13				
Banking77	BERT	58.44	47.76	40.13	$54.41_{\pm 0.46}\%$	$56.33_{\pm 0.51}\%$	$66.82_{\pm 0.28}\%$	$40.23_{\pm 1.06}\%$
	E5	62.60	71.66	71.56				
	GTE	62.89	75.29	75.62				
	BGE	64.68	79.42	**79.48**				
Reddit	BERT	45.63	38.42	32.05	$38.34_{\pm 0.44}\%$	$41.01_{\pm 0.66}\%$	$44.60_{\pm 1.23}\%$	$24.66_{\pm 0.09}\%$
	E5	48.80	52.44	51.63				
	GTE	49.95	53.93	**53.34**				
	BGE	50.20	51.76	51.51				
FewRel-Nat	BERT	35.40	41.29	**42.50**	$28.47_{\pm 0.57}\%$	$33.95_{\pm 0.85}\%$	$\mathbf{43.57}_{\pm 0.28}\%$	$23.49_{\pm 0.13}\%$
	E5	34.84	40.49	41.23				
	GTE	36.29	38.13	38.37				
	BGE	35.58	36.99	37.68				
Massive Intent	BERT	72.76	48.45	42.80	$60.91_{\pm 0.14}\%$	$65.44_{\pm 0.57}\%$	$74.43_{\pm 0.10}\%$	$53.32_{\pm 0.12}\%$
	E5	74.01	62.51	61.26				
	GTE	74.54	64.22	63.21				
	BGE	**75.45**	69.10	**68.83**				

Table 9: **Backbone ablation under batch-structured semi-supervised prompting (GPT-4o Mini). No-Hint / With-Hint** denote clustered batches without / with pseudo-label hints. **Cluster Quality** reflects the resulting batch homogeneity score under each backbone. CoT, FoT, Self-Consistency, and Self-Refine are single-prompt baselines and are shared across backbones.

Dataset	Struct. No-Hint	Struct. With-Hint	Random Batch	CoT	FoT	Self-Cons.	Self-Refine	Homo. (%)	Best
StackExchange	**43.58±0.51**	37.72±0.46	42.16±0.17	39.42±0.20	40.99±0.17	48.06±0.12	25.98±0.10	32.17±0.58	Struct. (No-Hint)
Reddit	43.35±0.09	**51.76±0.13**	37.27±0.00	38.34±0.44	41.01±0.66	44.60±1.23	24.66±0.09	51.51±0.18	Struct. (With-Hint)
Massive-Scenario	68.48±0.31	**75.40±1.02**	66.07±0.00	62.96±0.28	70.17±0.24	68.99±0.76	50.27±0.95	75.32±1.43	Struct. (With-Hint)
MTOP Intent	**73.61±0.66**	55.72±1.64	69.59±0.35	74.48±0.29	78.65±0.26	73.32±1.10	39.98±0.22	53.52±1.69	Struct. (No-Hint)
Massive-Intent	**74.53±0.60**	60.88±1.55	70.11±0.00	60.91±0.14	65.44±0.57	74.43±0.10	53.32±0.12	60.57±1.36	Struct. (No-Hint)
FewRel-Nat	36.54±0.82	33.73±0.28	**37.79±0.00**	28.47±0.57	33.95±0.85	43.57±0.28	23.49±0.13	34.98±0.32	Random Batch
CLINC	**83.31±0.09**	80.76±0.35	80.27±1.25	74.93±0.40	77.17±0.19	84.22±0.88	68.06±0.65	79.54±0.05	Struct. (No-Hint)
Banking77	64.64±0.41	**74.61±0.46**	60.62±0.46	54.41±0.46	56.33±0.51	66.82±0.28	40.23±1.06	74.14±0.62	Struct. (With-Hint)

Table 10: **Unified comparison of batch-structured prompting and single-prompt reasoning baselines (GPT-4o Mini).** Structured prompting processes multiple inputs per query, with or without pseudo-label hints. CoT, FoT, Self-Consistency, and Self-Refine are standard single-prompt baselines shared across backbones. Homogeneity denotes MiniLM student accuracy (%).

Table 11: **Batch-structured prompting vs. single-prompt reasoning baselines (GPT-4o Mini). Bold** indicates the best result within batching methods. <u>Underline</u> indicates the best result overall.

Dataset	Batching Methods			Single-Prompt Baselines			
	Struct. No-Hint	Struct. With-Hint	Random Batch	CoT	FoT	Self-Cons.	Self-Refine
StackExchange	**43.58**	37.72	42.16	39.42	40.99	<u>48.06</u>	25.98
Reddit	43.35	**51.76**	37.27	38.34	41.01	44.60	24.66
Massive-Scenario	68.48	**75.40**	66.07	62.96	70.17	68.99	50.27
MTOP Intent	**73.61**	55.72	69.59	74.48	<u>78.65</u>	73.32	39.98
Massive-Intent	**74.53**	60.88	70.11	60.91	65.44	74.43	53.32
FewRel-Nat	36.54	33.73	**37.79**	28.47	33.95	<u>43.57</u>	23.49
CLINC	**83.31**	80.76	80.27	74.93	77.17	<u>84.22</u>	68.06
Banking77	64.64	**74.61**	60.62	54.41	56.33	66.82	40.23

Dataset	Δ Struct. (No-Hint)	Δ Struct. (With-Hint)	Δ Random Batch	Best Single-Prompt
StackExchange	−4.48	−10.34	−5.90	Self-Consistency (48.06)
Reddit	−1.25	**+7.16**	−7.33	Self-Consistency (44.60)
Massive-Scenario	−1.69	**+5.23**	−4.10	FoT (70.17)
MTOP Intent	−5.04	−22.93	−9.06	FoT (78.65)
Massive-Intent	**+0.10**	−13.55	−4.32	Self-Consistency (74.43)
FewRel-Nat	−7.03	−9.84	−5.78	Self-Consistency (43.57)
CLINC	−0.91	−3.46	−3.95	Self-Consistency (84.22)
Banking77	−2.18	**+7.79**	−6.20	Self-Consistency (66.82)

Table 12: **Accuracy difference (Δ) between batch-structured prompting and the best single-prompt baseline.**(GPT-4o Mini) Positive values indicate that batching exceeds the strongest single-prompt method (CoT, FoT, Self-Consistency, or Self-Refine) on that dataset.

Dataset	Δ Struct. (No-Hint)	Δ Struct. (With-Hint)	Δ Random Batch	Best In-Context
StackExchange	**+2.59**	−3.27	+1.17	FoT (40.99)
Reddit	+2.34	**+10.75**	−3.74	FoT (41.01)
Massive-Scenario	−1.69	**+5.23**	−4.10	FoT (70.17)
MTOP Intent	−5.04	−22.93	−9.06	FoT (78.65)
Massive-Intent	**+9.09**	−4.56	+4.67	FoT (65.44)
FewRel-Nat	+2.59	−0.22	**+3.84**	FoT (33.95)
CLINC	**+6.14**	+3.59	+3.10	FoT (77.17)
Banking77	+8.31	**+18.28**	+4.29	FoT (56.33)

Table 13: **Accuracy difference (Δ) between batch-structured prompting and the strongest single-prompt in-context baseline (CoT or FoT).**(GPT-4o Mini) All methods use a single forward pass; Self-Consistency and Self-Refine are excluded.

Dataset	Struct.(No-Hint)	Random batch	Δ (Random − Zero)
Reddit	50.20	32.70	−17.50
StackExchange	40.69	39.77	−0.92
MTOP Intent	71.77	65.94	−5.83
Massive Scenario	61.87	**64.76**	+2.89
Massive Intent	75.45	70.11	−5.34
FewRel-Nat	35.58	**36.96**	+1.38
CLINC	78.38	**78.84**	+0.46
Banking77	64.68	60.29	−4.39
Win count	–	**3/8**	–

Table 14: **Random batch vs. single-instance zero-shot (GPT-4o Mini, No-Hint).** Random batching is not a reliable improvement over zero-shot single prompting: it underperforms on 5/8 datasets (often substantially, e.g., Reddit) and only yields small gains on 3/8 datasets. This supports the claim that *batching alone is insufficient*; consistent gains require *semantic homogeneity* (high-quality clustering) rather than random grouping.

Dataset	Random	No Hints (MiniLM)	No Hints (BERT)	With Hints (MiniLM)	With Hints (BERT)	MiniLM (High Homogeneity)	BERT (Low Homogeneity)	Winner
Banking77	73.96±0.14	**76.97±0.39**	74.73±0.90	76.84±0.80	63.04±0.25	74.58±0.61	39.45±0.95	No Hints (MiniLM)
CLINC	89.13±0.03	**90.64±0.40**	89.45±0.39	83.91±1.64	77.81±0.15	79.31±0.23	55.22±0.32	No Hints (MiniLM)
FewRel-Nat	**60.66±0.74**	58.98±0.55	59.90±0.62	43.59±0.93	48.66±0.89	34.75±0.29	42.50±0.41	Random
Massive-Intent	83.10±0.40	**83.47±0.17**	82.88±0.28	71.87±0.12	67.37±0.69	59.62±0.21	42.80±0.33	No Hints (MiniLM)
Massive-Scenario	72.69±0.64	74.23±0.22	72.36±0.29	**78.27±0.69**	71.45±1.67	75.33±0.26	58.47±0.81	With Hints (MiniLM)
Reddit	58.28±1.76	**60.92±2.40**	59.79±0.64	57.48±1.63	51.49±1.21	51.51±0.18	31.68±0.54	No Hints (MiniLM)
StackExchange	**67.56±0.03**	67.36±0.89	66.70±0.54	50.05±1.46	48.35±1.72	31.76±0.31	27.10±0.46	Random

Table 15: **Structure-Driven Batching on a Strong LLM (DeepSeek-chat).** Accuracy (%), reported as mean ± std. *Random* denotes random batching. *Structured* (*No Hints*) denotes clustered batching without pseudo-label hints. *Structured* (*With Hints*) denotes clustered batching with pseudo-label hints. Student accuracies are homogeneity proxies measured independently.

Dataset (LLaMA-3-8B-Instruct)	Batch Prompting Strategy (increasing semantic homogeneity)			
	Random	No-Hints (Low-Homogeneity)	No-Hints (High-Homogeneity)	With-Hints (High-Homogeneity)
Clinc	$25.57_{\pm 1.43}$	$33.96_{\pm 0.67}$	$32.49_{\pm 6.73}$	$\mathbf{69.40_{\pm 7.28}}$
Massive-Scenario	$41.56_{\pm 0.13}$	$44.42_{\pm 0.03}$	$43.52_{\pm 1.85}$	$\mathbf{66.74_{\pm 0.98}}$
Mtop-Intent	$27.59_{\pm 0.48}$	$34.67_{\pm 1.27}$	$34.17_{\pm 6.70}$	$48.23_{\pm 0.25}$
StackExchange	$10.46_{\pm 0.44}$	$13.84_{\pm 0.55}$	$11.02_{\pm 2.78}$	$\mathbf{26.26_{\pm 2.16}}$
Banking77	$12.34_{\pm 0.75}$	$21.27_{\pm 0.55}$	$33.06_{\pm 1.92}$	$\mathbf{69.66_{\pm 1.74}}$
Reddit	$16.65_{\pm 0.29}$	$26.29_{\pm 1.45}$	$36.31_{\pm 0.97}$	$\mathbf{46.00_{\pm 2.51}}$
FewRel-Nat	$8.56_{\pm 0.35}$	$13.17_{\pm 0.11}$	$14.25_{\pm 0.36}$	$\mathbf{31.80_{\pm 0.34}}$
Massive-Intent	$17.57_{\pm 0.35}$	$31.05_{\pm 0.49}$	$45.41_{\pm 0.06}$	$56.03_{\pm 0.08}$

Table 16: **Structure-driven batching on LLaMA-3-8B-Instruct.** Accuracy (%), mean ± std. Batching strategies are arranged in increasing semantic homogeneity. Performance improves markedly when high-homogeneity structure is combined with pseudo-label hints, supporting the hypothesis that semantic coherence acts as an inference-time supervisory signal in demonstration-free in-context learning.

C. Analysis across competency regimes.

The additional experiments reinforce our coherence–competency hypothesis.

Strong LLM (DeepSeek-chat). For a strong model (Table 15), structured batching without hints often matches or outperforms hint-based variants (e.g., **Banking77**, **CLINC**, **Reddit**, **Massive-Intent**). This suggests that high-competency models can already extract sufficient signal from semantically coherent batches, rendering pseudo-label hints redundant or even harmful when noisy. Hints help only when clustering aligns strongly with latent structure (e.g., **Massive-Scenario**), indicating that pseudo-label utility is conditional rather than uniformly beneficial.

Small Open-Source LLM (LLaMA-3-8B-Instruct). For the weaker model as show in Table 16, performance generally increases with semantic homogeneity. High-fidelity clustering substantially improves accuracy, and hint-augmented clustering yields large gains across most datasets. This pattern is consistent with a diffuse uncertainty profile: weaker models depend strongly on structured context and benefit from reliable pseudo-label anchoring when structural coherence is high.

Weak Closed-Source LLM (GPT-4o Mini). Table 17 exhibits clear non-monotonic behavior. Pseudo-label hints improve accuracy when homogeneity is high (e.g., **Reddit**, **Banking77**, **Massive-Scenario**) but degrade performance when coherence is moderate or low (e.g., **MTOP**, **Massive-Intent**). This aligns with our regime view: when batch coherence is insufficient, noisy hints amplify error; when coherence is strong, hints act as effective supervision.

Overall pattern. Across models, we observe a consistent interaction: (i) higher batch homogeneity increases the effectiveness of structure-driven prompting, and (ii) the competency of the LLM shifts the coherence threshold at which pseudo-label hints become beneficial. Stronger models require less explicit supervision, while weaker models depend more heavily on high-quality structural or pseudo-label guidance. These results empirically support our interpretation of batch-structured prompting as an implicit aggregation mechanism whose robustness is jointly governed by utility concentration (coherence) and worker reliability (competency).

Table 17: **Structure-Driven Batching on a Weak LLM (GPT-4o Mini).** Accuracy (%), mean ± std. *Structured* (*No Hints*) corresponds to zero-shot batching. *Structured* (*With Hints*) corresponds to retrieval-augmented batching. Student MiniLM accuracy is used as a homogeneity proxy.

Dataset	No Hints	With Hints	Random	Δ(With–No)	Homogeneity (MiniLM)	Winner
StackExchange	**43.58±0.51**	37.72±0.46	42.16±0.17	−5.86	32.17±0.58	No Hints
Reddit	43.35±0.09	**51.76±0.13**	37.27±0.00	+8.40	51.51±0.18	With Hints
Massive-Scenario	68.48±0.31	**75.40±1.02**	66.07±0.00	+6.93	75.32±1.43	With Hints
MTOP Intent	**73.61±0.66**	55.72±1.64	69.59±0.35	−17.89	53.52±1.69	No Hints
Massive-Intent	**74.53±0.60**	60.88±1.55	70.11±0.00	−13.66	60.57±1.36	No Hints
FewRel-Nat	36.54±0.82	33.73±0.28	**37.79±0.00**	−2.81	34.98±0.32	Random
CLINC	**83.31±0.09**	80.76±0.35	80.27±1.25	−2.55	79.54±0.05	No Hints
Banking77	64.64±0.41	**74.61±0.46**	60.62±0.46	+9.97	74.14±0.62	With Hints

D. 2×2 Homogeneity–Competency Regimes

Specifically, our experiments reveal a regime-dependent shift in how label supervision operates under implicit demonstration in-context prompting. In the absence of explicit demonstrations, pseudo-label hints do not function as conventional supervision. Instead, they act as an inference-time inductive bias whose utility depends jointly on (i) batch semantic homogeneity and (ii) LLM competency. Across datasets and models, we observe a consistent competency–homogeneity interaction governing when pseudo-label hints help or harm performance.

High-homogeneity batches (MiniLM-clustered). Strong LLMs (e.g., DeepSeek-chat) already exploit structural coherence effectively. In this regime, pseudo-label hints provide little additional signal and may amplify residual clustering noise, leading to degraded performance (e.g., Banking77, CLINC, Massive-Intent, Reddit). For weaker LLMs (e.g., GPT-4o Mini), however, pseudo-label hints become beneficial, acting as lightweight supervision that compensates for limited reasoning capacity (e.g., Reddit, Banking77, Massive-Scenario).

Low-homogeneity batches (BERT-clustered or random). When structure is unreliable, pseudo-label hints provide noisy or inconsistent signals. For strong LLMs, random batching or structure-only prompting remains competitive or superior (e.g., FewRel-Nat, StackExchange). For weaker LLMs, neither structure nor hints consistently improve performance, resulting in a capability-limited regime (e.g., FewRel-Nat, CLINC, MTOP).

These findings demonstrate that pseudo-label hints are not universally beneficial in demonstration-free in-context learning. Their effects depend jointly on batch homogeneity and model competency, which we formalize in the proposed 2×2 homogeneity–competency framework. Importantly, semantic batch homogeneity emerges as an orthogonal in-context signal, whose interaction with supervision differs fundamentally from traditional demonstration-based ICL. Performance degradation in certain regimes is therefore not anomalous but a predictable consequence of the competency–homogeneity interaction.

E. Homogeneity-Performance Relationship Across LLM Scales

To systematically study the role of batch homogeneity, we vary the **clustering encoder** from weaker to stronger semantic backbones (BERT → E5 → GTE → BGE), thereby inducing progressively higher levels of measured homogeneity. Importantly, the inference model remains fixed within each experiment (Llama-3-8B-Instruct, GPT-3.5 Turbo, or GPT-4o Mini), isolating the structural effect of batch composition.

Across all three inference models, we observe a strong positive association between measured homogeneity and No-Hint accuracy. The pooled Pearson correlations are:

- **Llama-3-8B-Instruct:** $r = 0.716$, $p < 10^{-5}$
- **GPT-3.5 Turbo:** $r = 0.803$, $p < 10^{-7}$
- **GPT-4o Mini:** $r = 0.718$, $p < 10^{-5}$

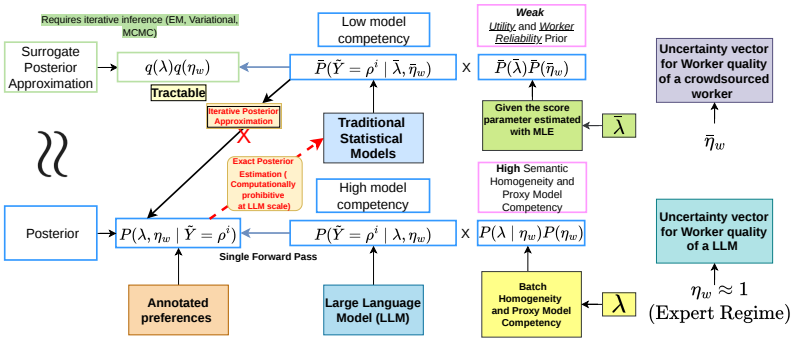

Figure 3: Interpretive Bayesian lens for batch-structured prompting. Conventional ranking aggregation methods aim to approximate the posterior over latent utilities and worker reliabilities through iterative inference procedures, which can be computationally intensive.In contrast, our approach does not explicitly estimate priors or optimize a posterior. Instead, semantic homogeneity in batch-structured prompting implicitly sharpens the effective utility landscape, while model competency governs uncertainty in stagewise selection.

These results indicate that higher-quality clustering—corresponding to more semantically coherent batches—consistently improves inference performance under batch-structured prompting. The effect is observed across models of varying strength, suggesting that homogeneity acts as a structural factor shaping posterior sharpness rather than a model-specific artifact.

Overall, the findings provide quantitative support for the **homogeneous batch hypothesis**: structured semantic coherence enhances reasoning efficiency and predictive reliability in semi-supervised batch prompting.

F. From Iterative Posterior Aggregation to Inference-Time Utility Concentration

Classical ranking aggregation. In traditional Plackett–Luce (PL) ranking aggregation, the goal is to infer latent utilities $\boldsymbol{\lambda}$ and worker reliabilities $\boldsymbol{\eta}_w$ from noisy observations $\tilde{Y}$. Bayesian inference yields

$$P(\boldsymbol{\lambda}, \boldsymbol{\eta}_w \mid \tilde{Y}) \;\propto\; P(\tilde{Y} \mid \boldsymbol{\lambda}, \boldsymbol{\eta}_w)\, P(\boldsymbol{\lambda})\, P(\boldsymbol{\eta}_w). \tag{7}$$

Computing or approximating this posterior typically requires iterative inference procedures, such as expectation–maximization, variational optimization, or Markov chain Monte Carlo sampling. These methods explicitly estimate both utility parameters and worker uncertainty profiles, and posterior concentration emerges through repeated updates over the data.

Scalability limitations at LLM scale. While principled, exact or iterative posterior inference becomes computationally infeasible when the likelihood term is instantiated by a large language model (LLM). The cost of repeated forward passes, parameter updates, or uncertainty estimation at LLM scale makes classical Bayesian aggregation impractical for inference-time deployment.

Inference-time surrogate in our framework. Our approach does not explicitly estimate $P(\boldsymbol{\lambda})$ or $P(\boldsymbol{\eta}_w)$. Instead, we introduce operational surrogates that approximate the effect of posterior concentration without performing Bayesian optimization.