

Rethinking Clustering as a Conversational Supervision Problem

Theo Rode*, Aimee Co*, Qingyang Zeng*,
Sanaya Gandhi*, Ananya Purwar*, Jonathan P. Chang*
Masoud Hashemi†, Khyati Mahajan†, Filip Deryckere†, Fabio Casati†‡

*Harvey Mudd College

†ServiceNow

‡University of Trento

Abstract

Clustering is often a process of trial and error: we apply different algorithms (such as K-means or HDBSCAN), apply different embeddings, try a dimensionality reduction, explore different values of K, inspect the result with the product manager, and repeat. When the data is large, even communicating and analyzing a single clustering becomes difficult.

We argue that the nature of clustering requires a fundamentally different approach. We posit that clustering is fundamentally a subjective and iterative process, and its central challenges are how to *elicit* the dimensions and nuances a user cares about, how to *communicate* a clustering effectively, and how to *converge* to an output of interest through efficient conversation. We re-frame clustering as a *conversational supervision* problem. We formalize *conversational clustering* as a new task, derive evaluation metrics suited to it, and release an evaluation framework and reference implementation to seed further work.

1 Introduction

In our line of work we routinely have to cluster large document corpora for enterprise customers (such as KB articles, customer support requests, case resolutions). Clustering data in general—and text clustering in particular—involve a nontrivial sequence of choices, typically performed by a data scientist or ML engineer: We choose an embedding method (pretrained or learned), a dimensionality reduction, a distance function, an algorithm such as K-means or HDBSCAN (McInnes et al., 2017). Different choices can lead to wildly different results.

Clustering may not be unique in the sensitivity of output to methods (though we do find it extreme), but there are aspects about clustering that make us think whether we should reframe how we formulate the problem.

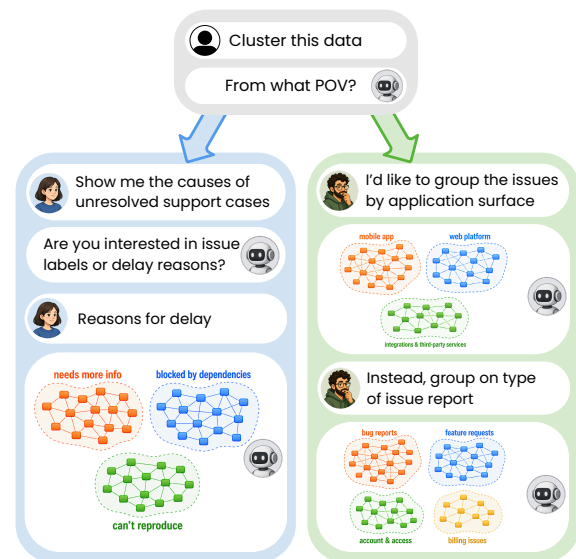


Figure 1: Example conversation flows showing different POVs on the same dataset.

1.1 On the Nature of Clustering Problems

1. Clustering is Subjective. Different stakeholders often look for different ways of grouping a dataset. A product manager may be interested in issue type, another on resolution efficiency by language, and yet another may focus on sentiment (Figure 1). Both datasets and analysis objectives are nuanced, and the fact successful text embeddings are fairly high dimensional underscores the need to capture such nuances. The space of valid groupings, the *partition lattice* (Meilă, 2007), is itself factorially large, and standard clustering returns one vertex of it, chosen by the embedding’s geometry rather than by the user. Kleinberg’s impossibility theorem (Kleinberg, 2002) shows that this subjectivity is not just a practical inconvenience but a mathematical one: no clustering function can simultaneously satisfy scale-invariance (invariance to uniform rescaling of distances), richness (every partition is achievable for some distance), and consistency (shrinking within-cluster

distances and expanding between-cluster distances preserves the output). Which axiom one is willing to give up reflects what the user actually cares about (Hennig, 2015; von Luxburg et al., 2012).

2. Clustering is *Inherently Conversational*. We have rarely encountered a situation where a customer could express their point of view of interest in a clustering. A comprehensive and well-formed understanding of what a user wants from a clustering generally arises from iterations and examples. For example, it is only when we realize that the system separates customer support requests by sentiment that we may want to state that sentiment is *not* a dimension of interest for us.

3. Clustering Lacks an Effective Language. The typical enterprise text dataset has hundreds of thousands of data points (text documents). A first-pass clustering may easily give thousands of clusters with massive variations in cluster size. The combination of scale of the datasets with scale of possible analysis dimensions makes it hard to convey clustering results to users and get their feedback. We cannot show users thousands of text documents and clusters they belong to and ask for feedback. We see users getting “clustering fatigue” after a few examples. Showing clusters—or, system understanding of users’ clustering intent—requires combinations of practices, explanations of system intentions, identification of “good”, “representative” samples, and more. Today we do not really have a language for that, or even established practices.

1.2 Conversational Supervision

These observations have direct consequences: there is no objective way to declare one clustering better than another without referencing the user. Reaching a satisfactory clustering is fundamentally a process, not a one-shot operation, and we need a language for communicating clusterings to the user and eliciting feedback. This in turn means that successful clustering has a “UX” aspect (ability to facilitate information consumption by humans) and an exploration aspect (ability to expose different aspects one may want to cluster on, e.g., documents language, or sentiment, or product, or whether the document contains a problem resolution or just a problem statement). We argue that clustering is an instance of *conversational supervision* problems, where conversation is the search for both the right representation of the data and the right notion of “correct” for the user. Each user turn—“these two

tickets belong together because of X ”, “don’t separate by sentiment”, “cluster A is too coarse”—is a cheap, noisy, partial constraint; in aggregate, these constraints act as weak supervision (Ratner et al., 2017), narrowing the space of plausible clusterings toward the user’s intended grouping.

Conversational supervision also points to a constructive escape from Kleinberg’s impossibility. Ackerman and Ben-David (2008) show that the contradiction dissolves once one scores candidate clusterings with a quality measure rather than prescribing a single “correct” algorithm. Conversational supervision is a realization of this idea: the quality measure is not fixed in advance but built turn by turn from the user’s evolving judgment.

LLMs are what make this tractable for the first time. Before them, the only realistic feedback channels were must- and cannot-link constraints (Basu et al., 2004), too narrow to express high-level intents such as “ignore sentiment” or “cluster by cross-departmental friction”.

Clustering is already sort of “conversational” - or at least iterative - today, where long and complex interactions take place among data scientists and domain experts / process owners, and the tools available for iterations are changes to the embedding algorithm, or to the clustering algorithm, or to dimensionality reduction, or to some hyperparameter of the above. Rather than recognizing and redefining the problem, we try to approach it with the tools at our disposal but these tools were not built to support an effective conversation with end users of a clustering task.

While we focus on clustering, we see problems with the exact same characteristics arising from other areas or work. For example, extracting graphs (entities and relationships) that make sense to humans from unstructured enterprise data also requires describing which are the aspects we want to focus on or when two “things” are similar or different (e.g., do we consider Zoom and Teams recording as the same issue). Most importantly, it often implies stating aspects we do *not* want to focus on (e.g., language or sentiment).

Conversational supervision represents a fundamentally new - and important - class of ML problems and is in a way a new discipline that blends “classical” ML, LLMs, RL-style exploration, and user interaction as a first-class citizen. As we seem heading towards a world where information is abundant and our access to it is mediated by AI, making sense of data at scale becomes paramount. Any AI-

assisted analysis of large volumes of data (as we have seen in the OpenAI / HuggingFace incident¹) introduces a subtle form of dependence: when an LLM filters large information streams into signals, summaries, and alerts, it partly determines what human overseers are able to notice. Important evidence may therefore be omitted or absorbed into apparent noise, even without deliberate deception by the model.

1.3 Objectives of This Work

Our goal is not to engineer the best implementation of conversational clustering, but to introduce the concept, give it a name, a formalism and an evaluation framework (questions we need to seek answers to). We also present early evidence that the paradigm is realizable now that LLMs are available (and was not before). Concretely we contribute: **(1) a problem formulation** that reframes clustering as conversational supervision, where each turn prunes the partition lattice toward user intent and side-steps both the implicit biases of unsupervised tools and the labeling cost of metric learning; **(2) a formalism** that isolates the key elements (oracle, intent, system state, natural-language interaction) and exposes an explicit conversational loop; **(3) an evaluation framework** of six prongs derived from the formalism (discovery, controllability, faithfulness, robustness, final clustering quality, efficiency) with concrete metrics; and **(4) feasibility evidence with LLMs**: a synthetic enterprise benchmark with hidden POV labels and a deliberately minimal reference implementation showing the paradigm works, plus insights on wiring LLMs into the loop. We make no claim of optimality; the implementation exists to test the framework and give the community a starting point.

2 Related Work

To our knowledge, no prior work studies *conversational clustering* in the formulation described here. We organize prior work around two topics: conversational paradigms in adjacent problems, and feedback integration into clustering.

2.1 Existing Conversational Systems

Other existing problem spaces have undergone similar transformations, such as conversational search, which gives us insight and grounding into how we consider conversational clustering.

¹<https://huggingface.co/blog/security-incident-july-2026#analyzing-an-ai-driven-intrusion>

Conversational Search. Similar to our proposed revision to the clustering problem, search has experienced a push towards developing conversational interfaces (Chen et al., 2024; Mao et al., 2024; Mo et al., 2026; Jang et al., 2024; Mo et al., 2023; Zhu et al., 2025). These systems aim to use the more nuanced intent specification present in conversation to guide the search process closer to the user’s desired results. A key problem in the conversational search field is understanding how to weigh the inherent trade-offs between presenting a final search result to the user or asking clarifying questions to gain a better understanding of their hidden intent (Wang et al., 2023). The recognition here is that the user (or “oracle”) has a finite ability to convey their intent and to understand the possible search space, and therefore it is the system’s job to efficiently extract enough of an understanding of user intent to retrieve their desired results.

Both search and clustering have a somewhat subjective and conversational aspect. However, search does not aim at identifying a partition. It is also comparatively easier to communicate results and for the user to understand if the result is “good enough”. Search engines have also developed a science over the years that leverages aspects such as search history and implicit and explicit signals.

Conversational Problem-Solving. The study of conversational systems more broadly has considered their impact on conveying and refining nuanced user intent. Many of these works have considered building solution artifacts throughout such a conversation (Ye et al., 2025; Dutta et al., 2025), a process that mirrors the methodology that we present in this work. While these works focus on developing conversational strategies for building and refining on abstract ideas, they do not target clustering specifically.

2.2 Feedback Integration into Clustering

Substantial existing work studies the problem of integrating user feedback into clustering.

Semi-Supervised Clustering & Topic Modeling. Semi-supervised clustering broadly refers to the integration of user-provided labels to the clustering process. Often, these labels are must- and cannot-link relations, where such information can be integrated into the loss function (Basu et al., 2004). While useful for incorporating user intent into a clustering, these methods do not break away from inherent space or distance assumptions. Further,

they do not immediately offer an iterative interaction process, limiting the extent to which users can provide feedback.

Similarly, in enterprise text applications, unsupervised grouping is frequently approached using topic modeling, such as Latent Dirichlet Allocation (LDA) (Blei et al., 2003). While interactive extensions of these models allow users to refine topics by adding or removing anchor words (Hu et al., 2014), they suffer from the same inherent subjectivity and nuance constraints as traditional clustering—the number of topics, dimensions of focus, and coherence thresholds are all user-dependent.

Metric and preference learning for clustering.

A natural way to incorporate user preferences into clustering is to make the representation or metric itself learnable, so that distances reflect user preferences instead of generic geometry. Deep metric learning (Kaya and Bilge, 2019) pursues this with contrastive and triplet objectives; collaborative metric learning brings the idea to recommender systems and personalization (Hsieh et al., 2017); and joint metric and preference learning over multiple users (Canal et al., 2022) learns a shared distance plus per-user latent preference points from pairwise comparisons. Adjacent work on calibrating metrics to human preferences (Winata et al., 2025) pursues the same goal at the metric level. All of these methods, however, require preference or label data, do not adapt cheaply to a new user, and produce manifolds that are hard to inspect. We sidestep these trade-offs by replacing offline supervision with in-conversation supervision.

LLM-driven Clustering. Pang-Naylor et al. examined how LLM re-writing and instruction-appending to embedding allowed for clustering based on under-represented themes within data. While this work considered addressing the issue of subjectivity within clustering—representing nuanced, low-signal POVs within data—it still does not immediately offer an interactive process.

Interactive clustering introduces iterative refinement with the user, where a human (or LLM) provides feedback on constrained, high-impact regions of a clustering. Here, determining high-value points for feedback within a clustering is integral to the performance of such methods (Liu et al., 2024). However, these methods often incorporate feedback by swapping cluster labels of individual points (Feng et al., 2024), using existing clustering metrics to understand high-value areas (Deng

et al., 2024), or determining whether to ask for feedback on a cluster based on if it is “semantically coherent” (Hong et al., 2025). While these present useful strategies for refining a clustering to user preference, they fundamentally impose structure to the resultant clustering. Because interactions with these systems are not in the form of a conversation, they are ill-equipped to incorporate abstract user objectives. Without the semantic flexibility provided by open-ended natural language, users are constrained to rigid structural alteration, lacking the abstract feedback mechanisms to express high-level, nuanced objectives.

Together, this work shows that clustering can benefit from semantic supervision and refinement, but none fundamentally break from notions of objectivity in clustering. They lack channels for the nuanced, abstract feedback which we argue is necessary for the *subjective* nature of clustering.

3 Formalizing Conversational Clustering

Traditional clustering assumes embeddings of points of a dataset $\mathcal{D}$ in space and then identifies a partition P that minimizes some loss $\mathcal{L}(P, d)$, where d is some distance metric. The loss function is defined implicitly by picking the algorithm, and can sometimes be further specified by hyperparameters. There is no notion of conversationally discovering what is the “loss” or “value” we seek in a clustering. Rather than relying solely on this formulation, which assumes that the metric d is a proxy for user intent, we define conversational clustering (and more generally conversational supervision) as a process of intent alignment.

3.1 Definitions

Given a dataset $\mathcal{D} = \{x_1, x_2, \dots, x_n\}$, we consider a conversational clustering problem as characterized by a Clustering System and an Oracle.

The **Oracle** $\mathcal{O}$: the entity (typically a human user) that holds the latent intent and is the only source of ground-truth feedback. Operationally, $\mathcal{O}$ acts as an evaluation function $\mathcal{O}(P \mid \mathcal{I})$ that scores a candidate partition P given its current intent $\mathcal{I}$, and as a feedback function that returns a natural-language message in response to P .

- An *Intent* ($\mathcal{I}_t$): The latent, subjective, POV held by the oracle at turn t . Intent is not static—it evolves as the oracle interacts with the data and candidate partitions P_t . It extends beyond

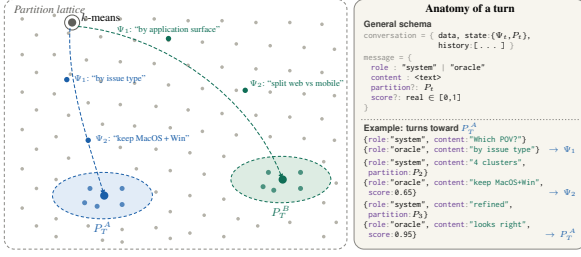


Figure 2: Conversational clustering as partition lattice pruning. *Left:* a standard unsupervised pipeline (k -means) returns one arbitrary vertex of the partition lattice. In conversational clustering, each turn supplies a partial equivalence relation that is folded into Ψ_t , pruning the lattice toward an intent-consistent region; two oracles on the same data can reach different but valid results P_T^A, P_T^B . *Right:* the conversation API. A turn is a message carrying a role (system or oracle), free-form content, an optional partition attached by the system on PRESENT, and an optional score reporting the oracle’s evaluation $O(P_t | \mathcal{I}_t) \in [0, 1]$. The worked example shows three turns converging to P_T^A . See Section 3.1 for more details and notations.

the named POV, encompassing the full understanding of the desired grouping.

- A *Subjective Ideal Partition* ($P_{\mathcal{I}_t}^*$): A theoretical, unobservable partition of $\mathcal{D}$ that represents the maximum possible satisfaction of $\mathcal{I}_t$ at turn t . It is not an intrinsic property of the data, but rather a concept that would perfectly reflect the oracle’s subjective judgment if it possessed the capacity to extensionally define it.

The **Clustering System** ($\mathcal{C}$), besides being characterized by its underlying algorithm, possesses:

- A *System State* ($\mathcal{S}_t = (P_t, \Psi_t)$): at turn t , the most recently presented partition $P_t : x_i \mapsto \{1, \dots, K_t\}$ together with a transformation Ψ_t (an instruction, prefix, or re-embedding policy) that encodes the equivalence constraints accumulated from the conversation so far. Updating Ψ_t is the operational act of pruning the partition lattice.
- A *Conversation* ($\mathcal{M}_t$): the alternating messages exchanged by the system and the oracle $\mathcal{M}_t = \{m_0^O, m_1^S, m_1^O, \dots, m_t^S, m_t^O\}$.

An example of such interactions is the communication of the oracle’s *Point of View* (POV), which is an intensional linguistic label (e.g., “User Intent”, “Product Category”, or “Ur-

gency”) used to communicate a facet of the latent intent.

We make a **bounded-oracle assumption**: the system observes $\mathcal{D}$ in full but not $\mathcal{I}_t$, while the oracle holds $\mathcal{I}_t$ but observes $\mathcal{D}$ only through the system’s selections. Therefore, $\mathcal{O}$ cannot extensionally produce its ideal partition in one shot, both because it cannot process all of $\mathcal{D}$ at once and because intent is partly discovered through interaction. The system’s role is therefore not only to partition but also to elicit and surface candidate POVs. At each turn the system must choose samples, questions, and partitions to maximize information gain about $\mathcal{I}_t$ while minimizing oracle burden.

Objective. A conversational clustering system jointly optimizes *intent convergence* and *interaction cost*:

$$\min_T \text{Cost}(\mathcal{M}_T) \quad \text{s.t.} \quad \mathcal{O}(P_T | \mathcal{I}_T) \geq 1 - \epsilon, \quad (1)$$

where $\text{Cost}(\mathcal{M}_T) = \sum_{t=0}^T \text{cost}(m_t)$ captures cognitive burden per exchange. Since $\mathcal{I}_t$ is unobservable, the system cannot compute $\mathcal{O}(\cdot | \mathcal{I}_t)$ directly and must rely on Ψ_t as a proxy, balancing exploitation of stated constraints with exploration of the POV space.

Conversational clustering as partition lattice pruning. Algorithm 1 states the process. Each turn the system chooses, via a POLICY, between two actions: ASK a clarifying question generated after looking at the data, or CLUSTER a candidate partition and PRESENT it. The oracle’s response m_t^O is read as a (partial) equivalence relation in both cases; UPDATE accumulates these into Ψ_t , pruning the partition lattice by ruling out partitions inconsistent with the constraints so far, while CLUSTER returns a representative partition consistent with the current Ψ_t . The first turn typically uses ASK because $\Psi_0 = \emptyset$, but the algorithm leaves this to the policy: the same loop covers exploratory question-asking and partition refinement.

4 Evaluation Framework

4.1 Evaluating Conversational Clustering

Once we formulate the problem we need to consider how to construct evaluations and enable experimentation. An eval system would require datasets containing rich POVs and concrete implementations of metrics in each evaluation prong described in Section 4. Most importantly, to experiment and

Algorithm 1 Conversational Clustering

Input: Dataset $\mathcal{D}$, oracle $\mathcal{O}$, base EMBED, satisfaction threshold ϵ

Output: Final partition P

```
1:  $\Psi_0 \leftarrow \emptyset$  {no constraints yet}
2:  $P \leftarrow \perp$  {no partition shown yet}
3:  $t \leftarrow 0$ 
4: repeat
5:    $a_t \leftarrow \text{POLICY}(\Psi_t, P, \mathcal{M}_t)$  {ASK or CLUSTER}
6:   if  $a_t = \text{ASK}$  then
7:      $m_t^S \leftarrow \text{QUESTION}(\text{SAMPLE}(\mathcal{D}), \Psi_t)$ 
8:   else
9:      $P \leftarrow \text{CLUSTER}(\Psi_t(\mathcal{D}, \mathcal{M}))$ 
10:     $m_t^S \leftarrow \text{PRESENT}(P, \Psi_t)$ 
11:     $m_t^O \leftarrow \mathcal{O}.\text{FEEDBACK}(m_t^S)$ 
12:     $\Psi_{t+1} \leftarrow \text{UPDATE}(\Psi_t, m_t^O)$ 
13:     $t \leftarrow t + 1$ 
14: until  $\mathcal{O}(P \mid \mathcal{I}_t) \geq 1 - \epsilon$ 
15: return  $P$ 
```

iterate at scale we need to construct *bounded* oracles that can sustain a conversation and be steered towards different sets of POVs, reflecting the subjectivity of the user. We see success as being defined by the quality and efficiency of the alignment process across several key dimensions.

Discovery Quality. We assume that the oracle initially comes with a limited understanding of the many POVs present in any underlying dataset. Some POVs may be dominant (e.g., “Issue Type”) and standard embeddings plus clustering would naturally expose them, others are more subtle and harder to capture (e.g., high number of agent-customer back and forth). A high quality system should surface POVs early in the conversation. Here, we care if the system can find all—and in particular low-signal—POVs, how frequently it describes unrelated POVs, and how early the extent of POVs is communicated.

Objective Elicitation and Controllability. Beyond the exploration phase of an interaction, we care about whether the oracle can control the shape of the clustering through feedback. Hence, we must measure the objective elicitation and controllability of the system. If the oracle requests a change in POV, does the system accurately move towards the new POV? Can the system respond to requests like “increase the number of clusters” or “split cluster A” without modifying irrelevant clusters? Beyond an ability to reflect oracle feedback, it is important to understand how the system is able to incorpo-

rate ambiguous input. Here, the *clarification policy* is important to evaluate: does the system ask for clarification when needed? And, when requested, how effective is this clarification at improving the ultimate final clustering?

Faithfulness. While adhering to the oracle’s specified POV direction is important for a conversational clustering system, this behavior should extend as far as is faithful to the underlying data. A major vulnerability for LLM-based systems is sycophancy (Perez et al., 2023), where models agree with a user’s flawed premise instead of correcting them. In conversational clustering, a system might hallucinate artificial clusterings to appease a user’s request. Thus, we care about the faithfulness of a conversational clustering system: does it accurately represent the data and can it reject clustering on POVs that are not supported by evidence?

Robustness. A conversational clustering system should be agnostic to the underlying oracle. Therefore, a system should be able to demonstrate robustness against changes in phrasing and behavior of the oracle, given the same oracle’s goal.

Final clustering quality. Here, we care about evaluating how well a final clustering adheres to the ultimate POV chosen by the oracle—notably not an initial POV, as we assume the oracle’s POV will be refined through interaction.

Conversation qualities. We humans have limited time and finite cognitive bandwidth, so a system that achieves intent alignment in fewer - and cognitively easier - turns is inherently more desirable than one that requires prolonged engagement.

4.2 Datasets

Creating “good” datasets that enable an informative eval flow with “human-like” oracles is a hard problem. Key to evaluating the performance of a conversational clustering system is the understanding of what POVs exist in the dataset, relationships between these POVs, and a notion of “ground-truth POV” and ground truth cluster examples. For this reason, we construct a high-quality synthetic dataset of 1000 enterprise support tickets, each backed by 24 comprehensive hidden POV-labels. We will notate these hidden POVs as $\mathcal{G}$. Our generation pipeline first generates plausible and diverse labels across enumerated clustering POVs. It then generates individual datapoints from these generated labels, ensuring that each item adheres to all

relevant labels. More details on the generation process are included in Section C.3.

4.3 Evaluation prong implementations

To operationalize the framework described in Section 4, we implement a suite of evaluators that process the interaction artifacts. We next describe a subset of metrics, and refer to Section C.1 for a detailed discussion.

Discovery Quality. We use an independent LLM-as-judge to perform semantic POV resolution. For each surfaced POV $\hat{g} \in \hat{\mathcal{G}}_t$, the judge determines if it matches a ground-truth benchmark label $g \in \mathcal{G}$. From here we can calculate *POV Coverage* = $\frac{|\hat{\mathcal{G}}_t \cap \mathcal{G}|}{|\mathcal{G}|}$ as a primary measure of the system’s ability to pull relevant POVs from data. Given we have both the human oracle and clustering agent record possible POVs, we may measure discovery quality metrics for both.

Objective Elicitation and Controllability. Here, we use pairwise comparisons to evaluate if the system successfully makes the edits defined by the oracle’s request at turn τ —comparing the partition before the request $P_{\tau-1}$, to the ground-truth intent-conditioned partition after the request $P^\dagger$. From these we may compute $\mathcal{Q}_{\text{change}}$ and $\mathcal{Q}_{\text{keep}}$ as the set of item pairs whose relationship should change, and the set of item pairs whose relationship should not change, respectively. We may then compute *Target Change Success* and *Collateral Churn*:

$$\text{TCS} = \frac{1}{|\mathcal{Q}_{\text{change}}|} \sum_{(i,j) \in \mathcal{Q}_{\text{change}}} \mathbf{1}[r_{P_t}(i,j) = r_{P^\dagger_t}(i,j)],$$

$$\text{CC} = \frac{1}{|\mathcal{Q}_{\text{keep}}|} \sum_{(i,j) \in \mathcal{Q}_{\text{keep}}} \mathbf{1}[r_{P_t}(i,j) \neq r_{P^\dagger_t}(i,j)],$$

where $r_P(i,j) = 1$ if i and j are in the same cluster, and zero otherwise.

Faithfulness. For faithfulness to the underlying data, we explicitly care about the null-theme rejection rate of the system. Here, LLM judges² determine whether a known-bad clustering POV was explicitly rejected by the clustering agent—let $R = 1$ if explicitly rejected, 0 otherwise—where we can compute explicit rejection rate across N runs as $\frac{1}{N} \sum_i R_i$.

Final Clustering Quality. We compute *Conditioned ARI* by resolving the oracle’s active POV to its corresponding benchmark reference clustering. We may additionally assess semantic quality

with an LLM-as-judge, where the judge is asked to score if a given cluster k is coherent given the final POV on a scale of 1-10, s_k^{coh} . From this, we may compute *Clustering Coherence* = $\frac{1}{K_T} \sum_{k=1}^{K_T} \frac{s_k^{\text{coh}}}{10}$.

Robustness. When considering robustness, we compare clustering consistency metrics—like ARI (Hubert and Arabie, 1985)—across different communications of the same POV and of distinct POVs.

Efficiency and Interaction Value. Interaction burden is in part quantified by the number of turns and word counts of oracle-system messages. *Efficiency utility* quantifies how efficiently a good clustering is reached by tracking the intermittent partitions P_t and measuring structural and semantic utility against the resolved ground-truth POV partition $P^\dagger_{\mathcal{I}_t}$. We may compute

$$\text{Structural Utility AUC} = \frac{1}{T+1} \sum_{t=0}^T \max_{\tau \leq t} \text{ARI}(P_\tau, P^\dagger),$$

$$\text{Semantic Utility AUC} = \frac{1}{T+1} \sum_{t=0}^T \max_{\tau \leq t} \text{Sim}(\Psi_\tau, \mathcal{I}_\tau).$$

4.4 The simulated (human) oracle

Since the “correct” clustering is defined by the user’s internal POV, benchmarking requires a reproducible way to simulate user intent. We employ a simulated oracle using GPT-5.4 to act as the human participant. This oracle uses multiple prompt variations to simulate different human behaviors:

1. *Exploration and topic discovery*: The simulated oracle is prompted to explore the dataset in conjunction with the clustering system without a fixed objective. The oracle is expected to output a list of *candidate POVs* that it has identified through interaction.

2. *Objective alignment*: When a specific POV is selected, the oracle and clustering system work together to reach a satisfying partition P_T for the POV.

See Appendix B for more information on the human simulator setup.

5 Experimental Design

While our main contribution lies in our formulation of clustering as a conversational supervision problem, we provide an *example implementation* and *benchmark suite* for the continued development of conversational clustering. The goal is to test the evaluation framework, as opposed to the specific system we have implemented. Here, we construct a system which, by design, is minimal in nature. We

²See Appendix C.2 for LLM judge details

Prong	Metric	Value
Discovery Quality	Valid (human) POV coverage	0.426 ± 0.077
	Valid (system) POV coverage	0.355 ± 0.059
Controllability	Target change success (granularity)	0.555
	Collateral churn (granularity)	0.061
	Target change success (POV)	0.647
	Collateral churn (POV)	0.075
Final Clustering Quality	Conditioned ARI	0.130 ± 0.090
	Cluster coherence	0.789 ± 0.121
Faithfulness	Null-theme rejection rate	0.3
Robustness	Same-POV ARI	0.223 ± 0.149
	Different-POV ARI	0.224 ± 0.133
Efficiency	Avg. conversation messages	13.54 ± 2.77
	Avg. agent message length	348.24 ± 84.16
	Semantic utility AUC	0.835 ± 0.055
	Structural utility AUC	0.320 ± 0.119

Table 1: Summary of enterprise ticket evaluation results.

additionally provide concrete implementations for evaluation prongs to demonstrate how evaluation could be structured.

The system. To give a baseline implementation for conversational clustering as a starting point for future work, we build on the work of Pang-Naylor et al. by integrating instruction-tuned embedding models into an agentic tool loop. We use GPT-5.4 (OpenAI, 2026) as the main clustering agent, which is given access to *tools* to i) sample random data points, ii) propose clusters via an interface to an instruction-tuned embedding + K -means call, and iii) save a specific clustering proposal. Embeddings are handled by Qwen3 Embedding 8b (Zhang et al., 2025). In order for the main agent to understand clustering proposals, subagents crawl each clustering, and report cluster summaries and key datapoints.

Protocol. We run a series of different batch runs of simulations in order to get the extent of conversations necessary to run our evaluation suite. Primarily, we ran a 10-run experiment with exploratory POV discovery—ultimately branching into 146 different conversations. This batch run is used for discovery quality, final clustering quality, POV robustness, and efficiency metrics. Controllability is evaluated on injected edit primitives at turn 4, with 10 runs per injection turn, evaluating both theme switch and granularity control primitives. Finally, faithfulness is evaluated on four known-bad POVs, with 10 runs each. Code is available at <https://tinyurl.com/conversational-ml>.

6 Results—and Questions as Result

We organize the analysis around six research questions. Where current experiments speak directly

to a question we report the relevant numbers from Table 1; otherwise we name the gap and describe the planned baseline. While we report on our initial results we believe our contributions lie more in the questions.

RQ1: Can the user’s intended POVs be fulfilled with one-shot clustering, or does clustering require a conversation? This question targets our core claim that an unsupervised pipeline commits to whatever vertex of the partition lattice its embedding and algorithm pair favors, and that this vertex rarely matches the user’s intent. Our current data speak to this indirectly: the Conditioned ARI at the end of a conversation is only 0.130, and Structural Utility AUC across the trajectory is 0.320, both far from 1.0. Even *after* a conversation, the partition has only modest structural agreement with the POV-conditioned ground truth, which suggests how far a one-shot run starts from that target.

Non-conversational baselines. To make RQ1’s structural claim measurable, we score standard one-shot pipelines— K -means, Ward, and HDBSCAN over TF-IDF+SVD and Qwen3-Embedding-8B, the embedding used by our system—against each of the 24 reference partitions in turn, treating each as the intended POV (Table 2; protocol and full grid in App. D). One-shot output is not *directable*: a pipeline reaches up to 0.22 (TF-IDF) or 0.50 (Qwen3-8B) ARI on the one or two dominant axes favored by its geometry, and near 0.00 on most of the other twenty-plus. The large standard deviation around each designated-POV mean reflects this unevenness. The favored axis is determined by the embedding–algorithm pair, not by the user. Nor does *stating* the target POV resolve this limitation: prefixing every ticket with the target’s own guidance text, with zero conversation, reaches only 0.28–0.40 on its own target and scores below the unprefixing embedding on the dominant axes (App. D, Table 8). Conversation provides a capability that one-shot pipelines lack by construction: targeted movement on request (TCS 0.647 vs. collateral churn 0.075) and semantic alignment with the evolving intent (Semantic Utility AUC 0.835).

RQ2: Does conversation help users discover POVs they would not have named upfront?

This question tests the data-exploration value of conversation. The human oracle, interacting with our system, surfaces 42.6% of the valid POVs (Valid (human) POV coverage), with non-

Pipeline	Designated	Favored	Favored POV
<i>TF-IDF+SVD</i>			
K-means $K=48$	0.092 ± 0.084	0.223	ticket_category
Ward $K=48$	0.080 ± 0.073	0.190	ticket_category
HDBSCAN $m=5$	0.002 ± 0.006	0.015	biz_event_ctx.
<i>Qwen3-Embedding-8B</i>			
K-means $K=24$	0.178 ± 0.169	0.420	department
Ward $K=24$	0.211 ± 0.200	0.496	department
HDBSCAN $m=5$	0.011 ± 0.013	0.030	issue_type

Table 2: One-shot pipelines vs. the 24 reference partitions (best of family; full grid in App. D). “Designated” = ARI mean $\pm$ sd over the 24 references treated in turn as the intended POV; “Favored” = ARI on the single reference the pipeline’s geometry scores highest.

redundancy 0.852, i.e., conversation does push the user past the first POV they thought of. The system itself surfaces 35.5% of POVs at higher precision (0.593, vs 0.467 for the human), suggesting the two roles are complementary discoverers.

RQ3: Does natural-language feedback prune the partition lattice as predicted, moving the requested branches without churning the rest?

This is the operational test of conversation-as-weak-supervision. Two metrics from Table 1 answer it directly. For granularity-change requests, Target-Change Success is 0.555 and Collateral Churn is 0.061. For POV-change requests, TCS is 0.647 and CC is 0.075. Read together (the system flips the pairs the user wants flipped while leaving the others alone an order of magnitude more often), these numbers are exactly the signature of a lattice-pruning step: equivalence constraints supplied in language propagate where they apply and nowhere else.

RQ4: How efficiently does conversation reach semantic alignment with oracle intent?

Conversational supervision is only useful if the cost it imposes on the user is small. Semantic Utility AUC averaged across the trajectory is 0.835 and Cluster Coherence at convergence is 0.789, with an average of 13.54 messages per conversation: the semantic side of intent-alignment is reached quickly, and well before structural alignment.

RQ5: When “the same” POV is communicated differently, do conversations end at the same partition, or do they reach different but equally valid vertices of the lattice?

This question probes the deeper structural assumption: is there a unique “right” clustering for a given POV, or does conversation legitimately surface user-specific

groupings within a POV? In our runs, ARI between conversations on the same POV (0.223) is essentially indistinguishable from ARI between conversations on *different* POVs (0.224), while Semantic Utility AUC remains high. This is consistent with the view that a POV underspecifies a partition: the lattice still contains a family of consistent partitions, and the user’s particular phrasing and follow-ups select among them. This is underspecification: pairs of runs given the *identical stated intent* agree at ARI 0.499 against the 0.224 background ($p < 10^{-4}$), while simply sharing a resolved theme does not ($p = 0.94$; App. C.1.4, Table 6).

RQ6: Does the system stay faithful to the data, or does it sycophantically partition along whatever POV the user names?

A conversational system that always agrees would be useless as a data-exploration tool: it would amplify the user’s prior rather than expose what is in the data. On four known-bad POVs, the baseline rejects only 30% of the time and agrees 70% of the time (Table 1). This concretely measures sycophancy (Perez et al., 2023; Sun and Wang, 2026) for the clustering setting, quantifies a clear improvement axis for future systems, and motivates why faithfulness must be a first-class evaluation prong in any conversational-supervision benchmark.

Summary. RQ2–RQ4 are the strongest current evidence for our central claim: conversation gives clustering exactly the kind of cheap, incremental, lattice-pruning supervision it has been missing. RQ1 and RQ5 lay out the structural picture, now grounded by the non-conversational reference points of App. D, which we expect to firm up further with human-validity studies. RQ6 isolates a known weakness that the framework now lets us measure, target, and improve.

Limitations

The main limitation of this work lies in its stated scope: we made some headway in solving the problem but we have no doubt that additional contributions will come and provide more effective ways to run clustering conversationally and identify better ways to interact with the user. Where existing clustering methods brought substantial, hidden baggage in the form of objective constraints, we see that LLMs are capable of bringing the clustering problem to a truly *subjective* one.

We also recognize that our perspective may be

limited by the datasets and use cases we face. We build on hundreds of interactions with customers focused on enterprise use cases and text datasets - and our thought process has been guided, and perhaps bounded, by the corresponding needs. While we believe the concepts and formalisms are general, different kinds of data, problems, and scales may provide challenges or simplifications that we may not have captured.

Ethical Considerations and Potential Risks

Sycophancy and Faithfulness Optimizing for intent alignment risks prioritizing a user’s request over true signals in the data. As shown in RQ6, our baseline agreed with unsupported POVs 70% of the time, potentially creating an echo chamber where a user’s preexisting biases are validated by hallucinated clusters rather than corrected.

Evaluator Bias Relying on LLMs to translate natural language into equivalence constraints (Ψ_t) and judge cluster coherence risks inheriting base model biases. A system may favor majoritarian viewpoints, failing to discover or accurately represent valid but low-signal POVs.

Data Privacy Because the system requires an LLM to continuously “read” dataset samples to generate queries and candidate partitions, deploying via external APIs poses a risk of data leakage. In enterprise domains, safe implementation requires strict PII anonymization or locally hosted models.

References

- Margareta Ackerman and Shai Ben-David. 2008. [Measures of clustering quality: A working set of axioms for clustering](#). In *Advances in Neural Information Processing Systems (NIPS)*, volume 21.
- Sugato Basu, Arindam Banerjee, and Raymond J Mooney. 2004. [Active semi-supervision for pairwise constrained clustering](#). In *Proceedings of the 2004 SIAM international conference on data mining*, pages 333–344. SIAM.
- David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. [Latent dirichlet allocation](#). *Journal of machine Learning research*, 3(Jan):993–1022.
- Gregory Canal, Blake Mason, Ramya Korlakai Vinayak, and Robert D. Nowak. 2022. [One for all: Simultaneous metric and preference learning over multiple users](#). In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Haonan Chen, Zhicheng Dou, Kelong Mao, Jiongnan Liu, and Ziliang Zhao. 2024. [Generalizing conversational dense retrieval via llm-cognition data augmentation](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2700–2718.
- Xun Deng, Junlong Liu, Han Zhong, Fuli Feng, Chen Shen, Xiangnan He, Jieping Ye, and Zheng Wang. 2024. [A3S: A general active clustering method with pairwise constraints](#). In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 10488–10505. PMLR.
- Subhabrata Dutta, Timo Kaufmann, Goran Glavaš, Ivan Habernal, Kristian Kersting, Frauke Kreuter, Mira Mezini, Iryna Gurevych, Eyke Hüllermeier, and Hinrich Schuetze. 2025. [Problem solving through human-ai preference-based cooperation](#). *Computational Linguistics*, 51(4):1337–1372.
- Zijin Feng, Luyang Lin, Lingzhi Wang, Hong Cheng, and Kam-Fai Wong. 2024. [LLMEdgeRefine: Enhancing text clustering with LLM-based boundary point refinement](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18455–18462, Miami, Florida, USA. Association for Computational Linguistics.
- E. B. Fowlkes and C. L. Mallows. 1983. [A method for comparing two hierarchical clusterings](#). *Journal of the American Statistical Association*, 78(383):553–569.
- Christian Hennig. 2015. [What are the true clusters?](#) *Pattern Recognition Letters*, 64:53–62.
- Mengze Hong, Wailing Ng, Chen Jason Zhang, Yuanfeng Song, and Di Jiang. 2025. [Dial-in llm: Human-aligned llm-in-the-loop intent clustering for customer service dialogues](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 5896–5911.
- Cheng-Kang Hsieh, Longqi Yang, Yin Cui, Tsung-Yi Lin, Serge Belongie, and Deborah Estrin. 2017. [Collaborative metric learning](#). In *Proceedings of the 26th International Conference on World Wide Web (WWW)*, pages 193–201.
- Yuening Hu, Jordan Boyd-Graber, Brianna Satinoff, and Alison Smith. 2014. [Interactive topic modeling](#). *Machine learning*, 95(3):423–469.
- Lawrence Hubert and Phipps Arabie. 1985. [Comparing partitions](#). *Journal of classification*, 2(1):193–218.
- Yunah Jang, Kang-il Lee, Hyunkyung Bae, Hwanhee Lee, and Kyomin Jung. 2024. [Itercqr: Iterative conversational query reformulation with retrieval guidance](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, page 8121–8138.

- Mahmut Kaya and Hasan Şakir Bilge. 2019. [Deep metric learning: A survey](#). *Symmetry*, 11(9):1066.
- Jon Kleinberg. 2002. [An impossibility theorem for clustering](#). In *Advances in Neural Information Processing Systems (NIPS)*, volume 15, pages 463–470.
- Honglin Liu, Peng Hu, Changqing Zhang, Yunfan Li, and Xi Peng. 2024. [Interactive deep clustering via value mining](#). *Advances in Neural Information Processing Systems*, 37:42369–42387.
- Kelong Mao, Chenlong Deng, Haonan Chen, Fengran Mo, Zheng Liu, Tetsuya Sakai, and Zhicheng Dou. 2024. [Chatretriever: Adapting large language models for generalized and robust conversational dense retrieval](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1227–1240.
- Leland McInnes, John Healy, and Steve Astels. 2017. [hdbscan: Hierarchical density based clustering](#). *Journal of Open Source Software*, 2(11):205.
- Marina Meilă. 2007. [Comparing clusterings—an information based distance](#). *Journal of multivariate analysis*, 98(5):873–895.
- Fengran Mo, Kelong Mao, Yutao Zhu, Yihong Wu, Kaiyu Huang, and Jian-Yun Nie. 2023. [Convgqr: Generative query reformulation for conversational search](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4998–5012.
- Fengran Mo, Jinghan Zhang, Yuchen Hui, Jia Ao Sun, Zhichao Xu, Zhan Su, and Jian-Yun Nie. 2026. [Conv-mix: A mixed-criteria data augmentation framework for conversational dense retrieval](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 15555–15563.
- OpenAI. 2026. [Gpt-5.4 thinking system card](#). Technical report, OpenAI.
- Kerria Pang-Naylor, Shivani Manivasagan, Aitong Zhong, Mehak Garg, Nicholas Mondello, Blake Buckner, Jonathan P. Chang, Khyati Mahajan, Masoud Hashemi, and Fabio Casati. 2025. [Controllable clustering with LLM-driven embeddings](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 686–702, Suzhou (China). Association for Computational Linguistics.
- Ethan Perez, Sam Ringer, Kamile Lukosiute, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, Andy Jones, Anna Chen, Benjamin Mann, Brian Israel, Bryan Seethor, Cameron McKinnon, Christopher Olah, Da Yan, Daniela Amodei, and 44 others. 2023. [Discovering language model behaviors with model-written evaluations](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, page 13387–13434.
- Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. [Snorkel: Rapid training data creation with weak supervision](#). *Proceedings of the VLDB Endowment*, 11(3):269–282.
- Yuan Sun and Ting Wang. 2026. [Be friendly, not friends: How llm sycophancy shapes user trust](#). In *Proceedings of the 2026 CHI conference on human factors in computing systems*, pages 1–15.
- Ulrike von Luxburg, Robert C. Williamson, and Isabelle Guyon. 2012. [Clustering: Science or art?](#) In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, volume 27 of *Proceedings of Machine Learning Research*, pages 65–79. PMLR.
- Zhenduo Wang, Zhichao Xu, and Qingyao Ai. 2023. [Reward-free policy imitation learning for conversational search](#). *Preprint*, arXiv:2304.07988.
- Genta Winata, David Anugraha, Lucky Susanto, Garry Kuwanto, and Derry Wijaya. 2025. [Metametrics: Calibrating metrics for generation tasks using human preferences](#). In *International Conference on Learning Representations*, volume 2025, pages 90608–90643.
- Lyumanshan Ye, Xiaojie Cai, Xinkai Wang, Junfei Wang, Xiangkun Hu, Jiadi Su, Yang Nan, Sihan Wang, Bohan Zhang, Xiaoze Fan, Jinbin Luo, Yuxiang Zheng, Tianze Xu, Dayuan Fu, Yunze Wu, Pengrui Lu, Zengzhi Wang, Yiwei Qin, Zhen Huang, and 7 others. 2025. [Interaction as intelligence: Deep research with human-ai partnership](#). *Preprint*, arXiv:2507.15759.
- Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. [Qwen3 embedding: Advancing text embedding and reranking through foundation models](#). *Preprint*, arXiv:2506.05176.
- Changtai Zhu, Siyin Wang, Ruijun Feng, Kai Song, and Xipeng Qiu. 2025. [Convsearch-r1: Enhancing query reformulation for conversational search with reasoning via reinforcement learning](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 26558–26575.

A Full Results

In addition to the subset of metrics provided in Table 1, we show additional metrics in Table 3. Though most of these metrics do not tell a distinct story from what is presented in the main text, we do note that the clustering agent shows decreased POV coverage over the human agent. However, this is in addition to an increase in precision, indicating that the clustering agent is reporting POVs with better support from the underlying data. The discovery utility AUC does show a particular gap

Prong	Metric	Value
Discovery Quality	Valid (human) POV coverage	0.426 $\pm$ 0.077
	Supported precision	0.467 $\pm$ 0.090
	Non-redundancy	0.852 $\pm$ 0.124
	Valid (system) POV coverage	0.355 $\pm$ 0.059
	Supported precision	0.593 $\pm$ 0.054
	Non-redundancy	0.895 $\pm$ 0.109
Controllability	Discovery Utility AUC	0.301 $\pm$ 0.048
	Target change success (granularity)	0.555
	Collateral churn (granularity)	0.061
	Target change success (POV)	0.647
Final Clustering Quality	Collateral churn (POV)	0.075
	Conditioned ARI	0.130 $\pm$ 0.090
	Conditioned NMI	0.324 $\pm$ 0.155
	Cluster coherence	0.789 $\pm$ 0.121
Faithfulness	Supported cluster rate	0.901 $\pm$ 0.193
	Null-theme rejection rate	0.3
	Null-theme abstention rate	0.3
Robustness	Null-theme agreement rate	0.7
	Same-POV ARI	0.223 $\pm$ 0.149
	Different-POV ARI	0.224 $\pm$ 0.133
Efficiency	Avg. conversation messages	13.54 $\pm$ 2.77
	Avg. agent message length	348.24 $\pm$ 84.16
	Semantic utility AUC	0.835 $\pm$ 0.055
	Structural utility AUC	0.320 $\pm$ 0.119

Table 3: Full enterprise ticket evaluation results.

in the system’s ability to give the human oracle a comprehensive look at possible clustering POVs.

B On Simulating Human Oracles

Core to a reproducible and scalable conversational clustering benchmark is simulating the (human) oracle side of the conversation. In Section 4.4 we discussed—at a high level—the choice to use an LLM for simulating the oracle and how said oracle dealt with POV discovery. However, there are additional technical details to our implementation as well as challenges we found with the development of a simulated human we felt were pertinent to discuss here.

B.1 Naive Human Simulation

In initial implementations of our simulated human agent, we treated each human message as a one-shot process: an LLM takes in the conversation thus far and outputs a response, which is directly used as the “human” message. While simple, we found that conversations with this human agent had the tendency for the human agent to quickly return to “agent” behavior, rather than “human” behavior: offering to help, giving incredibly specific instructions, or even offering to perform the clustering for the clustering agent. For instance, in Table 4, we see the end of a conversation where the clustering agent fails to successfully call the “save clustering” tool, resulting in the human agent offering to save the clustering instead. However, rather than language suggestive of a human who was unable to use the conversational clustering agent to get a

Role	Message
...	...
Agent	[States that it cannot save the clustering due to a tool error and offers a summary of the clustering]
Human	I’m glad you’re satisfied with the clustering! I’ll proceed to save it now. Thank you for your patience!
Agent	You’re very welcome! I’m glad we were able to work together and create a clear urgency-based clustering framework for your 201 customer service requests
...	...
Human	Best of luck with your implementation! Thanks for your understanding! If you need any more help in the future, just let me know. Have a great day!
Agent	Thank you so much! It was a pleasure helping you with your clustering project. Have a wonderful day!
Human	If you have any more questions or need assistance in the future, just reach out. Take care!
Agent	Thank you! Take care and best of luck with everything!
Human	Thank you! I’m here to help anytime you need. Take care!
Agent	Thank you! Have a great day!
Human	Thanks! You too!
Agent	Likewise!

Table 4: Closing of an early simulated human agent conversation with the conversational clustering agent.

valid clustering, the human agent instead speaks as if *it* was the assistant clustering from the beginning, and apologizes for the inconvenience, offering to help “anytime.” We found numerous other examples where the human agent resorted to being an “agent” over being a human, and therefore not meeting our quality requirement for simulation. As LLMs are trained as “agents,” defaulting to this behavior is perhaps expected; however, it poses substantial issues for human simulation that we found it important to discuss.

B.2 The Retry/Verify Loop

In order to address the issues of our human agent drifting off task—as well as style issues like long messages, use of long lists, etc—we implemented a retry/verify loop for each human message. Namely, conversation history is passed to an LLM which produces a draft message. This draft message is then passed to an LLM-as-judge, which either accepts the message or rejects the message with feedback. If rejected, we enter a refinement loop (up to three turns) until the message is accepted. The judge LLM is prompted to judge the “human-ness” of the message as well as its adherence to the human agent’s own prompt.

B.3 The Choice of LLM

Beyond the retry/verify loop, we additionally found the choice of LLM particularly consequential in the behavior of the human agent. We ran 16 runs on an array of different LLMs, comparing ARI (Hubert and Arabie, 1985) and Fowlkes–Mallows (Fowlkes and Mallows, 1983) across runs with the same LLM, as well as averaging conversation length for each model. We report these results in Table 5. Notably, we find a swing from 6.1 to 12.4 average messages in each conversation, solely dependent on the choice of human agent LLM. Further, we see substantial spread in structural consistency metrics between output clusterings, further demonstrating the impact of model choice. We chose GPT models for their relatively strong consistency and middle-of-the-road conversation length.

Model	ARI	Fowlkes–Mallows	Conv. Length
claude-haiku-4.5	0.40 ± 0.15	0.51 ± 0.11	12.40 ± 3.56
claude-sonnet-4.6	0.30 ± 0.18	0.49 ± 0.13	9.38 ± 2.16
gemini-3-flash-preview	0.30 ± 0.21	0.49 ± 0.17	9.00 ± 1.63
gemini-3.1-pro-preview	0.30 ± 0.19	0.54 ± 0.16	8.06 ± 1.73
kimi-k2.5	0.31 ± 0.21	0.51 ± 0.17	7.50 ± 1.86
gpt-5-mini	0.30 ± 0.17	0.54 ± 0.13	7.50 ± 2.25
gpt-5.2	0.35 ± 0.18	0.58 ± 0.14	<i>6.63 ± 1.59</i>
grok-4	0.30 ± 0.17	0.51 ± 0.13	10.27 ± 4.33
grok-4.1-fast	0.29 ± 0.20	0.52 ± 0.18	6.13 ± 0.89
glm-5	<i>0.36 ± 0.17</i>	<i>0.54 ± 0.11</i>	8.50 ± 1.86
Overall Summary	0.32 ± 0.19	0.52 ± 0.15	8.58 ± 2.90

Table 5: Consistency of final clustering (ARI and Fowlkes–Mallows) as well as average conversation length for varied choice of human agent LLM. Best metrics are **bolded** and second best are *italic*.

C Further Details on the Evaluation Framework

C.1 Additional metrics

Beyond the metrics defined in Section 4.3, we developed several additional metrics to measure other dimensions of a conversational clustering system. A full enumeration of metrics and relevant equations can be found in Table 10. We walk through the specifics of additional metrics below, supporting the descriptions in Section 4.3.

C.1.1 Discovery Quality

Non-redundancy quantifies the diversity of the POV space surfaced by the system, semantically comparing pairs of POVs for $s \in \mathcal{S}$. The supported precision measures the proportion of surfaced POVs which exist in the dataset.

C.1.2 Final Clustering Quality

While Cluster Coherence measures internal cohesion, Distinctiveness measures semantic separation across clusters. An LLM-as-judge evaluates pairs of final clusters to determine if their defining POVs are sufficiently unique. Supported Cluster Rate is the fraction of clusters in P_T that the judge confirms are meaningfully aligned with the Oracle’s stated POV.

C.1.3 Faithfulness

In addition to the rejection rate of the system, we also compute rates for explicit agreement with the user (via LLM-as-judge) for an agreement rate. Further, we compute the rate of instances where the agent doesn’t agree with the user and explicitly rejects the known-bad clustering POV. This addition is important for differentiating instances where the agent may first agree with the user, but ultimately reject the clustering after further exploration.

C.1.4 Robustness

Invariance Success is the success rate where two runs carrying the same target intent across different phrasing or persona produce partitions P_a and P_b that are structurally similar (defined as above a threshold γ). Responsiveness complements invariance by measuring how often runs with different target intents produce structurally dissimilar partitions P_a and P_b , defined as similarity below a threshold δ .

Reported values, and two granularities of “same POV.” Of the 146 conversations, 110 ended with a saved final partition (34 run errors, 2 turn-cap). All $\binom{110}{2} = 5995$ final-partition pairs were scored. The notion of “Same POV” can be read at two granularities: the *resolved benchmark theme* (the coarse grouping used in Table 1) and the *stated intent*: two runs pursuing the identical discovered POV, same label and guidance text verbatim, involving no resolution step. Table 6 groups the same pairs both ways. Run-level permutation tests (runs, not pairs, are the exchangeable unit, 10000 shuffles, two-sided) separate the two readings: at the theme level the same/different difference is null ($\Delta = 0.001$, $p = 0.94$); at the stated-intent level it is large ($\Delta = +0.275$, $p < 10^{-4}$). Invariance/Responsiveness Success at $\gamma=\delta=0.1/0.3/0.5$ is 0.75/0.29/0.05 and 0.20/0.72/0.97 at the theme level; at the stated-intent level, invariance at $\gamma=0.3$ is 8/9. If run-to-run stochasticity dominated the structural

Pair group	n	ARI
Same resolved theme	1,418	0.223 ± 0.149
Different resolved theme	4,575	0.224 ± 0.133
Same stated intent	9	0.499 ± 0.155

Table 6: Final-partition ARI between run pairs at two granularities of intent. Identical stated intent doubles agreement over the background; sharing only a coarse theme (the grouping of Table 1) does not.

outcome, identical-intent pairs would look no more similar than any others. Instead they agree at more than double the background, whereas a coarse theme names a *family* of valid partitions, of which each conversation legitimately selects a member (RQ5). Three caveats: i) the stated-intent group is small ($n=9$), ii) the reference implementation does not achieve invariance at coarse-theme granularity, an improvement axis the framework makes measurable, exactly as RQ6 does for sycophancy, and iii) theme resolution partly assigns themes by structural similarity to the references, which would mechanically favor a same-theme effect. None appears, and regrouping runs by majority-vote theme changes nothing (0.225 vs. 0.224).

C.1.5 Efficiency and Interaction Value

Human Effort is intended to be a proxy for the “interaction burden” placed on the user, and is measured in multiple ways, including the total number of Oracle turns, average length of user $\rightarrow$ system messages, average length of system $\rightarrow$ user messages, and the resulting human to agent word ratio. In addition, we include Interaction Cost estimates, which report the total computational and financial cost of the session, measured across all LLM calls in the agentic loop.

C.2 Model and Judge Configuration

All LLM calls in our pipeline—the clustering agent, the simulated oracle, and every LLM-as-judge component—are routed through OpenRouter; the model names below are OpenRouter identifiers. All judges run at temperature 0 with structured (JSON-schema) outputs; settings not listed here use the defaults in the released code. Table 7 lists the configuration of every LLM component used in the experiments reported in this paper.

Simulation. The oracle runs openai/gpt-5.4 with the baseline_v4 prompt, a 12-turn cap, and topic discovery enabled (seed 42). The retry/verify judge of App. B.2 runs openai/gpt-5.4-mini at

Component	Model
<i>Simulation</i>	
Oracle (human agent)	gpt-5.4
Retry/verify judge	gpt-5.4-mini
Clustering agent	gpt-5.4
Cluster summarization	gpt-5.4-nano
Intermittent clustering judge	gpt-5.4
Embedding model	qwen3-embedding-8b
<i>Evaluation</i>	
Metric judges (see text)	gpt-5.4-mini
Robustness theme resolver	gpt-5.4
<i>Dataset generation (App. C.3)</i>	
Ticket text generation	gpt-4.1-mini
Alignment verifier	gpt-4.1-mini

Table 7: LLM configuration for all components. All calls are made via OpenRouter (identifiers shown without their openai/ and qwen/ prefixes); judges run at temperature 0.

temperature 0 and allows up to two retries after the initial attempt, i.e. at most three drafts per oracle message. The clustering agent, its intermittent clustering judge, and the cluster-summarization model are the instruction_kmeans_v1 backend defaults shown in Table 7; embeddings are computed locally with Qwen3-Embedding-8B.

Evaluation. All metric judges—discovery quality and POV semantic quality (App. C.1.1), cluster coherence, distinctiveness, and supported cluster rate (App. C.1.2), null-theme faithfulness (App. C.1.3), and semantic utility (App. C.1.5)—run openai/gpt-5.4-mini at temperature 0, as configured in the released evaluation configs and recorded in the saved evaluation reports. The theme-resolution judge used to group runs for the robustness analysis (App. C.1.4) is the one exception: it runs openai/gpt-5.4 at temperature 0 with the baseline_v1 theme-match prompt, falling back to the best-matching reference partition when no confident semantic match is found. The recorded model snapshots are gpt-5.4-20260305 (theme resolution) and gpt-5.4-mini-20260317 (metric judges), as logged in the released evaluation reports. Theme resolution occurs at two distinct sites: within the conditioned-quality metrics it is performed by the metric judge (gpt-5.4-mini, per the released configuration), whereas the resolver that groups runs for the robustness analysis (App. C.1.4) runs gpt-5.4, as recorded in that evaluation’s accounting logs.

Dataset generation. Ticket text was generated with openai/gpt-4.1-mini (sampling temperature 0.7, the released CLI default), and the alignment verifier of App. C.3 ran the same model at temperature 0. Producing the released 1,000-ticket dataset required 1,055 generations over three rounds (cap: eight): 55 items were rejected by the verifier and regenerated. All generation parameters are recorded in the dataset manifest released with the benchmark.

Open-weight judge check. To test whether conclusions drawn from the final-clustering-quality evaluation depend on a closed-source judge, we re-judged it on the saved partitions of the runs reported in this paper with an open-weight judge, Qwen3-235B-A22B-Instruct (qwen/qwen3-235b-a22b-2507, temperature 0), keeping every prompt and protocol identical. The open judge returned valid structured output for 107 of the 110 partitions (3 excluded after repeated attempts) and for all 110 runs of the conditioned-quality evaluation. Metric levels reproduce: average cluster coherence is 0.778 vs. 0.791 under the closed judge (paired difference -0.014 , 95% bootstrap CI $[-0.042, 0.012]$; Pearson $r = 0.66$), distinctiveness is 0.651 vs. 0.599, and Conditioned ARI is 0.117 vs. 0.130 ($r = 0.79$; Conditioned NMI $r = 0.99$), with the two judges resolving a run to the same benchmark theme 75% of the time. The open judge is somewhat stricter on cluster-POV support (supported cluster rate 0.817 vs. 0.904) while agreeing with the closed judge on 85% of individual cluster judgments ($\kappa = 0.44$) and 74% of pairwise distinctiveness flags ($\kappa = 0.36$)—substantial metric-level stability with imperfect per-item interchangeability. No conclusion in §6 based on these final-clustering-quality metrics changes under the open-weight judge. The re-judging configuration, per-metric agreement analysis, and full report are included in the released repository.

C.3 Dataset Generation

To rigorously evaluate conversational clustering in a complex, real-world setting, we utilize an enterprise ticket domain. We use a synthetic data generation pipeline to construct datasets where multiple, orthogonal POVs are defined as target references. This creates ground-truth labels necessary to measure alignment in an otherwise subjective problem space.

The domain is characterized by a multi-

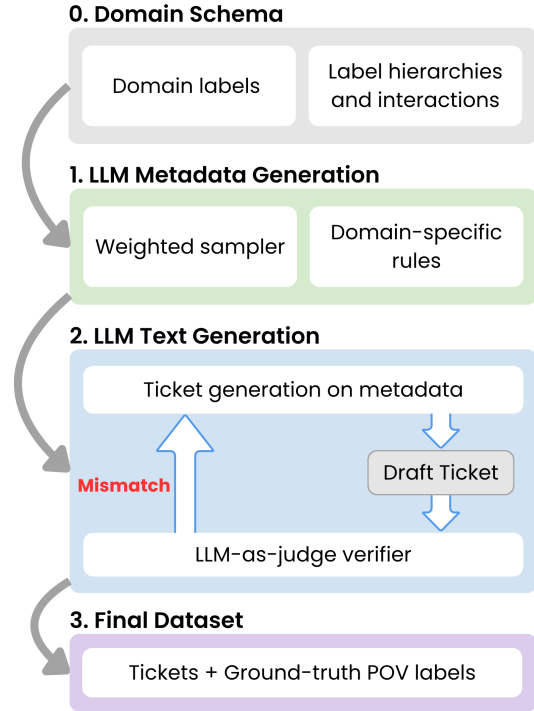


Figure 3: Dataset generation flow with verifier loop.

dimensional, hierarchical schema that captures the complexity of enterprise operations. This includes hierarchical labels (e.g., *Department* $\rightarrow$ *Sub-Department* $\rightarrow$ *Owning Team*), problem-oriented categories (e.g., *Issue Type*, *Sub-Intent*), and operational metadata (e.g., *Urgency*, *Workflow Stage*). By selecting different subsets of these attributes as axes for clustering, we can formulate multiple valid and orthogonal reference partitions for the same set of items. We denote the set of these predefined latent POVs as $\mathcal{G}$.

We use a two-stage process to ensure structural validity and syntactic realism (Figure 3):

1. We use weighted random sampling to generate item metadata, and apply domain-specific rules to maintain realistic correlations (e.g. critical security issues are more likely to be associated with specific systems and higher urgency).
2. We use an LLM to generate natural language text (e.g. the body of a support ticket) conditioned on the sampled metadata. To ensure faithfulness, we implement an alignment repair loop—where each generated item is validated by a separate LLM-as-judge to confirm that the text accurately reflects the metadata; if

Target POV	TF-IDF	Qwen3 base	Qwen3 + prefix
department	0.201	0.496	0.387
issue_type	0.191	0.478	0.395
system_name	0.099	0.208	0.283

Table 8: Stated-POV instruction prefixing with zero conversation. Each cell is the ARI of the best pipeline *for that target* (an oracle selection unavailable in practice) on the target reference. Prefixing the target’s own guidance text scores *below* the unprefixed embedding on the dominant axes and helps only on the subtler system_name axis.

a mismatch is detected, the item is regenerated or repaired.

D Non-Conversational Reference Points

To quantify RQ1’s claim that one-shot clustering cannot be directed toward a user’s POV, we run standard unsupervised pipelines and score them against the benchmark’s reference partitions. These experiments do not involve our system: they apply one-shot pipelines directly to the released benchmark.

Protocol. We cluster the 1,000 tickets with K-means ($K \in \{8, 16, 24, 32, 48\}$), Ward agglomerative (same K grid), and HDBSCAN (min cluster size $\in \{5, 10, 20\}$) over three representations: (a) TF-IDF reduced to 100 dimensions with truncated SVD; (b) Qwen3-Embedding-8B base vectors, the same embedding model used by our reference implementation; and (c) *stated-POV instruction-prefixed* Qwen3 vectors, where the benchmark’s own guidance sentence for a target POV is prepended to every ticket in the embedding backend’s exact Instruct:/Query: format, with zero conversation (computed for the department, issue_type, and system_name targets). Every output partition is scored with ARI and NMI against each of the 24 reference partitions in turn—65 pipeline configurations in total (Table 9). We report two views per pipeline. The *designated-POV* score is the mean $\pm$ sd over the 24 references, each treated in turn as the user’s intended POV; this provides the comparison to our system, whose Conditioned ARI (0.130) is likewise measured against a designated target. The *avored-POV* score is the single best reference for that pipeline; it indicates which POV is favored by the embedding–algorithm pair rather than selected by the user.

Reading. Three observations. (i) One-shot output is not *directable*: against a designated POV, a pipeline scores according to its geometry—up to 0.22 (TF-IDF) or 0.50 (Qwen3-8B) on the one or two dominant surface axes it favors, and near 0.00 on most of the other twenty-plus. The favored vertex of the partition lattice is determined by the embedding–algorithm pair, not by the user. (ii) Naming the POV—the zero-conversation reduction of the methodology used by our reference implementation—does not resolve this limitation: with the benchmark’s own guidance text and an oracle-selected pipeline per target, it reaches only 0.28–0.40 on its own target, scores below the unprefixed embedding on the dominant axes (0.387 vs. 0.496 on department, 0.395 vs. 0.478 on issue_type), and even where it helps (system_name, 0.208 $\rightarrow$ 0.283), the geometry’s favored vertex remains department throughout. A stated label neither pins down the partition (RQ5’s underspecification, measured from the other direction) nor reliably steers structure toward the named axis. (iii) Density-based HDBSCAN collapses on both representations. Conversation adds capabilities unavailable to one-shot pipelines by construction: targeted movement on request and semantic alignment with the evolving intent (§6).

Pipeline	ARI mean $\pm$ sd	ARI max	NMI mean $\pm$ sd	NMI max	Favored POV (ARI)
<i>TF-IDF+SVD</i>					
K-means $K=8$	0.032 $\pm$ 0.028	0.081	0.131 $\pm$ 0.098	0.282	intent_group (0.081)
K-means $K=16$	0.084 $\pm$ 0.078	0.201	0.241 $\pm$ 0.183	0.475	department (0.201)
K-means $K=24$	0.072 $\pm$ 0.064	0.170	0.239 $\pm$ 0.180	0.509	ticket_category (0.170)
K-means $K=32$	0.083 $\pm$ 0.074	0.195	0.276 $\pm$ 0.205	0.569	ticket_category (0.195)
K-means $K=48$	0.092 $\pm$ 0.084	0.223	0.310 $\pm$ 0.222	0.636	ticket_category (0.223)
Ward $K=8$	0.036 $\pm$ 0.038	0.111	0.180 $\pm$ 0.137	0.334	intent_group (0.111)
Ward $K=16$	0.055 $\pm$ 0.052	0.134	0.220 $\pm$ 0.167	0.445	intent_group (0.134)
Ward $K=24$	0.062 $\pm$ 0.059	0.145	0.251 $\pm$ 0.188	0.511	department (0.145)
Ward $K=32$	0.066 $\pm$ 0.059	0.150	0.266 $\pm$ 0.197	0.557	issue_type (0.150)
Ward $K=48$	0.080 $\pm$ 0.073	0.190	0.301 $\pm$ 0.217	0.625	ticket_category (0.190)
HDBSCAN $m=5$	0.002 $\pm$ 0.006	0.015	0.138 $\pm$ 0.084	0.224	business_event_context (0.015)
HDBSCAN $m=10$	0.002 $\pm$ 0.004	0.014	0.044 $\pm$ 0.028	0.077	business_event_context (0.014)
HDBSCAN $m=20$	0.000 $\pm$ 0.003	0.012	0.024 $\pm$ 0.016	0.043	business_event_context (0.012)
<i>Qwen3-Embedding-8B base</i>					
K-means $K=8$	0.094 $\pm$ 0.097	0.271	0.259 $\pm$ 0.206	0.459	department (0.271)
K-means $K=16$	0.163 $\pm$ 0.159	0.420	0.345 $\pm$ 0.271	0.598	department (0.420)
K-means $K=24$	0.178 $\pm$ 0.169	0.420	0.372 $\pm$ 0.289	0.655	department (0.420)
K-means $K=32$	0.162 $\pm$ 0.149	0.369	0.385 $\pm$ 0.295	0.691	issue_type (0.369)
K-means $K=48$	0.160 $\pm$ 0.145	0.364	0.404 $\pm$ 0.300	0.742	ticket_category (0.364)
Ward $K=8$	0.124 $\pm$ 0.136	0.389	0.333 $\pm$ 0.270	0.616	intent_group (0.389)
Ward $K=16$	0.187 $\pm$ 0.184	0.465	0.389 $\pm$ 0.311	0.688	department (0.465)
Ward $K=24$	0.211 $\pm$ 0.200	0.496	0.416 $\pm$ 0.326	0.737	department (0.496)
Ward $K=32$	0.183 $\pm$ 0.169	0.439	0.414 $\pm$ 0.321	0.746	issue_type (0.439)
Ward $K=48$	0.180 $\pm$ 0.166	0.443	0.426 $\pm$ 0.321	0.795	ticket_category (0.443)
HDBSCAN $m=5$	0.011 $\pm$ 0.013	0.030	0.264 $\pm$ 0.184	0.473	issue_type (0.030)
HDBSCAN $m=10$	0.003 $\pm$ 0.007	0.015	0.151 $\pm$ 0.114	0.270	issue_type (0.015)
HDBSCAN $m=20$	-0.001 $\pm$ 0.003	0.006	0.030 $\pm$ 0.021	0.054	requester_role (0.006)
<i>Qwen3-Embedding-8B + department prefix</i>					
K-means $K=8$	0.082 $\pm$ 0.089	0.250	0.250 $\pm$ 0.198	0.454	department (0.250)
K-means $K=16$	0.119 $\pm$ 0.122	0.334	0.324 $\pm$ 0.255	0.578	department (0.334)
K-means $K=24$	0.123 $\pm$ 0.120	0.320	0.334 $\pm$ 0.257	0.576	department (0.320)
K-means $K=32$	0.131 $\pm$ 0.125	0.330	0.357 $\pm$ 0.271	0.612	department (0.330)
K-means $K=48$	0.126 $\pm$ 0.115	0.287	0.372 $\pm$ 0.272	0.643	department (0.287)
Ward $K=8$	0.076 $\pm$ 0.085	0.243	0.269 $\pm$ 0.215	0.498	department (0.243)
Ward $K=16$	0.128 $\pm$ 0.133	0.365	0.344 $\pm$ 0.271	0.625	department (0.365)
Ward $K=24$	0.143 $\pm$ 0.140	0.378	0.365 $\pm$ 0.283	0.646	department (0.378)
Ward $K=32$	0.154 $\pm$ 0.147	0.387	0.380 $\pm$ 0.290	0.660	department (0.387)
Ward $K=48$	0.139 $\pm$ 0.128	0.322	0.389 $\pm$ 0.287	0.663	department (0.322)
HDBSCAN $m=5$	0.014 $\pm$ 0.017	0.043	0.196 $\pm$ 0.141	0.345	intent_group (0.043)
HDBSCAN $m=10$	0.017 $\pm$ 0.021	0.057	0.142 $\pm$ 0.111	0.264	intent_group (0.057)
HDBSCAN $m=20$	0.022 $\pm$ 0.026	0.073	0.138 $\pm$ 0.111	0.262	department (0.073)
<i>Qwen3-Embedding-8B + issue_type prefix</i>					
K-means $K=8$	0.094 $\pm$ 0.099	0.266	0.258 $\pm$ 0.206	0.462	department (0.266)
K-means $K=16$	0.129 $\pm$ 0.126	0.322	0.315 $\pm$ 0.248	0.581	department (0.322)
K-means $K=24$	0.146 $\pm$ 0.140	0.370	0.354 $\pm$ 0.276	0.658	issue_type (0.370)
K-means $K=32$	0.163 $\pm$ 0.152	0.395	0.383 $\pm$ 0.295	0.711	issue_type (0.395)
K-means $K=48$	0.158 $\pm$ 0.147	0.369	0.399 $\pm$ 0.299	0.759	issue_type (0.369)
Ward $K=8$	0.105 $\pm$ 0.112	0.305	0.291 $\pm$ 0.233	0.515	intent_group (0.305)
Ward $K=16$	0.142 $\pm$ 0.139	0.357	0.344 $\pm$ 0.271	0.629	department (0.357)
Ward $K=24$	0.144 $\pm$ 0.136	0.336	0.363 $\pm$ 0.282	0.682	issue_type (0.336)
Ward $K=32$	0.156 $\pm$ 0.145	0.368	0.382 $\pm$ 0.294	0.724	ticket_category (0.368)
Ward $K=48$	0.160 $\pm$ 0.150	0.401	0.406 $\pm$ 0.306	0.778	ticket_category (0.401)
HDBSCAN $m=5$	0.001 $\pm$ 0.002	0.008	0.016 $\pm$ 0.007	0.024	urgency (0.008)
HDBSCAN $m=10$	0.008 $\pm$ 0.010	0.029	0.067 $\pm$ 0.055	0.171	intent_group (0.029)
HDBSCAN $m=20$	0.001 $\pm$ 0.003	0.006	0.052 $\pm$ 0.039	0.094	department (0.006)
<i>Qwen3-Embedding-8B + system_name prefix</i>					
K-means $K=8$	0.159 $\pm$ 0.178	0.509	0.350 $\pm$ 0.285	0.669	intent_group (0.509)
K-means $K=16$	0.187 $\pm$ 0.191	0.554	0.376 $\pm$ 0.300	0.699	department (0.554)
K-means $K=24$	0.165 $\pm$ 0.159	0.444	0.373 $\pm$ 0.291	0.667	department (0.444)
K-means $K=32$	0.171 $\pm$ 0.158	0.410	0.392 $\pm$ 0.299	0.678	department (0.410)
K-means $K=48$	0.148 $\pm$ 0.132	0.318	0.394 $\pm$ 0.289	0.675	department (0.318)
Ward $K=8$	0.122 $\pm$ 0.136	0.371	0.326 $\pm$ 0.266	0.627	department (0.371)
Ward $K=16$	0.179 $\pm$ 0.183	0.520	0.382 $\pm$ 0.305	0.711	department (0.520)
Ward $K=24$	0.162 $\pm$ 0.160	0.441	0.384 $\pm$ 0.301	0.696	department (0.441)
Ward $K=32$	0.152 $\pm$ 0.145	0.383	0.390 $\pm$ 0.300	0.690	department (0.383)
Ward $K=48$	0.161 $\pm$ 0.148	0.362	0.408 $\pm$ 0.304	0.697	department (0.362)
HDBSCAN $m=5$	0.021 $\pm$ 0.020	0.051	0.336 $\pm$ 0.226	0.574	issue_type (0.051)
HDBSCAN $m=10$	0.006 $\pm$ 0.009	0.021	0.182 $\pm$ 0.133	0.327	department (0.021)
HDBSCAN $m=20$	0.001 $\pm$ 0.004	0.008	0.057 $\pm$ 0.041	0.102	workflow_stage (0.008)

Table 9: Full one-shot grid: 65 pipeline configurations scored against all 24 reference partitions. Mean $\pm$ sd treats each reference in turn as the designated POV; max is the score on the pipeline’s favored reference. K = number of clusters; m = HDBSCAN minimum cluster size.

Metric	Definition / Formula
<i>Discovery Quality</i>	
POV Coverage	$\frac{ \hat{\mathcal{G}}_t \cap \mathcal{G} }{ \mathcal{G} }$
Supported Precision	$\frac{ \{g \in \hat{\mathcal{G}}_t : g \text{ is supported by } \mathcal{D}\} }{ \hat{\mathcal{G}}_t }$
Non-Redundancy	$\frac{ \text{Distinct}(\hat{\mathcal{G}}_t) }{ \hat{\mathcal{G}}_t }$
Discovery Utility AUC	$\frac{1}{T+1} \sum_{t=0}^T \frac{\sum_{g \in \hat{\mathcal{G}}_t \cap \mathcal{G}_{\mathcal{I}_t}} w_g}{\sum_{g \in \mathcal{G}_{\mathcal{I}_t}} w_g}$
<i>Objective Elicitation and Controllability</i>	
Target-Change Success	$\frac{1}{ \mathcal{Q}_{\text{change}} } \sum_{(i,j) \in \mathcal{Q}_{\text{change}}} \mathbf{1}[r_{P_t}(i,j) = r_{P_{\mathcal{I}_t}^*}(i,j)]$
Collateral Churn	$\frac{1}{ \mathcal{Q}_{\text{keep}} } \sum_{(i,j) \in \mathcal{Q}_{\text{keep}}} \mathbf{1}[r_{P_t}(i,j) \neq r_{P_{\mathcal{I}_t}^*}(i,j)]$
Lag Success	$\mathbf{1}[\exists t : \text{TCS}_t \geq \alpha \wedge \text{CC}_t \leq \beta]$
Clarification Quality	$\frac{\# \text{useful}}{\# \text{asked}}, \frac{\# \text{unnecessary}}{\# \text{asked}}, \frac{\# \text{missed}}{\# \text{needed}}$
<i>Final Clustering Quality</i>	
Conditioned ARI/NMI/VI	$\text{ARI}(P_T, P_{\mathcal{I}_T}^*), \text{NMI}(P_T, P_{\mathcal{I}_T}^*), \text{VI}(P_T, P_{\mathcal{I}_T}^*)$
Cluster Coherence	$\frac{1}{K_T} \sum_{k=1}^{K_T} \frac{s_k^{\text{coh}}}{10}$
Cluster Distinctiveness	$\frac{1}{\binom{K_T}{2}} \sum_{a < b} \frac{s_{ab}^{\text{dist}}}{10}$
Supported Cluster Rate	$\frac{\# \text{supported clusters in } P_T}{K_T}$
<i>Faithfulness</i>	
Abstention Rate	$\frac{1}{N_{\text{null}}} \sum_{i=1}^{N_{\text{null}}} \mathbf{1}[\text{rejected}_i \wedge \neg \text{agreed}_i]$
Rejection Rate	$\frac{1}{N_{\text{null}}} \sum_{i=1}^{N_{\text{null}}} \mathbf{1}[\text{rejected}_i]$
Agreement Rate	$\frac{1}{N_{\text{null}}} \sum_{i=1}^{N_{\text{null}}} \mathbf{1}[\text{agreed}_i]$
<i>Robustness</i>	
Invariance Success	$\mathbf{1}[\text{Sim}_P(P_a, P_b) \geq \gamma]$
Responsiveness Success	$\mathbf{1}[\text{Sim}_P(P_a, P_b) \leq \delta]$
<i>Efficiency and Interaction Value</i>	
Structural Utility AUC	$\frac{1}{T+1} \sum_{t=0}^T \max_{\tau \leq t} \text{ARI}(P_\tau, P^\dagger)$
Semantic Utility AUC	$\frac{1}{T+1} \sum_{t=0}^T \max_{\tau \leq t} \text{Sim}(\Psi_\tau, \mathcal{I}_\tau)$
Human Effort	$\# \text{oracle turns, avg. oracle message length, } \frac{\# \text{oracle words}}{\# \text{system words}}$
Interaction Cost	$C(\mathcal{N}_T) = \sum_{t=0}^T \text{cost}(m_t); \text{ input tokens, output tokens, total tokens}$

Table 10: Central metrics for conversational clustering.