

Gradient-Free Test-Time Adaptation for Few-Shot Online Clustering with Large Language Models

Avyav Kumar Singh

Department of Informatics
King's College London
avyav_kumar.singh@kcl.ac.uk

Helen Yannakoudakis

Department of Informatics
King's College London
helen.yannakoudakis@kcl.ac.uk

Abstract

We study few-shot online clustering (FSOC), where an unlabelled stream must be assigned to a dynamically growing cluster inventory using only a few labelled support examples for currently observed classes. We propose a two-component test-time gradient-free framework. First, we adapt Wasserstein Generative Flow Networks (wGFN) as an amortised LLM-based policy for in-distribution assignment and out-of-distribution (OOD) rejection. Unlike prior work that trains separate policies for each evaluation class set, we show that a single policy trained on a few classes generalises to novel support classes at test time without parameter updates, and further replace fixed rejection thresholds with a Dirichlet evidence model over candidate class assignments. Second, rejected OOD examples enter an oracle-mediated cluster-formation stage. Under shuffled stream orderings, this stage can create noisy provisional support sets that corrupt later assignments. To address this, we introduce a von Mises–Fisher Dirichlet Process Mixture Model warm start to provide oracle-independent cluster structure over unit-normalised text embeddings before oracle-mediated formation. This improves clustering quality, in some settings by $\sim 2\times$, even with small LLMs ($< 1\text{B}$ parameters). To our knowledge, this is the first FSOC framework to couple amortised LLM-based assignment with Bayesian non-parametric warm-starting, enabling growing cluster inventories without data augmentation, novel-class parameter updates or predefined cluster counts.

1 Introduction

The majority of textual data arrives as unlabelled streams, making online clustering a key step for organising data in applications such as real-time event monitoring (Miranda et al., 2018; Saravanakumar et al., 2021), intent discovery and dialogue systems (Zhang et al., 2021a; Song et al., 2023), and LLM pretraining (Sorscher et al., 2022;

Abbas et al., 2023; Vo et al., 2024). Unlike offline clustering, which assumes access to the full dataset, online clustering assigns examples sequentially and must adapt as the data distribution changes (Garcia et al., 2025). This setting is challenging because the number of clusters can grow over time, incoming examples may induce cluster fragmentation or collapse (Mahon and Lukasiewicz, 2024), and new clusters may drift semantically from existing ones (Losing et al., 2018; Garcia et al., 2025). These challenges are amplified in few-shot online clustering (FSOC), where only a few labelled support examples are available for currently observed classes.

We study FSOC as a support-conditioned form of online clustering. The stream is unlabelled and the final number of clusters is unknown, but the model receives a few labelled support examples for currently observed or confirmed classes. At each step, it must either assign an example to the current support inventory or reject it as out-of-distribution (OOD) so that a new cluster can be formed. Thus, FSOC combines few-shot supervision with open-set assignment and open-world cluster discovery. We propose a two-component test-time gradient-free framework. First, an amortised LLM-based assignment policy handles in-distribution assignment and OOD rejection. Second, OOD-routed examples are passed to an LLM-oracle-mediated cluster-formation stage, which groups them into provisional support sets for newly emerging classes.

Our contributions First, we adapt Wasserstein Generative Flow Networks (wGFN) from few-shot open-set classification to FSOC by turning wGFN into a reusable LLM-based assignment-and-rejection policy for a changing support inventory. Unlike prior work, a single policy trained on only a few classes can assign or reject examples from novel support classes introduced only through few-shot support examples at test time, without parameter updates. We further replace fixed Z-score re-

jection with a Dirichlet evidence model over candidate class assignments. Second, we define the full FSOC pipeline by using wGFN for in-distribution assignment and invoking an instruction-tuned LLM oracle only for OOD-routed examples and provisional cluster formation. We evaluate our approach across NER, intent detection, and fine-grained emotion clustering under class-incremental and shuffled stream orderings, and show that we outperform multi-step CoT reasoning across all clustering quality metrics. Third, we introduce a von Mises–Fisher Dirichlet Process Mixture Model (vMF-DPMM) warm start over unit-normalised LLM embeddings. This provides oracle-independent cluster structure before oracle-mediated formation and mitigates noisy provisional support sets that can corrupt later assignments under shuffled streams.

2 Related work

FSOC sits at the intersection of few-shot assignment, OOD rejection and online cluster formation. Therefore, we review work on closed-set few-shot classification, LLM-based clustering, Bayesian non-parametric clustering and GFNs for LLMs.

Few-shot assignment and clustering Few-shot NLP includes episodic meta-learning (Bao et al., 2020; Zhang et al., 2021b, 2022a; Singh et al., 2024), in-context learning (Brown et al., 2020; Min et al., 2022), instruction tuning (Wei et al., 2022a; Chung et al., 2024) and parameter-efficient adaptation (Lester et al., 2021; Li and Liang, 2021). The most relevant line for FSOC is support-conditioned few-shot classification, where a model assigns each query to one of the displayed support classes in an N -way K -shot episode. These methods are typically closed-set, whereas FSOC must also reject queries that do not belong to the current support inventory and incorporate newly emerging classes online. Representation-based few-shot clustering methods based on Sentence-BERT (Reimers and Gurevych, 2019), contrastive learning (Zhang et al., 2021b, 2022b; Mou et al., 2022), SetFit (Tunstall et al., 2022), and prototypical objectives (Zhang et al., 2021a; Chen et al., 2022) provide strong assignment signals, but usually require large episodic training sets, rely on domain-sensitive augmentation, or operate in batch mode.

LLM-based clustering and augmentation Prior work uses LLMs to define cluster structure through natural-language descriptions, constraints, sum-

maries, or generated data. ClusterLLM (Zhang et al., 2023), TopicGPT (Pham et al., 2024), TnT-LLM (Wan et al., 2024), IDAS (De Raedt et al., 2023), GoalEx (Wang et al., 2023), and related methods Diaz-Rodriguez (2026) improve interpretability or refine embedding spaces, but typically assume a fixed cluster count or lack a mechanism for incrementally accommodating new categories as data streams in. Other methods use LLMs as data generators to expand small labelled sets (Yoo et al., 2021; Ye et al., 2022; Dai et al., 2025; Sahu et al., 2023; Viswanathan et al., 2024), but this adds compute overhead and a hard-to-control dependence on the LLM’s pretraining distribution. We instead study few-shot online clustering without augmenting the support set.

Bayesian non-parametrics and GFNs Bayesian non-parametric (BNP) models (Battamaghelich et al., 2016; Chen et al., 2021), such as Dirichlet process mixture models (Ferguson, 1973; Antoniak, 1974), hierarchical Dirichlet processes (Teh et al., 2006b), and DP-means (Kulis and Jordan, 2012), provide a principled way to infer the number of clusters from data. Their core limitation in our setting is data hunger: BNP posteriors often require enough observations per component for stable posterior concentration. We use BNP clustering differently, as a warm start that provides oracle-independent structure before LLM-mediated cluster formation. Generative Flow Networks (GFNs) (Bengio et al., 2023) train stochastic policies to sample structured objects proportionally to a reward, and have been shown to amortise intractable posterior inference in LLMs. Singh and Yannakoudakis (2025) use Wasserstein-GFNs (wGFN) for few-shot open-set classification with Gaussian Process (GP)-based in-distribution (ID) / OOD decisions, but do not study whether one policy can be reused as the support inventory changes online. We instead test this property and use it inside an FSOC pipeline.

3 Evaluating test-time generalisation of wGFN to novel support classes

The assignment component of our FSOC framework relies on a key property: a wGFN policy trained on a small fixed class set should remain usable when the support inventory changes at test time. We therefore first isolate this question before evaluating the full online clustering pipeline. Specifically, we test whether a single wGFN policy

trained on a small fixed set of classes can assign and reject examples from novel support classes introduced only through few-shot support examples.

Core principle of wGFN Given a few-shot support set X of K examples per class Y , Singh and Yannakoudakis (2025) sample *latent sequences* $Z \sim p_{\text{LLM}}(Z | X)$ from a frozen (base) LLM which serve as class descriptors, and train a policy q_{GFN} via the subtrajectory balance objective (Madan et al., 2023) given by:

$$\mathcal{L}_{\text{SubTB}}(Z; \theta) = \sum_{1 \leq j \leq n} \sum_{0 \leq i < j} \left[\log \frac{R(z_{1:i} \top) \prod_{k=i+1}^j q(z_k | z_{1:k-1}) q(\top | z_{1:j})}{R(z_{1:j} \top) q(\top | z_{1:i})} \right]^2 \quad (1)$$

to map each sampled latent Z to its corresponding label Y with a high reward $R(XZY)$ (effectively, $R(XZY) \propto q_{\text{GFN}}(XZY)$, thus training a policy model q_{GFN} to maximise the reward assigned to the correct latent $\rightarrow$ label mapping). At inference, if a test example’s latents come from a *known* class distribution, q_{GFN} assigns them high probability relative to the base LLM and a confident in-distribution (ID) assignment. Conversely, if the latents come from an *unknown* distribution, they cannot be confidently mapped to any support label, yielding an OOD prediction. We present a detailed overview in Appendix A.

Extending wGFN to novel classes Let C_{train} denote the classes used to train q_{GFN} , C_{support} the classes represented in the few-shot support set at test time, and C_{test} the full set of classes evaluated at test time. For a given support inventory, examples from C_{support} require ID assignment, while examples from $C_{\text{OOD}} = C_{\text{test}} \setminus C_{\text{support}}$ require OOD rejection. Singh and Yannakoudakis (2025) explicitly set $C_{\text{train}} = C_{\text{support}}$, thus evaluating q_{GFN} only on the same set of classes used for training q_{GFN} . We instead evaluate the harder setting where $C_{\text{support}} \setminus C_{\text{train}} \neq \emptyset$. Therefore, some support classes are introduced only through few-shot demonstrations at test time and have never received explicit policy training. The dual objective, is then to assign test examples to one of the C_{support} classes or reject them as OOD using the amortised policy model q_{GFN} .

Experimental setup We conduct experiments on MultiNERD (Tedeschi and Navigli, 2022) using Llama-3.2 1B, 3B & Llama-3.1 8B (MetaAI, 2024)

model backbones. Each backbone is trained with the wGFN objective on C_{train} where $|C_{\text{train}}| = 3$ and evaluated on a fixed test set C_{test} with $|C_{\text{test}}| = 10$. We vary the support inventory from $|C_{\text{support}}| = 3$ to 10 and evaluate two settings:

$\rightarrow$ **Incremental:** We begin with few-shot demonstrations for classes in C_{train} , so initially $C_{\text{support}} = C_{\text{train}}$. We then add one novel support class at a time, update $|C_{\text{support}}| \leftarrow |C_{\text{support}}| + 1$, sample few-shot demonstrations for all classes in C_{support} and record ID and OOD performance after each update. This evaluates how wGFN scales as the support inventory grows beyond the classes used for policy training.

$\rightarrow$ **Disjoint:** We construct C_{support} by randomly sampling classes from C_{test} so the support inventory has no overlap with C_{train} . This provides a stronger test of support-conditioned transfer, since the policy must assign examples to support classes that were not used during training.

In both settings, q_{GFN} is trained only on the three classes in C_{train} , while C_{support} may include classes from $C_{\text{test}} \setminus C_{\text{train}}$ that are introduced only through few-shot support examples. We report the macro F1 scores of ID classification and OOD rejection as $|C_{\text{support}}|$ increases from 3 – 10. We present our results in Figure 1 and our key finding:

Key Finding: wGFN extends its learned ID assignment and OOD rejection capabilities to classes it hasn’t been explicitly trained on.

Empirically, given as few as 4 demonstrations per support class, wGFN performs ID assignment and OOD rejection even when $C_{\text{support}} \setminus C_{\text{train}} \neq \emptyset$. No parameter updates are performed for novel support classes. Figure 1 shows graceful degradation as $|C_{\text{support}}|$ grows and the assignment problem becomes harder where the key result is functional transfer, not invariance to the number of support classes (see Appendix A for experimental methodology and complete results)

Discussion Llama-3.1 8B maintains robust ID and OOD performance across both settings, Llama-3.2 3B follows the same qualitative trend with a moderate performance reduction, and Llama-3.2 1B exhibits the steepest degradation as $|C_{\text{support}}|$ grows. Across all models and settings, wGFN generalises gradient-free to novel support classes via few-shot demonstrations alone, with larger models sustaining this capability over wider support

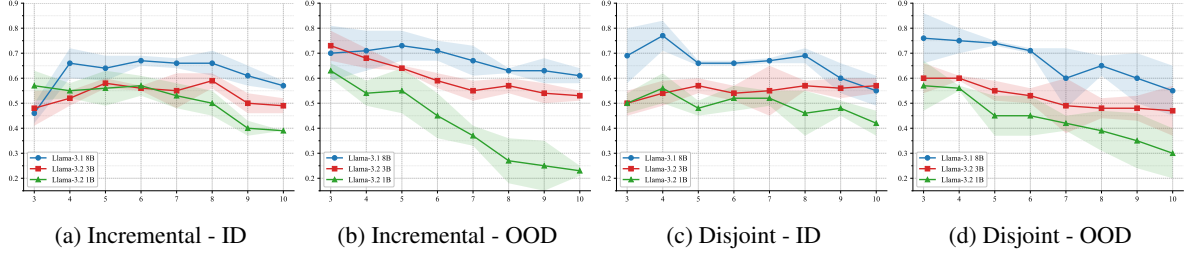


Figure 1: F1 scores of in-distribution assignment and OOD rejection as we increase the number of support classes in the incremental and disjoint settings.

inventories. This suggests that the policy exploits semantic structure in the support examples rather than memorising a fixed label set, consistent with reported implicit meta-learning properties of LLMs (Dai et al., 2023; Von Oswald et al., 2023).

4 Few-shot online clustering with wGFN

Motivation We now use wGFN’s property in Section 3 as the assignment component of FSOC. As shown in Figure 2, the pipeline has two components: an amortised wGFN policy for ID assignment and OOD rejection, and an oracle-mediated cluster-formation module for OOD-routed examples. This decomposition lets q_{GFN} handle the high-volume assignment path, while the oracle is invoked only when an example is rejected as OOD and may belong to a newly emerging class.

Probabilistic posterior inference Singh and Yannakoudakis (2025) detect OOD examples by Z-score normalising class-wise KL divergences $d_k = \text{KL}(\mathcal{N}_0^k \parallel \mathcal{N}_1)$ between the test logit distribution and a per-class GP, assigning to class c only when $Z_c^s \geq z$ (typically $z = 3.5$). While suitable for open-set classification, this fixed, scale-sensitive heuristic is brittle in online clustering, where the support inventory changes over time. We replace it with a Dirichlet evidence model over candidate assignments. Since the class-specific GP models an OOD signal, a larger divergence d_k provides stronger evidence that x is incompatible with the OOD distribution for class k , and therefore more likely to be in-distribution for that class.

For K classes, we initialise a symmetric Dirichlet concentration vector α with $\alpha_k = \alpha_0$ for all k , giving a uniform prior over candidate class-assignment probabilities $\pi = (\pi_1, \dots, \pi_K)$. We then convert the class-wise KL divergences into soft fractional pseudo-counts via temperature nor-

malisation:

$$\tilde{\alpha}_k(x) = \frac{\exp(d_k/\tau)}{\sum_{j=1}^K \exp(d_j/\tau)}, \quad k = 1, \dots, K \quad (2)$$

where $\tau > 0$ controls sharpness. In practice, candidate classes may be scored in partitions for context-window tractability, but the soft pseudo-counts are normalised globally over the current candidate set before updating α . These pseudo-counts update the local Dirichlet concentrations for the current inference decision: $\alpha_k \leftarrow \alpha_k + \tilde{\alpha}_k(x)$. The resulting Dirichlet distribution is

$$p(\pi \mid \alpha) = \frac{\Gamma(\sum_{k=1}^K \alpha_k)}{\prod_{k=1}^K \Gamma(\alpha_k)} \prod_{k=1}^K \pi_k^{\alpha_k - 1} \quad (3)$$

Unlike a fixed Z-score threshold, this parameterisation aggregates KL-based evidence into a prior-smoothed posterior over the current candidate assignments. The posterior mean

$$\hat{\pi}_k = \mathbb{E}[\pi_k \mid \alpha] = \frac{\alpha_k}{\sum_{j=1}^K \alpha_j} \quad (4)$$

gives the assignment probability for class C_k . We assign x to $\arg \max_k \hat{\pi}_k$ if and only if $\max_k \hat{\pi}_k \geq \delta$ for a hyperparameter δ ; otherwise, x is rejected as OOD and routed to the oracle buffer.

Usage of oracle Creating high-quality few-shot support demonstrations for newly emerging classes is non-trivial in the online setting. We therefore use a frozen instruction-tuned LLM as a formation oracle to estimate pairwise semantic similarity between incoming test points. When q_{GFN} rejects an incoming example as OOD, the oracle compares it with provisional clusters in the buffer and decides whether it belongs to one of them. Once a provisional cluster accumulates K sufficiently similar examples to form a support set, it is confirmed and added to the support inventory. This design avoids

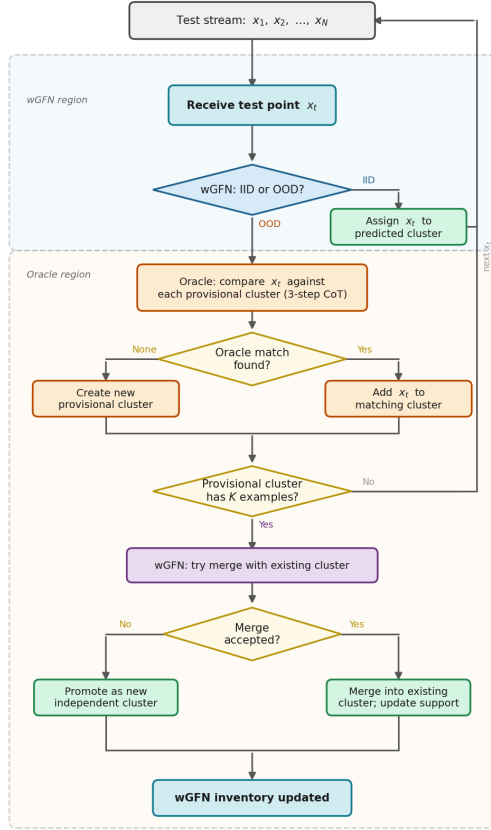


Figure 2: FSOC pipeline: wGFN for ID assignment and OOD rejection; OOD-routed examples passed to oracle-mediated cluster-formation module which forms provisional support sets for newly emerging classes.

exhaustive oracle assignment: q_{GFN} handles routine ID assignment, while the oracle is invoked only for OOD-routed examples and cluster-formation decisions. We provide the oracle model, prompt format, and decoding parameters in Appendix B.

Modes of test distribution To control for noise in the few-shot support set, we evaluate under two test distributions:

→ **Class Incremental:** All test points from cluster c_i arrive contiguously before those of c_{i+1} , yielding homogeneous provisional clusters before confirmation (Lee et al., 2020; Fan et al., 2024). This setting makes cluster formation easier because early OOD-routed examples tend to come from the same class, thus drawing cluster boundaries in the test distribution, which makes it easier for the oracle to distinguish clusters.

→ **Shuffled:** The incoming stream is uniformly randomised across all clusters simultaneously, producing potentially noisier provisional support sets and a harder evaluation setting. These settings stress-test both parts of the pipeline: the assign-



Figure 3: wGFN outperforms CoT consistently across all datasets, invariance conditions, models (3B, 8B) and stream orderings (CI = Class Incremental, S = Shuffled). Cells: signed improvement wGFN – CoT (Δpp); green indicates wGFN is better, red/orange indicate CoT is.

ment module, instantiated by either CoT (see Section 5) or wGFN and the oracle-mediated formation of provisional support sets.

Task datasets We evaluate on three benchmarks spanning named entity recognition (MultiNERD (Tedeschi and Navigli, 2022)), fine-grained emotion classification (GoEmotion (Demszky et al., 2020)), and intent detection (CLINC-150 (Larson et al., 2019)). Each dataset has 10 gold clusters with varying semantic overlap (see Appendix, Figure 9). Further details are in Appendix B.

5 Comparing wGFN and CoT for FSOC

Axes of comparison Online clustering is sensitive to arrival order: early errors can cascade forward and are difficult to correct in a single-pass setting (Lee et al., 2020; Tian et al., 2024). The few-shot support set adds a further source of variability, as the choice of support classes and examples can shift prompt-conditioned assignment performance (Vinyals et al., 2016; Snell et al., 2017). To

study the robustness of wGFN against multi-step Chain-of-Thought (CoT) reasoning (Llama-3.2 3B & Llama-3.1 8B), we evaluate it under 3 controlled axes with 3 runs per axis: (a) *Support sampling*: classes are fixed but few-shot support examples are re-sampled; (b) *Data order*: the test dataset is fixed but arrival order varies; (c) *Initialisation*: the initial support classes are varied across three starting cluster sets. We report the mean and std. across runs, models and stream settings.

Method and baseline CoT is a natural baseline because it uses the same few-shot support examples and the same autoregressive backbone to perform assignment directly through prompting: it thus tests whether wGFN training provides benefits beyond prompted reasoning alone. We give CoT an explicit OOD hint, making it a strong prompted baseline. Unlike wGFN which learns OOD rejection as part of the policy, CoT is instructed that a query may belong to none of the presented clusters (see Figure 8). We present our high-level pipeline in Figure 2. We use Qwen-3-8B (Yang et al., 2025) as the formation oracle and train q_{GFN} on Llama-3.1 8B and Llama-3.2 3B backbones. The CoT baseline replaces wGFN decisions in the same pipeline (ID/OOD assignment and merge decisions) with Llama-3.1 8B and Llama-3.2 3B instruction-tuned multi-step LLM reasoning. We report clustering accuracy after Hungarian label matching, which measures recovery of the true partition up to arbitrary cluster IDs. We also report ARI, AMI, and NMI: ARI and AMI evaluate partition quality with chance correction, while NMI measures shared information between predicted and gold partitions. Further details are in Appendix C.

Findings Figure 3 summarises the signed improvement $\Delta pp = \text{wGFN} - \text{CoT}$ across all three axes; full results are in Appendix C, Table 11. In the class-incremental setting, wGFN outperforms CoT by a substantial margin across datasets, model sizes, and invariance axes, with accuracy gains of roughly +15 to +30 percentage points and similar advantages in ARI, AMI, and NMI. In the shuffled setting, the margin narrows but wGFN remains stronger in most comparisons, with near-parity only for the 3B model on CLINC-150. This suggests that shuffled streams stress the cluster-formation stage rather than only the assignment policy: when examples from different classes arrive interleaved, the oracle can form noisy provisional support sets, and both wGFN and CoT inherit this corrupted sup-

port. This effect is compounded by task complexity: NERD examples are more tightly aligned to their cluster means, whereas Emotion and CLINC-150 exhibit greater semantic overlap and higher within-cluster variance (Figure 9).

6 Improving the oracle with vMF-DPMM

Motivation Section 5 shows that the main failure mode in shuffled streams is not only assignment, but support formation. When examples from many classes arrive interleaved, oracle-mediated cluster formation can create noisy provisional support sets: both CoT and wGFN inherit this corrupted support. Scaling to a larger or multi-agent instruction-tuned oracle would increase cost without removing this dependence on early pairwise decisions. We therefore add a Bayesian non-parametric warm start before oracle-mediated formation. Specifically, we use a Dirichlet Process Mixture Model (DPMM) over unit-normalised embeddings to provide oracle-independent cluster structure before OOD-routed examples are grouped into support sets. This component can be used either as a standalone clustering method or as a warm-start initialisation mechanism for the full online pipeline.

Using vMF to model clusters Embedding LLMs (Yang et al., 2025) produce unit-normalised text representations, placing examples on the unit hypersphere $\mathcal{S}^{d-1}$. The von Mises–Fisher (vMF) distribution is a natural density for this geometry: $\text{vMF}(\mathbf{x} \mid \boldsymbol{\mu}, \kappa) \propto \exp(\kappa \boldsymbol{\mu}^\top \mathbf{x})$, parameterised by a mean direction $\boldsymbol{\mu} \in \mathcal{S}^{d-1}$ and a concentration $\kappa > 0$ that controls the spread of mass around $\boldsymbol{\mu}$ (higher κ corresponds to a tighter cluster).

Modelling mixture proportion The Chinese Restaurant Process (CRP) given as:

$$p(\mathbf{z}_{1:n} \mid \alpha) = \frac{\Gamma(\alpha) \alpha^K}{\Gamma(\alpha + n)} \prod_{k=1}^K \Gamma(|C_k|)$$

acts as a prior over partitions: each arriving observation joins an existing cluster with probability proportional to its current size or opens a new one with probability proportional to α . The model thus infers the number of clusters directly from data.

Sequential variational approximation We adopt the widely-used (Lee et al., 2020; Jeong et al., 2023) sequential variational approximation framework of Lin (2013) to model the online clustering process and provide an extension of this

generative model (vMF likelihood) on the unit sphere $\mathcal{S}^{d-1}$:

$$\begin{aligned} \mathbf{z}_{1:n} &\sim \text{CRP}(\alpha) \\ \phi_k &\sim \text{vMF}(\boldsymbol{\mu}_0, \kappa_0) \quad \forall k \\ \mathbf{x}_i &\sim \text{vMF}(\phi_{z_i}, \kappa) \quad \forall i \end{aligned}$$

where $\alpha > 0$ is the DP concentration, $\boldsymbol{\mu}_0 \in \mathcal{S}^{d-1}$ and $\kappa_0 > 0$ are the prior mean and concentration, and $\kappa > 0$ is the likelihood concentration. The variational distribution factorises as $q(\mathbf{z}_{1:n}) = \prod_i \rho_i(z_i)$, where $\rho_i(k)$ is the soft responsibility of assigning observation i to cluster k . Each cluster posterior ν_k represents the cluster parameters for cluster k which are independently updated.

Update rules Using Lin (2013)’s framework and plugging in the closed-form vMF conjugate structure (see Appendix D), the collapsed responsibility update for an arriving observation $\mathbf{x}_i$ is:

$$\rho_i(k) \propto \begin{cases} n_k^{(i-1)} \frac{C_d(\|\mathbf{r}_k^{(i-1)}\|)}{C_d(\|\mathbf{r}_k^{(i-1)}\| + \kappa \mathbf{x}_i)} & k \leq K \\ \frac{\alpha C_d(\kappa_0)}{C_d(\|\kappa_0 \boldsymbol{\mu}_0 + \kappa \mathbf{x}_i\|)} & k = K+1 \end{cases}$$

where $C_d(\cdot)$ is the vMF normalising constant. The component posterior ν_k is updated implicitly by accumulating $\mathbf{r}_k$:

$$\mathbf{r}_k \leftarrow \mathbf{r}_k + \rho_i(k) \kappa \mathbf{x}_i \quad (5)$$

and $\nu_k = \text{vMF}(\phi_k \mid \mathbf{r}_k / \|\mathbf{r}_k\|, \|\mathbf{r}_k\|)$. The resultant $\mathbf{r}_k$ is the sole sufficient statistic of ν_k : its direction gives the posterior mean direction of cluster k , and its magnitude encodes accumulated evidence (growing as more observations are confidently assigned to the cluster). Updating $\mathbf{r}_k$ after each change in ρ_i automatically updates ν_k , so ν_k never needs to be constructed explicitly.

Iterative refinement After a single-pass, we refine assignments by iterating over all points. As each $\mathbf{x}_i$ has already contributed to $\mathbf{r}_k$, direct rescaling would double-count its influence. We instead subtract $\mathbf{x}_i$ ’s current soft contribution before rescaling (producing leave-one-out sufficient statistics $\mathbf{r}_{-i,k} = \mathbf{r}_k - \rho_i(k) \kappa \mathbf{x}_i$ and $n_{-i,k} = n_k - \rho_i(k)$) and then restore the contribution with the updated responsibility. This *subtract-and-restore* procedure is structurally identical to the E-step of Expectation Maximisation: holding all other assignments fixed, we recompute the posterior over z_i given

the leave-one-out statistics until convergence. All derivations are provided in Appendix D. We refer to the one-pass vMF-DPMM as DPMM and the subtract-and-rescore refined model as DPMM*.

7 Results and Discussion

Experimental setup We evaluate on NERD, CLINC-150, and Emotion using Qwen-3-0.6B and Qwen-3-4B (Yang et al., 2025) as autoregressive backbones for CoT and wGFN and Qwen-3-0.6B-Embedding and Qwen-3-4B-Embedding for DPMM, DPMM*, and DBSCAN. To ensure that we use only a single model per task, we use the same model as an oracle for wGFN and CoT, thus keeping comparison fair. This means that oracle performance in the shuffled setting for autoregressive clustering methods is constrained as smaller language models, particularly Qwen-3-0.6B, are not adept at multi-step CoT reasoning tasks.¹ DBSCAN serves as a strong encoder baseline against which DPMM/DPMM* gains are measured: it tests whether standard density-based clustering over LLM embeddings is sufficient. We run our experiments across 3 seeds and report the mean and standard deviation. Table 1 evaluates the full FSOC setting, comparing all methods using the previously defined metrics.

Results Table 1 shows a complementary pattern. In class-incremental streams, where provisional support sets are relatively clean, wGFN is strongest among autoregressive assignment methods and often leads on clustering quality metrics. In shuffled streams, where oracle-mediated support formation is noisier, DPMM* is consistently stronger, showing that oracle-independent structure is more robust when class examples are interleaved. This motivates using DPMM* not as a replacement for wGFN in all settings, but as a warm start when support formation is unreliable.

Analysis of wGFN Across datasets, orderings, and model sizes, wGFN outperforms multi-step CoT while using a single forward assignment path and smaller token budget. The gains are especially important at small model scales, where CoT often collapses but wGFN remains usable. This supports the role of wGFN as the assignment component of FSOC. However, the shuffled setting shows that assignment is not the only bottleneck; oracle quality

¹Qwen-3-0.6B-Embedding achieves up to 99% of Qwen-3-4B-Embedding’s performance on the MTEB benchmark.

Set.	Task	Model	Embedding-based			Autoregressive	
			DBSCAN	DPMM	DPMM*	CoT	wGFN
Class Incremental	NERD	0.6B	45.29	53.53 _{2.38}	56.05 _{0.32}	28.63 _{1.05}	48.21 _{1.59}
		4B	54.93	59.19 _{1.12}	63.12 _{2.38}	60.76 _{2.48}	64.17 _{2.56}
	CLINC-150	0.6B	37.33	57.33 _{3.03}	60.56 _{0.79}	29.00 _{1.41}	31.22 _{1.91}
		4B	34.89	51.11 _{1.89}	56.78 _{2.04}	46.15 _{2.83}	54.33 _{1.73}
	Emotion	0.6B	33.64	29.66 _{0.16}	37.39 _{3.70}	23.07 _{0.16}	35.91 _{0.96}
		4B	32.95	35.00 _{2.25}	37.07 _{1.90}	32.12 _{3.76}	43.41 _{0.04}
Shuffled	NERD	0.6B	55.61	49.55 _{0.95}	57.17 _{0.00}	18.95 _{1.74}	22.65 _{2.85}
		4B	37.22	61.60 _{3.96}	65.02 _{3.49}	53.24 _{3.01}	53.73 _{1.79}
	CLINC-150	0.6B	34.44	45.22 _{1.73}	50.56 _{1.41}	22.00 _{1.57}	21.11 _{0.88}
		4B	34.89	44.44 _{0.31}	48.89 _{0.00}	43.22 _{1.10}	24.78 _{1.41}
	Emotion	0.6B	31.15	30.80 _{0.17}	35.13 _{0.99}	19.91 _{0.00}	20.32 _{1.73}
		4B	33.96	28.81 _{0.99}	33.49 _{1.99}	35.83 _{3.97}	29.47 _{1.03}

(a) Accuracy

Set.	Task	Model	Embedding-based			Autoregressive	
			DBSCAN	DPMM	DPMM*	CoT	wGFN
Class Incremental	NERD	0.6B	44.39	53.80 _{3.05}	57.74 _{0.12}	23.28 _{0.76}	59.75 _{2.02}
		4B	53.83	54.30 _{0.44}	58.91 _{0.37}	59.22 _{3.03}	67.58 _{1.41}
	CLINC-150	0.6B	35.11	55.05 _{0.61}	58.29 _{1.85}	15.77 _{0.56}	42.71 _{1.59}
		4B	38.27	51.77 _{1.06}	58.77 _{1.06}	48.45 _{0.15}	58.98 _{0.41}
	Emotion	0.6B	22.10	23.90 _{0.83}	26.37 _{1.87}	11.95 _{1.75}	41.06 _{1.87}
		4B	25.15	25.07 _{0.82}	26.45 _{3.99}	24.16 _{2.64}	42.48 _{0.41}
Shuffled	NERD	0.6B	54.33	56.39 _{2.86}	58.73 _{2.41}	3.93 _{0.88}	12.93 _{0.88}
		4B	42.64	60.95 _{0.61}	62.48 _{1.81}	54.80 _{0.68}	57.08 _{3.27}
	CLINC-150	0.6B	35.38	47.27 _{1.12}	55.95 _{3.09}	7.03 _{2.76}	7.74 _{1.22}
		4B	43.87	49.63 _{3.03}	55.22 _{1.00}	37.62 _{1.96}	19.32 _{1.41}
	Emotion	0.6B	29.25	19.94 _{1.12}	24.32 _{0.97}	5.52 _{0.99}	8.75 _{1.36}
		4B	24.55	18.42 _{0.14}	23.04 _{0.18}	22.62 _{2.89}	23.78 _{1.82}

(c) AMI

Set.	Task	Model	Embedding-based			Autoregressive	
			DBSCAN	DPMM	DPMM*	CoT	wGFN
Class Incremental	NERD	0.6B	27.77	36.96 _{3.10}	41.05 _{0.03}	10.85 _{0.70}	36.03 _{1.85}
		4B	33.98	38.38 _{1.23}	43.81 _{1.16}	45.02 _{7.24}	47.00 _{0.06}
	CLINC-150	0.6B	21.18	36.01 _{3.35}	40.13 _{0.54}	8.53 _{0.02}	20.77 _{1.60}
		4B	21.34	32.28 _{0.68}	40.68 _{1.44}	31.81 _{0.55}	42.24 _{0.88}
	Emotion	0.6B	9.73	11.00 _{1.04}	17.45 _{2.35}	3.35 _{1.61}	19.12 _{0.17}
		4B	11.53	13.02 _{0.00}	15.55 _{1.93}	14.78 _{3.62}	24.02 _{1.11}
Shuffled	NERD	0.6B	37.47	35.26 _{5.40}	38.19 _{6.30}	2.15 _{0.04}	8.85 _{0.05}
		4B	18.44	42.47 _{0.69}	48.32 _{2.18}	37.75 _{2.74}	36.74 _{2.16}
	CLINC-150	0.6B	16.76	27.50 _{1.84}	36.54 _{3.18}	4.29 _{1.58}	5.03 _{0.64}
		4B	22.37	24.07 _{0.69}	33.18 _{2.55}	23.45 _{0.37}	13.00 _{2.12}
	Emotion	0.6B	10.01	15.01 _{1.59}	16.62 _{1.04}	4.07 _{1.31}	4.19 _{1.41}
		4B	11.95	12.31 _{1.24}	15.51 _{0.37}	15.17 _{2.10}	11.66 _{1.39}

(b) ARI

Set.	Task	Model	Embedding-based			Autoregressive	
			DBSCAN	DPMM	DPMM*	CoT	wGFN
Class Incremental	NERD	0.6B	48.61	56.72 _{2.80}	59.54 _{0.12}	28.39 _{0.57}	63.88 _{2.36}
		4B	56.89	56.36 _{0.67}	60.56 _{0.09}	62.73 _{2.90}	70.62 _{1.88}
	CLINC-150	0.6B	40.19	57.69 _{0.39}	61.30 _{1.87}	21.26 _{0.42}	49.74 _{1.47}
		4B	44.17	54.88 _{0.68}	61.42 _{1.19}	52.74 _{1.08}	62.15 _{1.31}
	Emotion	0.6B	24.36	31.39 _{2.12}	30.65 _{1.56}	16.84 _{1.72}	45.84 _{2.35}
		4B	27.70	29.24 _{0.16}	30.16 _{2.79}	31.76 _{2.00}	46.80 _{0.36}
Shuffled	NERD	0.6B	58.30	60.81 _{2.89}	61.70 _{2.41}	10.34 _{0.20}	18.59 _{0.44}
		4B	46.78	63.33 _{0.68}	65.39 _{1.57}	58.82 _{1.77}	59.22 _{2.38}
	CLINC-150	0.6B	40.78	50.71 _{0.72}	57.77 _{2.78}	13.15 _{1.82}	15.26 _{2.19}
		4B	50.23	52.30 _{2.86}	57.96 _{0.95}	41.18 _{2.02}	25.92 _{1.83}
	Emotion	0.6B	31.13	25.84 _{1.34}	28.80 _{0.87}	10.14 _{0.84}	11.77 _{0.94}
		4B	26.75	23.22 _{0.15}	26.76 _{0.15}	28.06 _{2.48}	28.44 _{1.27}

(d) NMI

Table 1: Clustering performance (Accuracy, ARI, AMI, NMI) across tasks and stream orderings. All models belong to the Qwen-3 family. Subscripts denote standard deviation. **green** indicates the best result per row. Columns shaded light blue (DPMM, DPMM*, wGFN) denote our proposed methods; DBSCAN and CoT are baselines.

is also a limiting factor: noisy provisional support sets suppress all autoregressive methods, motivating the vMF-DPMM warm start.

DPMM in shuffled setting DPMM’s advantage in the shuffled setting stems from order-robust evidence accumulation: as each $\mathbf{x}_i$ arrives, its soft responsibility $\rho_i(k)$ updates the sufficient statistic $\mathbf{r}_k$ (Equation 5), refining cluster structure independently of provisional support quality. Autoregressive methods, by contrast, depend on early oracle-mediated support formation; when shuffled streams create noisy provisional support sets, both CoT and wGFN inherit this noise. DPMM is therefore useful not only as a standalone method, but also as a low-cost warm-start that provides oracle-independent cluster structure before autoregressive assignment.

Warm-starting wGFN We next test whether improving cluster structure is more effective than scaling the formation oracle. We run wGFN with a 4B oracle after a DPMM warm start (~ 5 s processing time) on the shuffled evaluation stream (wGFN*) and compare against larger-oracle variants (Qwen-3-8B and Qwen-3-30B-A3B) using the same 4B wGFN backbone (Table 4). DPMM warm-starting

substantially improves wGFN, with performance gains reaching $\sim 2\times$ at times. wGFN* can match or even exceed a $10\times$ larger oracle, suggesting that improving cluster structure can be more effective than simply scaling the oracle.

wGFN vs pairwise oracle-only clustering Despite its high cost, we broadly expect oracle-only clustering to produce denser individual clusters than CoT: whereas CoT processes all candidate classes within a single in-context query, pairwise comparison resolves each cluster decision in isolation and therefore tends to produce a larger number of finer-grained clusters. This is reflected in Table 2: accuracy and ARI, which penalise over-segmentation, tend to be lower for oracle-only clustering than for CoT, while AMI and NMI, which reward intra-cluster homogeneity, tend to be higher (the pattern is clearest on NERD and Emotion). Across all datasets, wGFN consistently outperforms both baselines on every metric, demonstrating that a trained policy paired with DPMM-based merging can achieve both accurate cluster granularity and high intra-cluster quality while avoiding the $O(NK)$ oracle overhead.

	Method	Accuracy	ARI	AMI	NMI
NERD	CoT	60.76 _{2.48}	45.02 _{7.24}	59.22 _{3.03}	62.73 _{2.90}
	Oracle	51.77 _{1.00}	42.91 _{2.30}	64.13 _{0.18}	65.92 _{1.17}
	wGFN	64.17_{2.56}	47.00_{0.06}	67.58_{1.41}	70.62_{1.88}
CLINC-150	CoT	46.15 _{2.83}	31.81 _{0.55}	48.45 _{0.15}	52.74 _{1.08}
	Oracle	31.88 _{4.58}	15.57 _{5.04}	27.23 _{5.08}	38.54 _{4.41}
	wGFN	54.33_{1.73}	42.24_{0.88}	58.98_{0.41}	62.15_{1.31}
Emotion	CoT	32.12 _{3.76}	14.78 _{3.62}	24.16 _{2.64}	31.76 _{2.00}
	Oracle	33.22 _{0.16}	19.97 _{0.36}	33.66 _{1.00}	39.72 _{0.53}
	wGFN	43.41_{0.04}	24.02_{1.11}	42.48_{0.41}	46.80_{0.36}

Table 2: Comparison of CoT, oracle-only, and wGFN across three tasks; subscripts denote standard deviation. Best result per task and metric highlighted in **green**. We use Qwen-3-4B as our backbone for all methods.

	Model	Accuracy		ARI		NMI	
		wGFN	wGFN*	wGFN	wGFN*	wGFN	wGFN*
NERD	0.6B	22.65 _{2.85}	42.49_{0.79}	8.85 _{0.05}	26.12_{0.37}	18.59 _{0.44}	46.55_{0.26}
	4B	53.73 _{1.79}	59.67_{1.54}	36.74 _{2.16}	45.37_{1.71}	59.22 _{2.38}	67.03_{0.12}
	8B (DPMM)	61.32 _{4.28}		48.70 _{0.27}		66.36 _{1.89}	
CLINC-150	0.6B	21.11_{0.88}	20.89 _{1.41}	5.03 _{0.64}	8.73_{0.46}	15.26 _{2.19}	30.66_{2.02}
	4B	24.78 _{1.41}	37.33_{3.14}	13.00 _{2.12}	20.25_{0.17}	27.92 _{2.83}	41.25_{0.48}
	8B (DPMM)	45.89 _{0.16}		32.31 _{1.76}		57.28 _{0.57}	
Emotion	0.6B	20.32 _{1.73}	20.61_{1.66}	4.19 _{1.41}	4.79_{1.24}	11.77 _{0.94}	16.81_{1.56}
	4B	29.47_{1.03}	28.77 _{1.15}	11.66 _{1.39}	11.79_{0.91}	31.11 _{1.27}	31.35_{1.33}
	8B (DPMM)	33.96 _{0.05}		15.91 _{0.54}		27.90 _{0.07}	
	8B (CoT)	41.33 _{3.13}		22.49 _{2.98}		36.57 _{1.08}	

Table 3: Effect of DPMM warm-start on wGFN in the shuffled setting. wGFN* denotes wGFN initialised with DPMM cluster assignments. Qwen-3 8B DPMM and CoT results are standalone baselines.

Warm start of wGFN To address the oracle noise bottleneck in the shuffled setting, we initialise wGFN with the cluster assignments produced by a single pass of DPMM (denoted wGFN* in Table 3). We compare against Qwen-3-8B with both DPMM and CoT (deep thinking enabled) as upper-bound baselines. The warm start produces sizeable gains, for example, average NMI pre and post DPMM processing goes from 77% to 90% of the performance of CoT with Qwen-3-8B. Gains are modest for Emotion, reflecting the greater semantic ambiguity of that dataset. These results confirm that a single DPMM pass is a cost-effective warm start that substantially recovers the performance degradation caused by noisy few-shot demonstrations in the shuffled setting.

Runtime analysis We additionally report runtime/token-cost analysis for all methods in Appendix E.

	Model	Accuracy		ARI		NMI	
		wGFN	wGFN*	wGFN	wGFN*	wGFN	wGFN*
NERD	0.6B	22.65 _{2.85}	42.49_{0.79}	8.85 _{0.05}	26.12_{0.37}	18.59 _{0.44}	46.55_{0.26}
	4B	53.73 _{1.79}	59.67_{1.54}	36.74 _{2.16}	45.37_{1.71}	59.22 _{2.38}	67.03_{0.12}
	8B (Oracle)	52.83 _{0.95}		45.25 _{4.14}		59.31 _{3.12}	
CLINC-150	0.6B	21.11_{0.88}	20.89 _{1.41}	5.03 _{0.64}	8.73_{0.46}	15.26 _{2.19}	30.66_{2.02}
	4B	24.78 _{1.41}	37.33_{3.14}	13.00 _{2.12}	20.25_{0.17}	27.92 _{2.83}	41.25_{0.48}
	8B (Oracle)	24.88 _{0.68}		10.99 _{1.17}		25.07 _{1.39}	
Emotion	0.6B	20.32 _{1.73}	20.61_{1.66}	4.19 _{1.41}	4.79_{1.24}	11.77 _{0.94}	16.81_{1.56}
	4B	29.47_{1.03}	28.77 _{1.15}	11.66 _{1.39}	11.79_{0.91}	28.44 _{1.27}	31.35_{1.33}
	8B (Oracle)	23.42 _{1.66}		10.69 _{1.87}		25.53 _{1.31}	
	30B (Oracle)	22.37 _{1.97}		10.93 _{1.31}		31.03 _{1.54}	

Table 4: Effect of DPMM warm-start on wGFN in the shuffled setting. wGFN* denotes wGFN initialised with DPMM cluster assignments.

8 Conclusion

We introduced a two-component test-time gradient-free framework for FSOC with LLMs. A transferable wGFN policy provides amortised ID assignment and OOD rejection for novel support classes without parameter updates, while a vMF-DPMM warm start provides oracle-independent structure for OOD cluster formation under shuffled streams. Together, these components enable dynamically growing cluster inventories without data augmentation or predefined cluster counts.

9 Limitations

wGFN relies on autoregressive LLM scoring, so although it avoids test-time parameter updates and reduces exhaustive oracle calls, it remains more expensive than purely embedding-based clustering. Scaling to hundreds of clusters may therefore require batching, approximate candidate retrieval, a hybrid pipeline in which embedding-based clustering first narrows the candidate set, or parallel wGFN-agentic systems (which can be a future area of research). The oracle remains a potential failure point: although it is invoked only for OOD-routed examples and cluster-formation decisions, noisy provisional support sets can still propagate errors, especially under shuffled streams with semantically overlapping classes. Our experiments use fixed benchmark tasks and finite cluster inventories; future work can evaluate open-world streams with unknown class frequencies and concept drift.

References

Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S. Morcos. 2023. [Semdedup: Data-efficient learning at web-scale through semantic deduplication](#). *Preprint*, arXiv:2303.09540.

- Charles E. Antoniak. 1974. [Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems](#). *The Annals of Statistics*, 2(6):1152–1174.
- Yujia Bao, Menghua Wu, Shiyu Chang, and Regina Barzilay. 2020. [Few-shot text classification with distributional signatures](#). In *International Conference on Learning Representations*.
- Kayhan Batmanghelich, Ardavan Saeedi, Karthik Narasimhan, and Sam Gershman. 2016. [Nonparametric spherical topic modeling with word embeddings](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 537–542, Berlin, Germany. Association for Computational Linguistics.
- Yoshua Bengio, Salem Lahlou, Tristan Deleu, Edward J. Hu, Mo Tiwari, and Emmanuel Bengio. 2023. [Gflownet foundations](#). *Journal of Machine Learning Research*, 24(210):1–55.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Junfan Chen, Richong Zhang, Yongyi Mao, and Jie Xu. 2022. [Contrastnet: A contrastive learning framework for few-shot text classification](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):10492–10500.
- Ziye Chen, Cheng Ding, Zusheng Zhang, Yanghui Rao, and Haoran Xie. 2021. [Tree-structured topic modeling with nonparametric neural variational inference](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2343–2353, Online. Association for Computational Linguistics.
- Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tai, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, and 16 others. 2024. Scaling instruction-finetuned language models. *J. Mach. Learn. Res.*, 25(1).
- Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. 2023. [Why can GPT learn in-context? language models secretly perform gradient descent as meta-optimizers](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4005–4019, Toronto, Canada. Association for Computational Linguistics.
- Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Yihan Cao, Zihao Wu, Lin Zhao, Shaochen Xu, Fang Zeng, Wei Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, Hongmin Cai, Lichao Sun, Quanzheng Li, Dinggang Shen, Tianming Liu, and Xiang Li. 2025. [Auggpt: Leveraging chatgpt for text data augmentation](#). *IEEE Transactions on Big Data*, 11(3):907–918.
- Maarten De Raedt, Frédéric Godin, Thomas Demeester, and Chris Develder. 2023. [IDAS: Intent discovery with abstractive summarization](#). In *Proceedings of the 5th Workshop on NLP for Conversational AI (NLP4ConvAI 2023)*, pages 71–88, Toronto, Canada. Association for Computational Linguistics.
- Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. [GoEmotions: A dataset of fine-grained emotions](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4040–4054, Online. Association for Computational Linguistics.
- Jairo Diaz-Rodriguez. 2026. [Summaries as centroids for interpretable and scalable text clustering](#). In *The Fourteenth International Conference on Learning Representations*.
- Lu Fan, Jiashu Pu, Rongsheng Zhang, and Xiao-Ming Wu. 2024. [LANID: LLM-assisted new intent discovery](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10110–10116, Torino, Italia. ELRA and ICCL.
- Thomas S. Ferguson. 1973. [A Bayesian analysis of some nonparametric problems](#). *The Annals of Statistics*, 1(2):209–230.
- Cristiano Mesquita Garcia, Ramon Abilio, Alessandro Lameiras Koerich, Alceu de Souza Britto, and Jean Paul Barddal. 2025. [Concept drift adaptation in text stream mining settings: A systematic review](#). *ACM Trans. Intell. Syst. Technol.*, 16(2).
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. [Lora: Low-rank adaptation of large language models](#). *Preprint*, arXiv:2106.09685.
- Kuhwan Jeong, Minwoo Chae, and Yongdai Kim. 2023. [Online learning for the dirichlet process mixture model via weakly conjugate approximation](#). *Computational Statistics & Data Analysis*, 179:107626.
- Brian Kulis and Michael I. Jordan. 2012. Revisiting k-means: new algorithms via bayesian nonparametrics. In *Proceedings of the 29th International Conference on Machine Learning, ICML’12*, page 1131–1138, Madison, WI, USA. Omnipress.
- Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham

- Kakade, Prateek Jain, and Ali Farhadi. 2022. [Ma-tryoshka representation learning](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 30233–30249. Curran Associates, Inc.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. [Efficient memory management for large language model serving with pagedattention](#). In *Proceedings of the 29th Symposium on Operating Systems Principles, SOSP '23*, page 611–626, New York, NY, USA. Association for Computing Machinery.
- Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. [An evaluation dataset for intent classification and out-of-scope prediction](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1311–1316, Hong Kong, China. Association for Computational Linguistics.
- Soochan Lee, Junsoo Ha, Dongsu Zhang, and Gunhee Kim. 2020. [A neural dirichlet process mixture model for task-free continual learning](#). In *International Conference on Learning Representations*.
- Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. [The power of scale for parameter-efficient prompt tuning](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Xiang Lisa Li and Percy Liang. 2021. [Prefix-tuning: Optimizing continuous prompts for generation](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.
- Dahua Lin. 2013. Online learning of nonparametric mixture models via sequential variational approximation. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1, NIPS'13*, page 395–403, Red Hook, NY, USA. Curran Associates Inc.
- Viktor Losing, Barbara Hammer, and Heiko Wersing. 2018. [Incremental on-line learning: A review and comparison of state of the art algorithms](#). *Neurocomputing*, 275:1261–1274.
- Kanika Madan, Jarrod Rector-Brooks, Maksym Korablyov, Emmanuel Bengio, Moksh Jain, Andrei Cristian Nica, Tom Bosc, Yoshua Bengio, and Nikolay Malkin. 2023. [Learning GFlowNets from partial episodes for improved convergence and stability](#). In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 23467–23483. PMLR.
- Louis Mahon and Thomas Lukasiewicz. 2024. Hard regularization to prevent deep online clustering collapse without data augmentation. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence, AAAI 2024, Vancouver, BC, Canada, February 22 – 25, 2024*. AAAI Press.
- Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, Benjamin Bossan, and Marian Tietz. 2022. PEFT: State-of-the-art parameter-efficient fine-tuning methods. <https://github.com/huggingface/peft>.
- Leland McInnes, John Healy, and James Melville. 2020. [Umap: Uniform manifold approximation and projection for dimension reduction](#). *Preprint*, arXiv:1802.03426.
- MetaAI. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Sewon Min, Mike Lewis, Luke Zettlemoyer, and Han-naneh Hajishirzi. 2022. [MetaICL: Learning to learn in context](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2791–2809, Seattle, United States. Association for Computational Linguistics.
- Sebastião Miranda, Artūrs Znotiņš, Shay B. Cohen, and Guntis Barzdins. 2018. [Multilingual clustering of streaming news](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4535–4544, Brussels, Belgium. Association for Computational Linguistics.
- Yutao Mou, Keqing He, Yanan Wu, Zhiyuan Zeng, Hong Xu, Huixing Jiang, Wei Wu, and Weiran Xu. 2022. [Disentangled knowledge transfer for OOD intent discovery with unified contrastive learning](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 46–53, Dublin, Ireland. Association for Computational Linguistics.
- Chau Minh Pham, Alexander Hoyle, Simeng Sun, Philip Resnik, and Mohit Iyyer. 2024. [TopicGPT: A prompt-based topic modeling framework](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2956–2984, Mexico City, Mexico. Association for Computational Linguistics.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

- Gaurav Sahu, Olga Vechtomova, Dzmitry Bahdanau, and Issam Laradji. 2023. [PromptMix: A class boundary augmentation method for large language model distillation](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5316–5327, Singapore. Association for Computational Linguistics.
- Kailash Karthik Saravanakumar, Miguel Ballesteros, Muthu Kumar Chandrasekaran, and Kathleen MCKeown. 2021. [Event-driven news stream clustering using entity-aware contextual embeddings](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2330–2340, Online. Association for Computational Linguistics.
- Avyav Singh, Ekaterina Shutova, and Helen Yannakoudakis. 2024. [Learning new tasks from a few examples with soft-label prototypes](#). In *Proceedings of the 9th Workshop on Representation Learning for NLP (RepL4NLP-2024)*, pages 215–236, Bangkok, Thailand. Association for Computational Linguistics.
- Avyav Kumar Singh and Helen Yannakoudakis. 2025. [Few-shot open-set classification via reasoning-aware decomposition](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 13854–13875, Suzhou, China. Association for Computational Linguistics.
- Jake Snell, Kevin Swersky, and Richard Zemel. 2017. [Prototypical networks for few-shot learning](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Xiaoshuai Song, Keqing He, Pei Wang, Guanting Dong, Yutao Mou, Jingang Wang, Yunsen Xian, Xunliang Cai, and Weiran Xu. 2023. [Large language models meet open-world intent discovery and recognition: An evaluation of ChatGPT](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10291–10304, Singapore. Association for Computational Linguistics.
- Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari S. Morcos. 2022. [Beyond neural scaling laws: beating power law scaling via data pruning](#). In *Advances in Neural Information Processing Systems*.
- Simone Tedeschi and Roberto Navigli. 2022. [MultiNERD: A multilingual, multi-genre and fine-grained dataset for named entity recognition \(and disambiguation\)](#). In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 801–812, Seattle, United States. Association for Computational Linguistics.
- Yee Teh, David Newman, and Max Welling. 2006a. [A collapsed variational bayesian inference algorithm for latent dirichlet allocation](#). In *Advances in Neural Information Processing Systems*, volume 19. MIT Press.
- Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. 2006b. [Hierarchical dirichlet processes](#). *Journal of the American Statistical Association*, 101(476):1566–1581.
- Songsong Tian, Lusi Li, Weijun Li, Hang Ran, Xin Ning, and Prayag Tiwari. 2024. [A survey on few-shot class-incremental learning](#). *Neural Networks*, 169:307–324.
- Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. 2022. [Efficient few-shot learning without prompts](#). *Preprint*, arXiv:2209.11055.
- Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. 2016. [Matching networks for one shot learning](#). In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.
- Vijay Viswanathan, Kiril Gashteovski, Carolin Lawrence, Tongshuang Wu, and Graham Neubig. 2024. [Large language models enable few-shot clustering](#). *Transactions of the Association for Computational Linguistics*, 12:321–333.
- Huy V. Vo, Vasil Khalidov, Timothée Darcet, Théo Moutakanni, Nikita Smetanin, Marc Szafraniec, Hugo Touvron, camille couprie, Maxime Oquab, Armand Joulin, Herve Jegou, Patrick Labatut, and Piotr Bojanowski. 2024. [Automatic data curation for self-supervised learning: A clustering-based approach](#). *Transactions on Machine Learning Research*.
- Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. 2023. Transformers learn in-context by gradient descent. In *Proceedings of the 40th International Conference on Machine Learning, ICML’23*. JMLR.org.
- Mengting Wan, Tara Safavi, Sujay Kumar Jauhar, Yujin Kim, Scott Counts, Jennifer Neville, Siddharth Suri, Chirag Shah, Ryen W. White, Longqi Yang, Reid Andersen, Georg Buscher, Dhruv Joshi, and Nagu Rangan. 2024. [Tnt-llm: Text mining at scale with large language models](#). In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD ’24*, page 5836–5847, New York, NY, USA. Association for Computing Machinery.
- Zihan Wang, Jingbo Shang, and Ruiqi Zhong. 2023. [Goal-driven explainable clustering via language descriptions](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10626–10649, Singapore. Association for Computational Linguistics.
- Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022a. [Finetuned language models are zero-shot learners](#). In *International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022b. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS '22*, Red Hook, NY, USA. Curran Associates Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. **Transformers: State-of-the-art natural language processing**. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. **Qwen3 technical report**. *Preprint*, arXiv:2505.09388.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. **ZeroGen: Efficient zero-shot learning via dataset generation**. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11653–11669, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyoung Park. 2021. **GPT3Mix: Leveraging large-scale language models for text augmentation**. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2225–2239, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hanlei Zhang, Hua Xu, Ting-En Lin, and Rui Lyu. 2021a. Discovering new intents with deep aligned clustering. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14365–14373.

Haode Zhang, Haowen Liang, Yuwei Zhang, Liming Zhan, Xiaolei Lu, Albert Lam, and Xiao-Ming Wu. 2022a. **Fine-tuning pre-trained language models for few-shot intent detection: Supervised pre-training and isotropization**. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 532–542, Seattle, United States. Association for Computational Linguistics.

Jianguo Zhang, Trung Bui, Seunghyun Yoon, Xiang Chen, Zhiwei Liu, Congying Xia, Quan Hung Tran, Walter Chang, and Philip Yu. 2021b. **Few-shot intent detection via contrastive pre-training and fine-tuning**. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages

1906–1912, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yuwei Zhang, Zihan Wang, and Jingbo Shang. 2023. **ClusterLLM: Large language models as a guide for text clustering**. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13903–13920, Singapore. Association for Computational Linguistics.

Yuwei Zhang, Haode Zhang, Li-Ming Zhan, Xiao-Ming Wu, and Albert Lam. 2022b. **New intent discovery with pre-training and contrastive learning**. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 256–269, Dublin, Ireland. Association for Computational Linguistics.

A Experimental details for Section 3

Sampling latent sequences Singh and Yannakoudakis (2025) train a policy q_{GFN} by constructing episodic training sets of $N \cdot K$ episodes, each comprising few-shot demonstrations for N classes with K examples per class and a single query example which should be classified into the correct class. For each episode, latent sequences $Z \sim p_{LLM}(Z | X)$ are sampled from a frozen base LLM in two forms: *support latents*, which describe the class using few-shot demonstrations, and *query latents*, which describe only the query example itself (see examples in Figure 4 inspired by Singh and Yannakoudakis (2025)).

The test entity broadly represents *names of people*.
The test entity broadly represents *famous people*.
The test entity broadly represents *a famous neo-Gothic era English architect*.
The test entity broadly represents *a celebrated British architect*.

Figure 4: **Support** and **query** latents sampled from the base LLM for the entity “James Wyatt”.

Reward landscape shaping The GFN policy q_{GFN} is trained via the subtrajectory balance loss where the reward $R(z_{1:t}) = \prod p_{LLM}(X Z_{1:t} Y)$ incorporates a Wasserstein-interpolated negative offset

$$\Delta_t \sim -\mathcal{N}((1 - \alpha)\Delta_0 + \Delta \cdot \alpha, 1)$$

with $\alpha = t/T$ on correct-label trajectories. This is done to ensure stability in the policy training by gradually widening the reward margin rather than applying a constant penalty. The key idea is to train the policy model q_{GFN} such that a distribution of

latent sequences which corresponds to a particular label is sampled with a high reward while out-of-distribution latent sequences for a given set of labels are sampled with low rewards, thus yielding a clear dual objective of simultaneous in-distribution classification and out-of-distribution rejection for each test example.

Inference For OOD detection at inference time, a per-class Gaussian Process (GP) is fitted to the logit-normalised differences

$$\delta(XZY) = \log p_{LLM}(XZY) - \log q_{GFN}(XZY)$$

$\delta(XZY)$ is computed from negative (OOD) training examples for each class c (thus, the latent sequence Z does not correspond to the distribution of the latent distribution trained for the label Y which helps us define an OOD signal using logit distributions). The GP uses a linear mean function and a Matérn kernel (with $\nu = 5/2$). Given a test input, the KL divergence between the test sequence’s logit distribution $\mathcal{N}_1$ and the class-specific GP $\mathcal{N}_0^c$ is computed in closed form for each class c give as:

$$\text{KL}(\mathcal{N}_0^c \parallel \mathcal{N}_1) = \frac{1}{2} \sum_{t=1}^T \left[\log \frac{|\Sigma_1|}{|\Sigma_0|} - n + \text{tr}(\Sigma_1^{-1} \Sigma_0) + (\mu_1^t - \mu_0^t)^\top \Sigma_1^{-1} (\mu_1^t - \mu_0^t) \right]$$

where μ_0^t represents at the token t the mean of f_c and μ_1^t represents the same token-level mean function for the test sequence while Σ_0 and Σ_1 denote their respective covariance matrices (Singh and Yannakoudakis, 2025). Note that a high KL divergence between these two distributions can be interpreted as an in-distribution classification as the GP $\mathcal{N}_0^c$ represents the tokenwise distribution for an out-of-distribution signal, thus setting a baseline for an unrecognised latent sequence. See Figure 5 for a diagrammatic overview.

Constructing the few-shot training set For MultiNERD (Tedeschi and Navigli, 2022), 10 classes are sampled (*Person, Location, Animal, Cosmos, Disease, Food, Mythology, Plant, Vehicle, Time*) and anonymised with single capital letters. Training sets are constructed by sampling $N \in \{3 \dots 10\}$ support classes and $K = 4$ examples per class from the training split according to either the *Incremental* or *Disjoint* setting.

Constructing the test set Three test sets are created by randomly sampling 50 examples per class

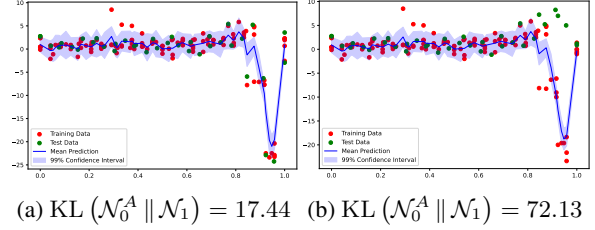


Figure 5: Note that an out-of-distribution logit distribution has low KL divergence with the GP defining the OOD signal for Class A (left), however an in-distribution latent has a high KL divergence (on the right) with the same class. Figure taken from Singh and Yannakoudakis (2025).

from each of the 15 classes in the test split. The remaining classes excluding the ones used in construction of the training set (*Organisation, Biological, Events, Instruments, Media*) in MultiNERD are sampled in addition to the previous 10 classes which are to be used for few-shot demonstrations. Given a few-shot support set consisting of demonstrations of seen classes, seen classes are evaluated under their anonymised label; while unseen classes (all classes which are not part of the set of seen classes) are labelled as *None of these*. Therefore, as an example, if we sample 3 classes as seen classes, we have to reject test examples from the remaining 12 classes as out-of-distribution for each test set. We repeat this experiment gradually increasing the number of seen classes.

Sample prompts We provide sample prompts for the *Incremental* and *Disjoint* settings in Figure 6 and Figure 7 respectively. Note that in the former, we gradually add classes which have not been explicitly trained on by wGFN while the latter can comprise a completely disjoint sets of support and training classes. Labels are randomly shuffled after each experiment to ensure strict generalisability and there is no semantic label $\leftrightarrow$ latent sequence mapping.

Hyperparameters and training details We present our hyperparameters for training the models with LORA adapters (Hu et al., 2021) with the $\mathcal{L}_{\text{SubTB}}(Z; \theta)$ objective defined in Equation 1 in Table 5. We train all models on a single 40GB A100 and evaluate on a single H200 NVIDIA GPU. Training q_{GFN} takes 15 – 30 minutes per task.

Results Results for the *Incremental* and *Disjoint* settings across all model sizes are reported in Tables 7, 8 and 9. For each row, $|N_{\text{test}}|$ denotes the cumulative number of evaluation classes; the first

In the sentence 'A large group of stars bound by gravitation is generally called a " star cluster " or " galaxy ", although, broadly speaking, they are also star systems.' the entity ['galaxy'] represents Class F.
In the sentence 'Possible causes are an accumulation of material from a binary companion through accretion, or a stellar merger.' the entity ['binary companion'] represents Class F.
In the sentence 'Fragmentation of the molecular cloud during the formation of protostars is an acceptable explanation for the formation of a binary or multiple star system.' the entity ['molecular cloud'] represents Class F.
In the sentence 'The candy is made from sugar, glucose syrup, gelatin, citric acid, flavoring, curcumin (coloring), carmine, and mixed carotenes.' the entity ['carmine'] represents Class I.
In the sentence 'Cendol " is a popular iced drink from this region containing chilled coconut milk and green jellies made of rice flour.' the entity ['Cendol'] represents Class I.
In the sentence 'Founded in 1965, the company sells soy milk, horchata, egg-free mayonnaise, chocolate and carob bars.' the entities ['soy milk', 'horchata'] represent Class I.
In the sentence 'Ares responded to this challenge by transforming himself into a bull and easily winning the contest.' the entity ['Ares'] represents Class L.
In the sentence 'The opera featured Philip Langridge as Oedipus and Jessye Norman as Jocasta.' the entities ['Oedipus', 'Jocasta'] represent Class L.
In the sentence 'In 1848 he independently co-discovered Hyperion.' the entity ['Hyperion'] represents Class L.
Which class does the test entity ['Anteros'] in the sentence 'The fountain is crowned by Alfred Gilbert 's aluminium statue of Anteros as a nude, butterfly-winged archer.' represent?

(a) Initial experiment where the model is trained on Class F, I, L and subsequently queried for either an in-distribution classification (either one of Class F, I, L) or out-of-distribution rejection (none of the above classes)

In the sentence 'A large group of stars bound by gravitation is generally called a " star cluster " or " galaxy ", although, broadly speaking, they are also star systems.' the entity ['galaxy'] represents Class F.
In the sentence 'Possible causes are an accumulation of material from a binary companion through accretion, or a stellar merger.' the entity ['binary companion'] represents Class F.
In the sentence 'Fragmentation of the molecular cloud during the formation of protostars is an acceptable explanation for the formation of a binary or multiple star system.' the entity ['molecular cloud'] represents Class F.
In the sentence 'The candy is made from sugar, glucose syrup, gelatin, citric acid, flavoring, curcumin (coloring), carmine, and mixed carotenes.' the entity ['carmine'] represents Class I.
In the sentence 'Cendol " is a popular iced drink from this region containing chilled coconut milk and green jellies made of rice flour.' the entity ['Cendol'] represents Class I.
In the sentence 'Founded in 1965, the company sells soy milk, horchata, egg-free mayonnaise, chocolate and carob bars.' the entities ['soy milk', 'horchata'] represent Class I.
In the sentence 'Ares responded to this challenge by transforming himself into a bull and easily winning the contest.' the entity ['Ares'] represents Class L.
In the sentence 'The opera featured Philip Langridge as Oedipus and Jessye Norman as Jocasta.' the entities ['Oedipus', 'Jocasta'] represent Class L.
In the sentence 'In 1848 he independently co-discovered Hyperion.' the entity ['Hyperion'] represents Class L.
In the sentence 'Thus, most mammals do not have complex color vision - they are dichromatic but they are sensitive to ultraviolet light, though they cannot see its colors.' the entity ['dichromatic'] represents Class G.
In the sentence 'Damage to these cortical regions can lead to deficits in consciousness such as hemispatial neglect.' the entity ['hemispatial neglect'] represents Class G.
In the sentence 'Female GPER knockout mice display hyperglycemia and impaired glucose tolerance, reduced body growth, and increased blood pressure.' the entity ['hyperglycemia'] represents Class G.
Which class does the test entity ['Anteros'] in the sentence 'The fountain is crowned by Alfred Gilbert 's aluminium statue of Anteros as a nude, butterfly-winged archer.' represent?

(b) In the next experiment, we randomly sample few-shot support demonstrations for a new class (given in red) which wGFN has not been trained on and repeat the experiment, with the new objective being in-distribution classification (either one of Class F, I, L, G) or out-of-distribution rejection (none of the above four classes) for each test example.

Figure 6: Experimental setup in the *Incremental* setting. Examples depicted in **green** represent the classes the model is trained on (N_{train}) while examples depicted in **red** represent few-shot demonstrations from novel classes which were not seen at training time

In the sentence '353 and Delwyn Young 's.' the entity ['Delwyn Young'] represents Class A.
In the sentence 'The game features a commentary team consisting of Bob Fitzgerald and Tom Tolbert ; Kevin Frazier hosts pre-game shows.' the entities ['Bob Fitzgerald', 'Tom Tolbert', 'Kevin Frazier'] represent Class A.
In the sentence 'In December 2015, it was confirmed that Joseph Trapanese would return to compose the score for the film.' the entity ['Joseph Trapanese'] represents Class A.
In the sentence 'One popular type is " Vollkornbrot " (wholegrain bread), which has a sourish full savoury taste, due to the use of sourdough as a leavening agent, and which often contains rye, albeit bread made from wheat flour is usually the most common variety.' the entities ['rye', 'wheat'] represent Class M.
In the sentence 'It typically consists of fava beans, chopped tomatoes, onion, parsley, lemon juice, olive oil, pepper and salt.' the entities ['tomatoes', 'onion', 'parsley', 'lemon'] represent Class M.
In the sentence 'It is used for tropical plants, such as fruiting banana plants, coffee, rubber and giant bamboo, and is kept at a tropical temperature and moisture level.' the entities ['banana', 'bamboo'] represent Class M.
In the sentence 'The delayed launch decision was a result of delays of the Airbus A380 and discussions on how to fund development.' the entity ['Airbus A380'] represents Class O.
In the sentence 'True GPs were discontinued after the completion of the last GP60 in 1994.' the entity ['GP60'] represents Class O.
In the sentence 'The first jet airliner, the de Havilland Comet, was introduced in 1952.' the entity ['de Havilland Comet'] represents Class O.
Which class does the test entity ['ornamental plant'] in the sentence 'It is widely cultivated throughout the tropics as an ornamental plant.' represent?

Figure 7: Experimental setup in the *Disjoint* setting. Note that sentences depicted in **red** represent few-shot demonstrations from novel classes which were not seen at training time. It is possible that in a particular experiment, we evaluate the model on a set of classes which are completely disjoint from the set of classes used for training.

row lists the base class set and subsequent rows indicate the class added at that step. We evaluate the macro F1 score on seen and unseen label spaces to account for any imbalance in the test set. Classes shown in **green** represent the classes the model was trained on while classes shown in **red** depict classes introduced only at test-time.

Comparison with existing work Singh and Yanakoudakis (2025) train separate policy models q_{GFN} for each support set sampled from their training distribution. To compare the performance of their explicitly trained policy models using classes contained N_{train} to the performance of our single model trained on the classes contained $N_{support}$, we conduct a small ablation study as described in Table 6. Note that $N_{support} \neq N_{train}$, therefore this directly shows the generalisable behaviour of q_{GFN} and how it needs no policy updates for a new test-time support class distribution. We evaluate in-distribution classification and out-of-distribution rejection on the classes contained in $N_{test} = N_{train}$. We can thus see that a policy model more or less maintains test-time performance on a completely different split from which it was trained on.

is

B Algorithmic details of FSOC

Complete online clustering algorithm Stream processing proceeds in two phases: online assignment and end-of-stream convergence. For each

arriving test point x_t , $\text{INFER}(x_t, \mathcal{S}, \delta)$ (Algorithm 1) evaluates whether x_t belongs to any confirmed cluster $c \in \mathcal{C}$ by maintaining a Dirichlet posterior over class assignments. Concretely, N_Z query latents $Z^{(i)} \sim p_{\text{LLM}}(\cdot | x_t)$ are sampled from the frozen base LLM, and for each confirmed cluster c the per-token logit-normalised differences $\delta_i^c = \log p_{\text{LLM}}(X Z^{(i)} Y_c) - \log q_{\text{GFN}}(X Z^{(i)} Y_c)$ are compared against the class-specific Gaussian Process $\mathcal{GP}_c$ via the KL divergence $\kappa_i^c = \text{KL}(\mathcal{GP}_c \| \mathcal{N}(\delta_i^c))$ (see Appendix A for the closed form). After all candidate classes have been scored, the mean KL scores are temperature-normalised globally into soft fractional pseudo-counts, which are accumulated into per-class Dirichlet concentration parameters α . The posterior mean $\hat{\pi}_c = \alpha_c / \sum_{j \in \mathcal{C}} \alpha_j$ summarises the evidence across all classes; if $\max_c \hat{\pi}_c \geq \delta$, the point is assigned in-distribution to $\arg \max_c \hat{\pi}_c$. Otherwise, x_t is routed to the oracle buffer $\mathcal{B}$, which accumulates provisional clusters until the promotion threshold K is reached, at which point a merge check is triggered with a cosine-decayed threshold δ_t . At end-of-stream, a convergence pass resolves all remaining provisional clusters by running INFER with the relaxed threshold $\delta_{\text{conv}} < \delta$.

Merging within the oracle buffer The oracle buffer $\mathcal{B}$ maintains a set of provisional clusters $\{b_j\}$, each collecting sentences from OOD test points that share a common semantic category.

Hyperparameter	Search Space	Value
Sampling temperature	{0.6, 0.7, 0.8, 0.9}	0.6
Modules trained	—	$[q_proj, v_proj, o_proj]$
LORA rank	{32, 64, 256}	256
LORA α	{4, 8, 16}	16
Learning rate	{1e-6 : 1e-4}	$5e - 5$
Learning rate scheduler	{ <i>cosine</i> , <i>uniform</i> }	<i>cosine</i>
Dropout	{0.0 : 0.5}	0.0
Epochs	{1, 3, 5, 10}	10
Batch size for latents	{16, 24, 32}	64
Length of latent Z	[10, 15]	12
Support latents	{48, 64}	64
Query latents	{8, 16}	8
Test latents	{9}	9
$k(x, x')$	RBF, Matérn{3/2, 5/2}	Matérn {5/2}
Δ	{50, 80, 90}	90

Table 5: Hyperparameters for training wGFN with Llama-3.2 3B and Llama-3.1 8B

N_{train}	q_{GFN} trained on	Seen	Unseen
Person, Plant, Vehicle	N_{train}	0.50 _{0.09}	0.58 _{0.07}
	$N_{support}$	0.50 _{0.05}	0.60 _{0.06}
Person, Plant, Time, Mythology	N_{train}	0.51 _{0.05}	0.55 _{0.08}
	$N_{support}$	0.54 _{0.03}	0.59 _{0.04}
Animal, Person, Time, Plant, Disease	N_{train}	0.52 _{0.04}	0.47 _{0.05}
	$N_{support}$	0.51 _{0.03}	0.51 _{0.04}

Table 6: Macro-F1 comparison with Singh and Yanakoudakis (2025) across different class splits; subscripts denote standard deviation. Best result per split and metric highlighted in green. Note that $N_{support} = \{\text{Cosmos, Disease, Mythology}\}$.

When an OOD test point x_t arrives, the oracle LLM is queried independently for each existing cluster $b_j \in \mathcal{B}$ using a structured three-step chain-of-thought prompt: (a) Get the high-level semantic category of x_t (b) Get the high-level semantic category of the existing cluster sentences $\{x_s\}_{s \in b_j}$ (c) Return *Yes* only if both categories are strictly the same. The verdict is extracted from the final sentence of the oracle’s chain-of-thought response. Three outcomes are possible: (i) if exactly one cluster b_j returns *Yes*, x_t is appended to b_j ; (ii) if multiple clusters return *Yes* (indicating co-referent provisional clusters that have been maintained separately), all matching clusters are merged into one and x_t is appended to the result; (iii) if no cluster matches, a fresh provisional cluster is initialised with x_t as its sole member. A cluster is promoted when $|b_j| = K$: a merge check against the confirmed cluster set $\mathcal{C}$ is triggered using either wGFN (see below) or the CoT baseline.

$ N_{test} $	Classes	Seen	Unseen
3	Cosmos, Food, Mythology	0.46 _{0.04}	0.70 _{0.11}
4	+ Animal	0.66 _{0.06}	0.71 _{0.08}
5	+ Location	0.64 _{0.05}	0.73 _{0.06}
6	+ Disease	0.67 _{0.02}	0.71 _{0.04}
7	+ Vehicle	0.66 _{0.02}	0.67 _{0.06}
8	+ Person	0.66 _{0.05}	0.63 _{0.01}
9	+ Plant	0.61 _{0.04}	0.63 _{0.05}
10	+ Time	0.57 _{0.02}	0.61 _{0.03}

(a) Incremental

$ N_{test} $	Classes	Seen	Unseen
3	Person, Plant, Vehicle	0.69 _{0.11}	0.76 _{0.10}
4	+ Location	0.77 _{0.06}	0.75 _{0.05}
5	+ Animal	0.66 _{0.01}	0.74 _{0.01}
6	+ Mythology	0.66 _{0.01}	0.71 _{0.01}
7	+ Food	0.67 _{0.01}	0.60 _{0.12}
8	+ Cosmos	0.69 _{0.03}	0.65 _{0.04}
9	+ Disease	0.60 _{0.06}	0.60 _{0.10}
10	+ Time	0.55 _{0.06}	0.55 _{0.10}

(b) Disjoint

Table 7: Macro-F1 results for Llama 3.1 8B; subscripts denote standard deviation.

Merging using wGFN When a promoted cluster b^* with sentences $\{x_s\}_{s \in \mathcal{I}}$ is evaluated for a merge, the cluster sentences serve as a mini-batch of test inputs to INFER with $J = J_{\text{merge}}$ (see Algorithm 1). For each x_s , N_Z query latents $Z^{(i)} \sim p_{\text{LLM}}(\cdot | x_s)$ are sampled from the *frozen base LLM* (i.e. without the GFN LoRA adapter), exactly as in standard query inference. The token-level logit differences δ_i^c and KL scores $\kappa_i^c = \text{KL}(\mathcal{GP}_c || \mathcal{N}(\delta_i^c))$ are computed against each confirmed cluster $c \in \mathcal{C}$, and the

$ N_{\text{test}} $	Classes	Seen	Unseen
3	Cosmos, Food, Mythology	0.48 _{0.07}	0.73 _{0.06}
4	+ Animal	0.52 _{0.03}	0.68 _{0.04}
5	+ Location	0.58 _{0.02}	0.64 _{0.01}
6	+ Disease	0.56 _{0.01}	0.59 _{0.03}
7	+ Vehicle	0.55 _{0.07}	0.55 _{0.04}
8	+ Person	0.59 _{0.03}	0.57 _{0.03}
9	+ Plant	0.50 _{0.04}	0.54 _{0.04}
10	+ Time	0.49 _{0.03}	0.53 _{0.02}

(a) Incremental

$ N_{\text{test}} $	Classes	Seen	Unseen
3	Person, Plant, Vehicle	0.50 _{0.05}	0.60 _{0.06}
4	+ Location	0.54 _{0.05}	0.60 _{0.01}
5	+ Animal	0.57 _{0.03}	0.55 _{0.04}
6	+ Mythology	0.54 _{0.03}	0.53 _{0.03}
7	+ Food	0.55 _{0.10}	0.49 _{0.11}
8	+ Cosmos	0.57 _{0.02}	0.48 _{0.04}
9	+ Disease	0.56 _{0.04}	0.48 _{0.05}
10	+ Time	0.57 _{0.03}	0.47 _{0.10}

(b) Disjoint

Table 8: Macro-F1 results for Llama 3.2 3B; subscripts denote standard deviation.

$ N_{\text{test}} $	Classes	Seen	Unseen
3	Cosmos, Food, Mythology	0.57 _{0.06}	0.63 _{0.03}
4	+ Animal	0.55 _{0.03}	0.54 _{0.05}
5	+ Location	0.56 _{0.07}	0.55 _{0.09}
6	+ Disease	0.57 _{0.04}	0.45 _{0.09}
7	+ Vehicle	0.53 _{0.05}	0.37 _{0.04}
8	+ Person	0.50 _{0.05}	0.27 _{0.09}
9	+ Plant	0.40 _{0.03}	0.25 _{0.10}
10	+ Time	0.39 _{0.00}	0.23 _{0.02}

(a) Incremental

$ N_{\text{test}} $	Classes	Seen	Unseen
3	Person, Plant, Vehicle	0.50 _{0.04}	0.57 _{0.10}
4	+ Location	0.56 _{0.06}	0.56 _{0.01}
5	+ Animal	0.48 _{0.03}	0.45 _{0.08}
6	+ Mythology	0.52 _{0.05}	0.45 _{0.08}
7	+ Food	0.52 _{0.03}	0.42 _{0.03}
8	+ Cosmos	0.46 _{0.09}	0.39 _{0.08}
9	+ Disease	0.48 _{0.03}	0.35 _{0.11}
10	+ Time	0.42 _{0.05}	0.30 _{0.10}

(b) Disjoint

Table 9: Macro-F1 results for Llama 3.2 1B; subscripts denote standard deviation.

soft fractional pseudo-counts are accumulated into a shared Dirichlet concentration vector α over $\mathcal{C}$. If the resulting posterior mean $\hat{\pi}_{c^*} = \alpha_{c^*} / \sum_{j \in \mathcal{C}} \alpha_j$ meets or exceeds the cosine-decayed threshold δ_t for any confirmed cluster c^* , the provisional cluster is merged into c^* and all indices $\mathcal{I}$ are retroactively assigned to c^* . If no class meets or exceeds δ_t , the provisional cluster is confirmed as a new cluster: it is added to the support $\mathcal{S}$ and all indices in $\mathcal{I}$ are assigned to the new label c_{new} .

Merge scheduling with wGFN The acceptance threshold δ_t used during the merge check follows a cosine decay schedule:

$$\delta_t = \delta_{\min} + (\delta_0 - \delta_{\min}) \cdot \frac{1}{2} \left(1 + \cos \left(\frac{\pi t}{T_s} \right) \right)$$

decaying from δ_0 to $\delta_{\min}$ over T_s merge events. A high initial threshold δ_0 enforces strict selectivity early in the stream when the confirmed cluster set $\mathcal{C}$ is small and a false merge would corrupt many future assignments; the threshold relaxes toward $\delta_{\min}$ as more clusters are confirmed and the stream provides richer distributional context. In the reported experiments, first-time ID/OOD inference uses $J_{\text{ID}} = 1$, while merge decisions use $J_{\text{merge}} = 3$ and therefore apply the top- k refinement step. To focus computation during the merge, for iterations $j > 1$ in INFER only the top- k classes

under the current posterior mean $\hat{\pi}$ are retained as merge candidates, with their concentrations reset uniformly to $1/k$; all remaining classes are excluded from subsequent partitions.

Merging using CoT When a promoted cluster b^* with sentences $\{x_s\}_{s \in \mathcal{I}}$ is evaluated for a merge under the CoT baseline, a pairwise semantic similarity check is performed against each confirmed cluster $c \in \mathcal{C}$. The cluster sentences are formatted as few-shot demonstrations under a randomly sampled dummy label $\ell \in \mathcal{A} \setminus \mathcal{C}$ (an uppercase letter not currently assigned to any confirmed cluster), producing strings of the form “ x_s represents [LABEL] ℓ ” and the model is queried if the two clusters represent the same, high-level semantic category. The Yes/No verdict is extracted from the final sentence of the chain-of-thought response. A greedy first-Yes strategy is applied: the first confirmed cluster c for which the model returns *Yes* absorbs the provisional cluster retroactively, assigning all indices $\mathcal{I}$ to c . If no confirmed cluster returns *Yes*, the provisional cluster is promoted as a new confirmed cluster with dummy label ℓ , and its sentences are added to the support $\mathcal{S}$. The oracle buffer logic and the end-of-stream convergence pass are otherwise identical between the wGFN and CoT variants.

You are an assistant who will reply to the question clearly and succinctly. Finish each sentence with a full stop.

In the sentence 'He was the backup of Klaus-Dietrich Flade for the Soyuz TM-14 mission.' the entity ['Soyuz TM-14'] represents Class O.

In the sentence 'The Avenis Verso was a large MPV introduced in 2001, with room for seven occupants and only initially available with 2.0 - litre engines.' the entity ['Avenis Verso'] represents Class O.

In the sentence 'It is a four-door coupé featuring a new COMAND interface utilizing a touchpad, design cues inspired by the Mercedes SLS AMG, and sliding rear doors.' the entity ['Mercedes SLS AMG'] represents Class O.

In the sentence '"Pouteria virescens" is a species of plant in the family Sapotaceae.' the entities ['plant', 'Sapotaceae'] represent Class M.

In the sentence 'The amount can be less than one percent in animals consuming less digestible plants, and it can be as high as forty percent in zooplankton consuming phytoplankton.' the entity ['phytoplankton'] represents Class M.

In the sentence 'When possible, three washings are performed: first with water infused with plum leaves, then with water infused with camphor, and lastly with purified water.' the entity ['plum'] represents Class M.

In the sentence 'Minister Margaret Thatcher complained when, by adroit image editing, the programme implied she had crashed a car.' the entity ['Margaret Thatcher'] represents Class A.

In the sentence 'He generally skated with captain Jonathan Toews on the team's first line.' the entity ['Jonathan Toews'] represents Class A.

In the sentence 'He is buried in Vilnius Cathedral together with his brother Karigaila, who died in the civil war in 1390.' the entity ['Karigaila'] represents Class A.

Which class does the entity ['William F. Romain'] in the sentence 'William F. Romain has suggested an array of lunar alignments based on the curves in the effigy's body.' represent? Think step by step and reply with the correct Class. Reply with 'None of these' if the answer does not fall in any Class.

To determine the Class of the entity 'William F. Romain', we analyze the context of the sentence. The phrase 'has suggested' implies an action typically performed by a human expert or professional. Class A corresponds to human beings or professionals. Therefore, the entity ['William F. Romain'] represents Class A.

Figure 8: Few-shot clustering via Llama-3.2 3B using multi-step chain-of-thought (CoT) reasoning on MultiNERD (Tedeschi and Navigli, 2022). The model response is highlighted in green.

Hyperparameters Table 10 lists the hyperparameters specific to inference and online clustering. Training hyperparameters (LoRA rank, learning rate, etc.) are reported separately in Appendix A.

Libraries We use the Hugging Face Transformers library (Wolf et al., 2020) for model loading and tokenisation and PEFT (Mangrulkar et al., 2022) for applying LoRA adapters to the wGFN backbone. The CoT model and oracle are served using vLLM (Kwon et al., 2023), which enables efficient batched generation with PagedAttention and substantially reduces per-query latency during the on-line evaluation loop.

C Results of wGFN and CoT with Llama

Metrics evaluated To measure clustering quality we report four complementary metrics. Accuracy applies the Hungarian algorithm to find the optimal label assignment between predicted and true classes before computing accuracy. Adjusted Rand Index (ARI) measures the fraction of concordantly grouped or separated example pairs, corrected for chance with range $[-1, 1]$. Adjusted Mutual Information (AMI) measures mu-

tual information between two partitions adjusted for chance, making it insensitive to the number of clusters. Normalized Mutual Information (NMI) similarly measures shared information but normalises by cluster entropy, yielding a score in $[0, 1]$ without a chance correction.

Detailed results We present our results in Table 11. The equivalent difference in performance in percentage points ($\Delta_{pp} = wGFN - CoT$) is presented in Figure 3, Section 5.

D Derivation of Online DPMM with vMF

This appendix provides the complete derivations underlying the online DPMM described in Section 6. We derive the component posteriors, the batch update rules via subtract-and-rescore and describe the full algorithm.

D.1 Generative Model and CRP Structure

We work with a Dirichlet Process Mixture of von Mises-Fisher distributions on the unit sphere $\mathcal{S}^{d-1}$. In the Chinese Restaurant Process (CRP) represen-

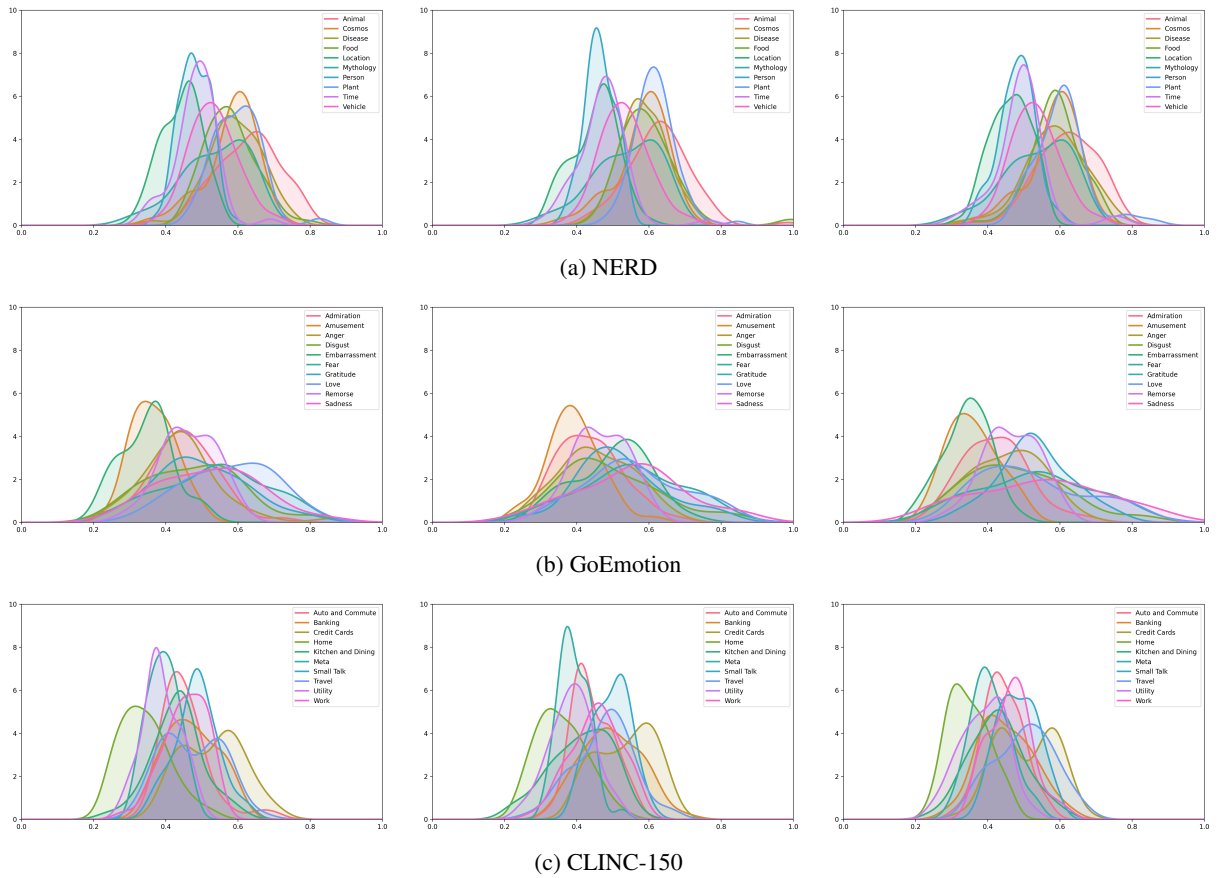


Figure 9: Distribution of cosine similarity between each test point and its true cluster centroid, computed with [Gemma-Embedding](#), for NERD, CLINC-150, and Emotion. NERD exhibits higher mean similarity and lower variance, indicating tightly-aligned clusters; Emotion shows lower mean similarity and higher spread, reflecting greater intra-class ambiguity while CLINC-150 falls in between the two.

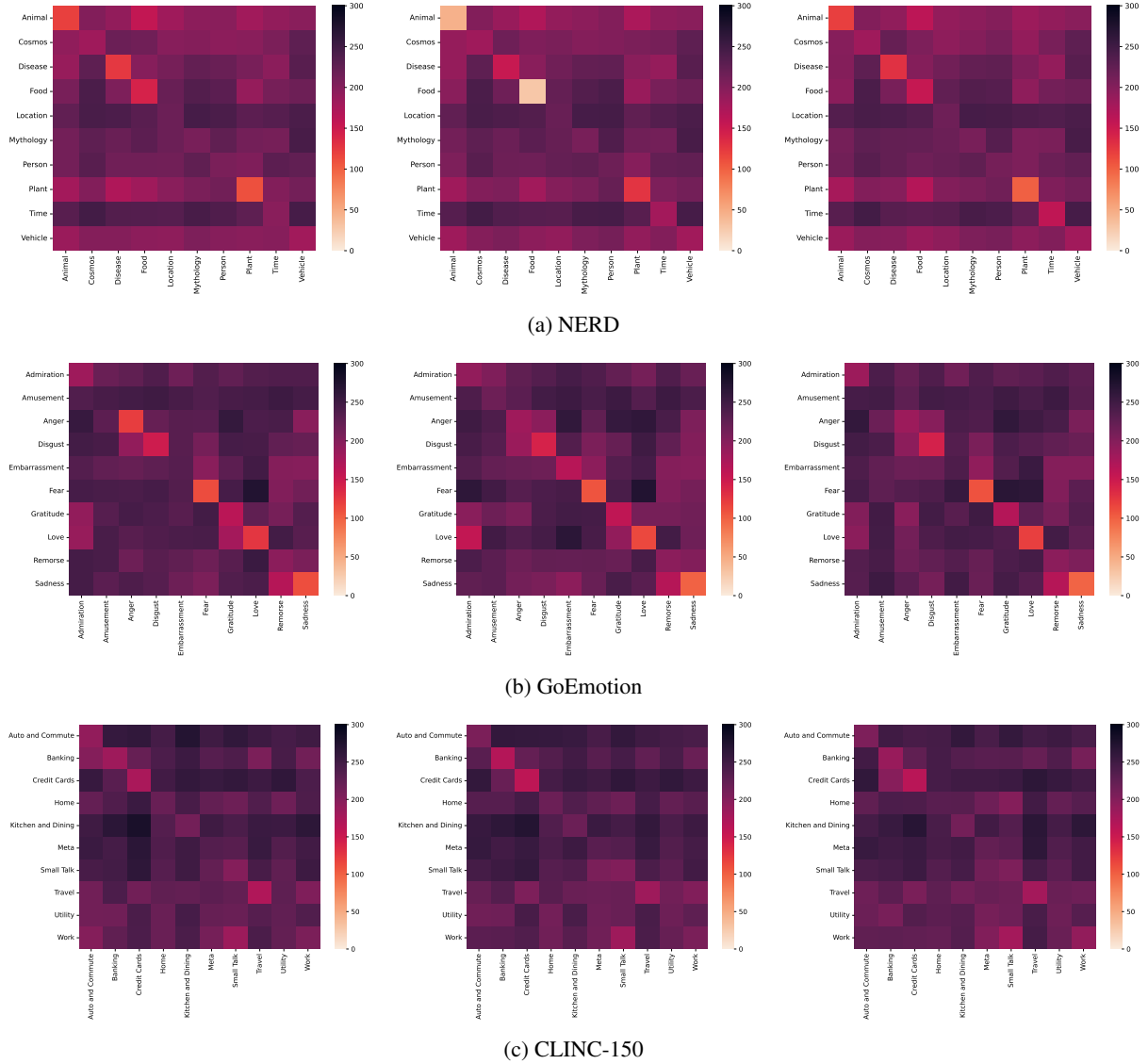


Figure 10: Pairwise Mahalanobis distances between class cluster distributions across the three tasks computed with [Gemma-Embedding](#) for all tasks. Each row shows the three test splits for NERD, GoEmotion and CLINC-150 respectively. Each heatmap reports the Mahalanobis distance between pairs of class distributions; lower values indicate clusters that are statistically closer while higher values reflect far-off class distributions. We observe that NERD clusters are generally closer in high-dimensional embeddings as opposed to GoEmotion/CLINC-150 which can aid in identification of similar clusters as these NERD clusters are less sparsely distributed as compared to GoEmotion/CLINC-150.

Hyperparameter	Description	Value
α_0	Initial symmetric Dirichlet concentration per class	1/3
τ	Temperature scaling for KL scores	{8 – 40} (depends on task)
δ	In-distribution acceptance threshold	{0.8 – 0.9} (depends on task)
J_{ID}	wGFN refinement iterations for first-time ID/OOD inference	1
J_{merge}	wGFN refinement iterations for merge decisions	2
k	Top- k classes retained after first iteration during merge refinement	3
$n_{\min} / n_{\max}$	Min / max classes per partition	3 / 6
N_Z	Latent trajectories per test point	9
τ_Z	Token sampling temperature	0.6
K	Provisional cluster promotion threshold	4
δ_0	Initial merge acceptance threshold	0.9
$\delta_{\min}$	Final merge threshold (cosine decay)	0.9 – 0.75 (depends on task)
T_s	Cosine decay schedule length (steps)	300
δ_{conv}	End-of-stream convergence threshold	0.5
$n_{\max}^m$	Max classes per partition (merge)	5

Table 10: Inference and online clustering hyperparameters with wGFN

tation the generative model is:

$$\begin{aligned}
\mathbf{z}_{1:n} &\sim \text{CRP}(\alpha) \\
\phi_k &\sim \text{vMF}(\boldsymbol{\mu}_0, \kappa_0) \quad \forall k \\
\mathbf{x}_i &\sim \text{vMF}(\phi_{z_i}, \kappa) \quad \forall i
\end{aligned}$$

where $\alpha > 0$ is the DP concentration, $\boldsymbol{\mu}_0 \in \mathcal{S}^{d-1}$ and $\kappa_0 > 0$ are the prior mean direction and concentration, and $\kappa > 0$ is the likelihood concentration. The vMF density on $\mathcal{S}^{d-1}$ is

$$\begin{aligned}
\text{vMF}(\mathbf{x} \mid \boldsymbol{\mu}, \kappa) &= C_d(\kappa) \exp(\kappa \boldsymbol{\mu}^\top \mathbf{x}) \\
\text{where } C_d(\kappa) &= \frac{\kappa^{d/2-1}}{(2\pi)^{d/2} I_{d/2-1}(\kappa)}
\end{aligned}$$

where I_ν is the modified Bessel function of the first kind of order ν . The CRP prior over partitions depends only on cluster sizes:

$$p(\mathbf{z}_{1:n} \mid \alpha) = \frac{\Gamma(\alpha) \alpha^K}{\Gamma(\alpha + n)} \prod_{k=1}^K \Gamma(|C_k|)$$

where K is the number of clusters and $|C_k|$ is the size of cluster k . Because the CRP is exchangeable, the conditional $p(z_i = k \mid \mathbf{z}_{-i})$ is the same regardless of the processing order of observations; this is the property that licenses the batch updates in Section D.7.

D.2 Lin’s Sequential Update Rules

Lin (2013) derives two update rules for the sequential variational approximation. When observation

$\mathbf{x}_{i+1}$ arrives and the current model has K clusters with weights $w_k^{(i)}$ and component posteriors $\nu_k^{(i)}$, the updates are:

Responsibility update (Lin’s Eq. 15)

$$\rho_{i+1}(k) \propto \begin{cases} w_k^{(i)} \int F(\mathbf{x}_{i+1} \mid \theta) \nu_k^{(i)}(d\theta) & k \leq K \\ \alpha \int F(\mathbf{x}_{i+1} \mid \theta) \mu(d\theta) & k = K+1 \end{cases}$$

Posterior update (Lin’s Eq. 16)

$$\nu_k^{(i+1)}(d\theta) \propto \mu(d\theta) \prod_{j=1}^{i+1} F(\mathbf{x}_j \mid \theta)^{\rho_j(k)} \quad (6)$$

We evaluate both integrals in closed form for the vMF conjugate case.

D.3 Collapsed Predictive for an Existing Cluster

Given the current posterior $\nu_k = \text{vMF}(\phi_k \mid \mathbf{r}_k / \|\mathbf{r}_k\|, \|\mathbf{r}_k\|)$ where $\mathbf{r}_k$ is the resultant vector, we evaluate the integral in Equation 6:

$$\begin{aligned}
&\int \text{vMF}(\mathbf{x}_i \mid \phi_k, \kappa) \nu_k(d\phi_k) \\
&= \int C_d(\kappa) C_d(\|\mathbf{r}_k\|) \exp(\kappa \phi_k^\top \mathbf{x}_i + \mathbf{r}_k^\top \phi_k) d\phi_k \\
&= C_d(\kappa) C_d(\|\mathbf{r}_k\|) \int \exp((\mathbf{r}_k + \kappa \mathbf{x}_i)^\top \phi_k) d\phi_k
\end{aligned}$$

The integrand is an unnormalised vMF with natural parameter $\mathbf{r}_k + \kappa \mathbf{x}_i$. Since $C_d(\|\mathbf{r}_k +$

	Task	Model	Accuracy		ARI		AMI		NMI	
			CoT	wGFN	CoT	wGFN	CoT	wGFN	CoT	wGFN
Class Incremental	NERD	Llama-3.2 3B	26.68 _{1.90}	53.13_{2.05}	10.16 _{1.25}	39.76_{4.02}	14.75 _{1.54}	53.52_{2.53}	21.86 _{1.30}	57.68_{2.90}
		Llama-3.1 8B	50.65 _{3.77}	67.01_{2.09}	27.67 _{2.34}	52.17_{3.63}	42.18 _{2.54}	68.19_{3.05}	44.38 _{3.08}	70.98_{2.11}
	CLINC-150	Llama-3.2 3B	37.78 _{0.63}	59.85_{2.24}	16.58 _{0.42}	37.47_{1.91}	22.69 _{0.19}	56.14_{2.53}	28.30 _{0.29}	59.08_{3.27}
		Llama-3.1 8B	45.78 _{6.29}	61.88_{2.17}	28.19 _{3.62}	41.49_{3.77}	37.38 _{4.97}	59.35_{2.25}	41.64 _{4.08}	61.39_{2.51}
	Emotion	Llama-3.2 3B	25.00 _{2.25}	48.21_{3.29}	8.61 _{1.75}	30.37_{2.90}	13.48 _{1.45}	47.88_{3.51}	21.18 _{0.38}	49.57_{4.52}
		Llama-3.1 8B	29.09 _{0.32}	53.03_{2.88}	11.28 _{2.16}	37.28_{3.42}	18.60 _{1.80}	56.33_{3.02}	25.61 _{1.24}	58.37_{3.72}
Shuffled	NERD	Llama-3.2 3B	29.60 _{1.27}	49.77_{3.53}	11.47 _{1.22}	33.44_{2.78}	17.22 _{0.35}	49.07_{1.72}	23.73 _{0.04}	52.15_{2.15}
		Llama-3.1 8B	48.65 _{1.59}	60.08_{3.58}	27.51 _{0.19}	47.62_{2.18}	38.79 _{1.47}	59.53_{1.83}	43.77 _{1.23}	62.37_{2.03}
	CLINC-150	Llama-3.2 3B	31.33_{2.20}	31.33_{2.83}	14.61_{1.30}	12.97 _{2.37}	19.49 _{0.95}	20.45_{3.99}	23.73 _{2.22}	26.02_{3.57}
		Llama-3.1 8B	38.56 _{0.16}	42.56_{1.10}	17.29 _{0.14}	24.80_{0.81}	28.89 _{0.52}	39.23_{1.60}	33.68 _{0.15}	41.94_{0.66}
	Emotion	Llama-3.2 3B	22.37 _{1.16}	31.26_{1.16}	8.42 _{2.00}	12.74_{1.79}	10.69 _{0.10}	20.37_{2.02}	18.40 _{1.30}	26.17_{3.07}
		Llama-3.1 8B	28.45 _{0.17}	31.03_{1.17}	9.47 _{2.21}	12.91_{0.62}	14.40 _{1.81}	20.95_{1.36}	20.41 _{1.12}	27.31_{0.82}

(a) Support invariance

	Task	Model	Accuracy		ARI		AMI		NMI	
			CoT	wGFN	CoT	wGFN	CoT	wGFN	CoT	wGFN
Class Incremental	NERD	Llama-3.2 3B	32.36 _{1.35}	56.50_{1.16}	13.42 _{0.55}	44.14_{1.36}	20.34 _{1.16}	61.16_{0.77}	26.46 _{0.96}	63.16_{0.40}
		Llama-3.1 8B	53.44 _{1.07}	65.79_{1.01}	32.74 _{0.88}	51.73_{2.78}	44.16 _{0.86}	66.96_{0.92}	48.10 _{0.49}	69.06_{0.86}
	CLINC-150	Llama-3.2 3B	36.67 _{1.42}	62.22_{2.77}	15.37 _{0.91}	42.23_{3.64}	21.79 _{2.78}	58.81_{3.50}	27.08 _{2.80}	61.69_{3.41}
		Llama-3.1 8B	47.04 _{1.03}	62.67_{2.30}	26.18 _{1.58}	42.78_{3.19}	37.24 _{1.35}	60.76_{1.68}	40.71 _{1.62}	62.88_{1.89}
	Emotion	Llama-3.2 3B	23.96 _{0.31}	52.04_{3.01}	8.13 _{0.62}	31.68_{2.66}	14.18 _{1.19}	51.68_{1.16}	21.75 _{1.46}	54.13_{1.18}
		Llama-3.1 8B	28.75 _{0.46}	53.45_{2.59}	12.20 _{0.52}	36.44_{2.83}	18.38 _{0.84}	55.25_{4.28}	23.93 _{1.15}	59.79_{4.56}
Shuffled	NERD	Llama-3.2 3B	30.83 _{3.74}	50.11_{1.11}	14.14 _{1.44}	34.73_{3.02}	21.14 _{1.48}	50.20_{0.72}	27.92 _{1.35}	53.68_{0.25}
		Llama-3.1 8B	50.82 _{2.13}	54.37_{3.33}	29.47 _{2.48}	41.95_{4.61}	41.92 _{2.59}	56.30_{3.22}	45.98 _{2.35}	59.56_{2.82}
	CLINC-150	Llama-3.2 3B	34.07_{0.46}	32.44 _{1.74}	13.55_{1.43}	13.17 _{2.05}	18.69 _{1.31}	20.89_{2.40}	23.72 _{1.10}	26.42_{2.52}
		Llama-3.1 8B	40.81 _{1.79}	41.85_{0.34}	20.00 _{1.14}	23.87_{1.32}	29.38 _{2.93}	39.45_{1.50}	33.40 _{2.82}	41.39_{1.10}
	Emotion	Llama-3.2 3B	24.28 _{2.84}	30.21_{1.63}	7.64 _{0.92}	11.61_{2.06}	12.25 _{1.30}	19.13_{2.74}	19.99 _{1.86}	24.69_{3.24}
		Llama-3.1 8B	28.73 _{1.12}	31.03_{2.87}	10.97 _{1.72}	12.02_{1.94}	15.59 _{1.85}	19.69_{2.50}	21.84 _{1.93}	26.95_{1.01}

(b) Order invariance

	Task	Model	Accuracy		ARI		AMI		NMI	
			CoT	wGFN	CoT	wGFN	CoT	wGFN	CoT	wGFN
Class Incremental	NERD	Llama-3.2 3B	31.62 _{3.26}	62.78_{2.78}	12.44 _{2.72}	45.14_{3.49}	18.63 _{0.92}	61.16_{1.16}	25.51 _{0.98}	63.73_{1.20}
		Llama-3.1 8B	51.35 _{4.02}	63.45_{1.93}	30.01 _{5.13}	52.86_{2.60}	40.61 _{2.60}	67.43_{1.48}	45.30 _{2.39}	69.77_{1.96}
	CLINC-150	Llama-3.2 3B	32.03 _{1.68}	61.78_{5.61}	16.77 _{2.86}	41.80_{7.34}	22.93 _{2.81}	58.47_{7.25}	27.90 _{2.62}	61.32_{7.06}
		Llama-3.1 8B	44.67 _{4.46}	62.44_{2.84}	25.04 _{3.94}	45.85_{2.52}	35.80 _{4.86}	62.32_{2.91}	39.00 _{4.92}	64.44_{3.18}
	Emotion	Llama-3.2 3B	22.05 _{1.49}	52.41_{3.01}	7.54 _{3.25}	30.89_{3.74}	12.00 _{3.05}	49.08_{2.88}	19.50 _{2.55}	53.37_{2.92}
		Llama-3.1 8B	27.80 _{2.34}	53.45_{4.22}	10.01 _{2.06}	35.87_{4.76}	16.44 _{2.60}	56.20_{5.12}	22.83 _{1.79}	60.36_{5.07}
Shuffled	NERD	Llama-3.2 3B	28.55 _{2.69}	46.79_{3.65}	9.01 _{4.11}	31.49_{6.05}	14.82 _{5.26}	44.28_{6.17}	21.99 _{5.47}	48.49_{4.98}
		Llama-3.1 8B	49.33_{3.02}	49.22 _{0.16}	30.28 _{3.74}	39.08_{7.57}	41.10 _{3.07}	53.36_{6.19}	45.44 _{2.77}	57.40_{5.44}
	CLINC-150	Llama-3.2 3B	30.96_{2.73}	30.15 _{2.94}	12.87 _{1.18}	12.99_{1.45}	17.86 _{1.40}	20.79_{2.36}	23.07 _{0.86}	26.66_{1.65}
		Llama-3.1 8B	36.07 _{2.37}	41.04_{3.39}	16.16 _{2.04}	24.07_{1.15}	22.03 _{2.59}	39.54_{1.18}	26.16 _{2.52}	41.06_{1.25}
	Emotion	Llama-3.2 3B	22.80 _{1.05}	28.73_{3.66}	6.35 _{1.76}	12.05_{1.97}	10.34 _{2.56}	20.79_{1.40}	18.18 _{3.01}	25.88_{1.86}
		Llama-3.1 8B	25.68 _{2.87}	31.54_{0.49}	8.99 _{2.35}	13.08_{0.72}	12.38 _{3.50}	20.54_{1.84}	18.85 _{3.25}	25.23_{2.62}

(c) Initialisation invariance

Table 11: Clustering performance (Accuracy, ARI, AMI, NMI) across tasks and stream orderings for Llama-3.2 3B and Llama-3.1 8B, comparing CoT and wGFN. Subscripts denote standard deviation. **Green** entries indicate the best result per row and metric.

Algorithm 1 INFER: Dirichlet-posterior wGFN inference

Require: test point or promoted mini-batch x ; support $\mathcal{S}$; threshold δ ; wGFN q_{GFN} ; GPs $\{\mathcal{GP}_c\}_{c \in \mathcal{C}}$; $\alpha_0, \tau, J, k, n_{\min}, n_{\max}, N_Z$

Ensure: $\hat{c} \in \mathcal{C}$ (ID) or $\emptyset$ (OOD)

- 1: $Z^{(1:N_Z)} \sim p_{\text{LLM}}(\cdot | x)$ $\triangleright$ sample query latents
- 2: $\alpha_c \leftarrow \alpha_0 \forall c \in \mathcal{C}$ $\triangleright$ initialise symmetric Dirichlet prior
- 3: **for** $j = 1, \dots, J$ **do**
- 4: **if** $j > 1$ **then**
- 5: $\mathcal{C} \leftarrow \text{top-}k(\mathcal{C}, \hat{\pi})$
- 6: $\alpha_c \leftarrow 1/k \forall c \in \mathcal{C}$ $\triangleright$ retain top- k classes and reset local prior
- 7: **end if**
- 8: Partition $\mathcal{C}$ into groups $\{\mathcal{P}_l\}$ with $|\mathcal{P}_l| \in [n_{\min}, n_{\max}]$
- 9: **for** each partition $\mathcal{P}_l$ **do**
- 10: **for** $i = 1, \dots, N_Z$ **do**
- 11: **for** each $c \in \mathcal{P}_l$ **do**
- 12: $\delta_i^c \leftarrow \log p(XZ^{(i)}Y_c) - \log q(XZ^{(i)}Y_c)$
- 13: $\kappa_i^c \leftarrow \text{KL}(\mathcal{GP}_c \| \mathcal{N}(\delta_i^c))$
- 14: **end for**
- 15: **end for**
- 16: $d_c \leftarrow \frac{1}{N_Z} \sum_{i=1}^{N_Z} |\kappa_i^c| \quad \forall c \in \mathcal{P}_l$
- 17: **end for**
- 18: $\tilde{\alpha}_c \leftarrow \frac{\exp(d_c/\tau)}{\sum_{c' \in \mathcal{C}} \exp(d_{c'}/\tau)} \quad \forall c \in \mathcal{C}$ $\triangleright$ global soft fractional pseudo-counts
- 19: $\alpha_c \leftarrow \tilde{\alpha}_c \quad \forall c \in \mathcal{C}$
- 20: $\hat{\pi}_c \leftarrow \frac{\alpha_c}{\sum_{c' \in \mathcal{C}} \alpha_{c'}} \quad \forall c \in \mathcal{C}$
- 21: $\hat{c} \leftarrow \arg \max_c \hat{\pi}_c$
- 22: **if** $\hat{\pi}_{\hat{c}} \geq \delta$ **then**
- 23: **return** $\hat{c}$
- 24: **end if**
- 25: **end for**
- 26: **return** $\emptyset$

$\kappa \mathbf{x}_i \|) \exp((\mathbf{r}_k + \kappa \mathbf{x}_i)^\top \phi_k)$ integrates to 1 over $\mathcal{S}^{d-1}$:

$$\int \exp((\mathbf{r}_k + \kappa \mathbf{x}_i)^\top \phi_k) d\phi_k = \frac{1}{C_d(\|\mathbf{r}_k + \kappa \mathbf{x}_i\|)}$$

Therefore the collapsed predictive likelihood for an existing cluster is:

$$\begin{aligned} & \int \text{vMF}(\mathbf{x}_i | \phi_k, \kappa) \nu_k(d\phi_k) \\ &= C_d(\kappa) \cdot \frac{C_d(\|\mathbf{r}_k\|)}{C_d(\|\mathbf{r}_k + \kappa \mathbf{x}_i\|)} \end{aligned} \quad (7)$$

D.4 Collapsed Predictive for a New Cluster

For a new cluster, ν_{K+1} is the prior $\mu = \text{vMF}(\cdot | \mu_0, \kappa_0)$. The computation is identical with $\mathbf{r}_k$ replaced by $\kappa_0 \mu_0$:

$$\begin{aligned} & \int \text{vMF}(\mathbf{x}_i | \phi, \kappa) \mu(d\phi) \\ &= C_d(\kappa) \cdot \frac{C_d(\kappa_0)}{C_d(\|\kappa_0 \mu_0 + \kappa \mathbf{x}_i\|)} \end{aligned} \quad (8)$$

Algorithm 2 FSOC with wGFN

Require: stream $x_{1:T}$; wGFN q_{GFN} ; GPs $\{\mathcal{GP}_c\}$; support $\mathcal{S}$; $\delta, \delta_0, \delta_{\min}, \delta_{\text{conv}}, T_s, K, J_{\text{ID}}, J_{\text{merge}}, n_{\max}^m$

Ensure: assignments $\hat{y}_{1:T}$

- 1: $\hat{\mathbf{y}} \leftarrow -1; \mathcal{B} \leftarrow \emptyset$
- 2: **for** $t = 1, \dots, T$ **do**
- 3: $\hat{c} \leftarrow \text{INFER}(x_t, \mathcal{S}, \delta; J = J_{\text{ID}})$ $\triangleright$ Algorithm 1
- 4: **if** $\hat{c} \neq \emptyset$ **then**
- 5: $\hat{y}_t \leftarrow \hat{c}$
- 6: **else**
- 7: $(\text{ready}, b^*) \leftarrow \text{ORACLEUPDATE}(\mathcal{B}, x_t)$
- 8: **if** ready **then** $\triangleright b^*$ has reached K examples
- 9: $\{x_s\}_{s \in \mathcal{I}}, \mathcal{I} \leftarrow \text{POP}(\mathcal{B}, b^*)$
- 10: $\delta_t \leftarrow \delta_{\min} + (\delta_0 - \delta_{\min}) \frac{1 + \cos(\pi t/T_s)}{2}$
- 11: $\hat{c}_m \leftarrow \text{INFER}(\{x_s\}, \mathcal{S}, \delta_t; J = J_{\text{merge}}, n_{\max}^m)$
- 12: **if** $\hat{c}_m \neq \emptyset$ **then**
- 13: $\hat{y}_s \leftarrow \hat{c}_m \forall s \in \mathcal{I}$ $\triangleright$ merge
- 14: **else**
- 15: $\text{NEWCLUSTER}(\mathcal{C}, \mathcal{S}, \mathcal{I})$
- 16: $\hat{y}_s \leftarrow c_{\text{new}} \forall s \in \mathcal{I}$
- 17: **end if**
- 18: **end if**
- 19: **end if**
- 20: **end for**
- 21: **Convergence:** $\forall b \in \mathcal{B}$, POP, INFER with δ_{conv} and $J = J_{\text{merge}}$, merge or NEWCLUSTER
- 22: **return** $\hat{\mathbf{y}}$

D.5 Component Posterior from Equation 6

Applying Equation 6 to the vMF conjugate case:

$$\begin{aligned} \nu_k(d\phi) &\propto \exp(\kappa_0 \mu_0^\top \phi) \prod_{j=1}^n \exp(\rho_j(k) \kappa \phi^\top \mathbf{x}_j) \\ &= \exp\left(\phi^\top \left[\kappa_0 \mu_0 + \kappa \sum_{j=1}^n \rho_j(k) \mathbf{x}_j \right]\right) \\ &= \exp(\phi^\top \mathbf{r}_k) \end{aligned}$$

This is a vMF with natural parameter $\mathbf{r}_k$:

$$\begin{aligned} \nu_k &= \text{vMF}\left(\phi_k \mid \frac{\mathbf{r}_k}{\|\mathbf{r}_k\|}, \|\mathbf{r}_k\|\right), \\ \mathbf{r}_k &= \kappa_0 \mu_0 + \kappa \sum_{j=1}^n \rho_j(k) \mathbf{x}_j \end{aligned}$$

In the collapsed representation, ν_k is never explicitly constructed. We maintain only the sufficient statistic $\mathbf{r}_k$, which simultaneously parameterises ν_k and enters the collapsed predictive. Updating $\mathbf{r}_k$ after each change in ρ_i automatically updates ν_k .

D.6 Sequential Parameter Updates

In the sequential (one-pass) setting, observation $\mathbf{x}_i$ arrives fresh and has never contributed to the cluster statistics. Equation 6 applies directly. Substituting Equations 7 and 8, and identifying the

weight $w_k^{(i-1)} = n_k^{(i-1)}$:

$$\rho_i(k) \propto \begin{cases} n_k^{(i-1)} \frac{C_d(\|\mathbf{r}_k^{(i-1)}\|)}{C_d(\|\mathbf{r}_k^{(i-1)} + \kappa \mathbf{x}_i\|)} & k \leq K \\ \frac{\alpha C_d(\kappa_0)}{C_d(\|\kappa_0 \boldsymbol{\mu}_0 + \kappa \mathbf{x}_i\|)} & k = K+1 \end{cases}$$

Taking logarithms and dropping the shared $\log C_d(\kappa)$ (which cancels in normalisation):

$$\begin{aligned} s_k &= \log n_k^{(i-1)} + \log C_d(\|\mathbf{r}_k^{(i-1)}\|) \\ &\quad - \log C_d(\|\mathbf{r}_k^{(i-1)} + \kappa \mathbf{x}_i\|) \quad (k \leq K) \\ s_{K+1} &= \log \alpha + \log C_d(\kappa_0) \\ &\quad - \log C_d(\|\kappa_0 \boldsymbol{\mu}_0 + \kappa \mathbf{x}_i\|) \quad (k = K+1) \end{aligned}$$

with $\rho_i(k) = \exp(s_k) / \sum_l \exp(s_l)$. After computing ρ_i , the statistics are updated:

$$\begin{aligned} \mathbf{r}_k^{(i)} &= \mathbf{r}_k^{(i-1)} + \rho_i(k) \kappa \mathbf{x}_i \\ n_k^{(i)} &= n_k^{(i-1)} + \rho_i(k) \end{aligned}$$

If $\rho_i(K+1) > \epsilon$, a new cluster $K+1$ is created with $\mathbf{r}_{K+1}^{(i)} = \kappa_0 \boldsymbol{\mu}_0 + \rho_i(K+1) \kappa \mathbf{x}_i$ and $n_{K+1}^{(i)} = \rho_i(K+1)$.

In the sequential setting, $\mathbf{x}_i$ has never contributed to $\mathbf{r}_k^{(i-1)}$ or $n_k^{(i-1)}$, so the leave-one-out condition is satisfied automatically therefore no explicit subtraction is needed.

D.7 Convergence via subtract-and-rescore

After the sequential pass, we refine assignments by iterating. The challenge: $\mathbf{x}_i$ has already contributed to the statistics, so Lin’s Eq. 15 (which assumes a fresh observation) cannot be applied directly. We resolve this using exchangeability.

Step 1: Subtract $\mathbf{x}_i$ ’s contribution. Define leave-one-out statistics:

$$\begin{aligned} \mathbf{r}_{-i,k} &= \mathbf{r}_k^{\text{full}} - \rho_i^{\text{old}}(k) \kappa \mathbf{x}_i, \\ n_{-i,k} &= n_k^{\text{full}} - \rho_i^{\text{old}}(k) \end{aligned}$$

After this subtraction, $\mathbf{x}_i$ has zero contribution to the statistics.

Step 2: Invoke exchangeability. The DPMM is exchangeable: the conditional $p(z_i = k \mid \mathbf{z}_{-i})$ and the predictive $p(\mathbf{x}_i \mid \mathbf{x}_{-i,k})$ depend only on which observations belong to each cluster, not on the order they were processed. Therefore, the state after subtracting $\mathbf{x}_i$ ’s contribution is probabilistically equivalent to the state where $\mathbf{x}_i$ is genuinely the next observation to arrive.

Parameter	Description	Value
<i>Model priors</i>		
α	DP concentration	1.0
κ	Likelihood vMF concentration	$[800, 1500]^\dagger$
κ_0	Prior vMF concentration	0
$\boldsymbol{\mu}_0$	Prior mean direction	—
<i>Inference</i>		
ε_r	Responsibility update tolerance	10^{-5}

Table 12: Hyperparameters of vMF-DPMM. $^\dagger \kappa$ is tuned per task with the lower and upper thresholds provided.

Step 3: Apply Equation 6. With $\mathbf{r}_{-i,k}$ and $n_{-i,k}$ playing the role of the current statistics:

$$\begin{aligned} s_{i,k} &= \log n_{-i,k} + \log C_d(\|\mathbf{r}_{-i,k}\|) \\ &\quad - \log C_d(\|\mathbf{r}_{-i,k} + \kappa \mathbf{x}_i\|) \quad (k \leq K) \\ s_{i,K+1} &= \log \alpha + \log C_d(\kappa_0) \\ &\quad - \log C_d(\|\kappa_0 \boldsymbol{\mu}_0 + \kappa \mathbf{x}_i\|) \end{aligned}$$

with $\rho_i^{\text{new}}(k) = \exp(s_{i,k}) / \sum_l \exp(s_{i,l})$.

Step 4: Restore $\mathbf{x}_i$ ’s contribution.

$$\begin{aligned} \mathbf{r}_k^{\text{full}} &\leftarrow \mathbf{r}_{-i,k} + \rho_i^{\text{new}}(k) \kappa \mathbf{x}_i, \\ n_k^{\text{full}} &\leftarrow n_{-i,k} + \rho_i^{\text{new}}(k) \end{aligned}$$

The subtract-and-rescore procedure is operationally equivalent to collapsed Gibbs sampling (Teh et al., 2006a), where z_i is resampled from $p(z_i \mid \mathbf{z}_{-i}, \mathbf{x}_{1:n})$ after removing i from the counts. The only difference is that collapsed Gibbs samples a hard assignment while SVA takes the soft expectation under q .

E Experimental details for Section 7

Validation set To determine all hyperparameters for DPMM, wGFN and CoT across datasets, we sample across 10 data points across 5 classes from the validation set of each task (MultiNERD, CLINC-150 and GoEmotion). Therefore, we have 50 points in total per task to determine all valid hyperparameters.

Hyperparameters Table 12 lists the hyperparameters of the vMF-DPMM model and inference procedure used in Section 7. We point-estimate κ as opposed to maintaining a posterior as a full Bayesian treatment is intractable in closed form (Batmanghelich et al., 2016) and ablate sensitivity in Table 12. For wGFN, we follow similar hyperparameters as referenced in Table 10. For CoT, we use the recommended Qwen-3 hyperparameters for sampling (we set: temperature=0.6, top-p=0.95, top-k=20).

You are an assistant who will reply to the question clearly and succinctly. Finish each sentence with a full stop. In the sentence 'R with regular season drivers Oliver Gavin and Tommy Milner by the smallest ever margin of victory 0.034 seconds.' the entities ['Oliver Gavin', 'Tommy Milner'] represent Class A.

In the sentence 'All songs composed by Robbie Robertson.' the entity ['Robbie Robertson'] represents Class A.

In the sentence 'In April 2020, he conceded the nomination to Joe Biden, who had won a series of decisive victories as the field narrowed.' the entity ['Joe Biden'] represents Class A.

In the sentence 'At the far end of the hall are two large murals by ornithologist and artist, Louis Agassiz Fuertes.' the entity ['Louis Agassiz Fuertes'] represents Class A.

In the sentence 'The rally included video messages from Lou Brock, Ernie Banks, Ozzie Smith, Joe Montana, Wayne Gretzky and Michael Jordan.' the entities ['Lou Brock', 'Ernie Banks', 'Ozzie Smith', 'Joe Montana', 'Wayne Gretzky', 'Michael Jordan'] represent Class A.

In the sentence 'Another well-known early locomotive was "Puffing Billy", built 1813–14 by engineer William Hedley.' the entity ['Puffing Billy'] represents Class O.

In the sentence 'These were used for only a short time before being replaced by Yak-11 s and Yak-18 s in late 1952.' the entities ['Yak-11', 'Yak-18'] represent Class O.

In the sentence 'A special higher type of catering truck has been designed to accommodate the Airbus A380.' the entity ['Airbus A380'] represents Class O.

In the sentence 'The Imperial Japanese Army developed the Kayaba Ka-1 autogyro for reconnaissance, artillery-spotting, and anti-submarine uses.' the entity ['Kayaba Ka-1'] represents Class O.

In the sentence 'The Alvis Saladin armoured car was a 6x6 design with the engine compartment in the rear, a 76.2 mm low pressure gun turret in the centre and the driver in front.' the entity ['Alvis Saladin'] represents Class O.

In the sentence 'He also set a large number of "dietary interdictions", some based on calendrical notions (no water chestnuts in the 7th month), others on purported interactions between foods (no clear wine with horse meat) or between different flavors.' the entity ['water chestnuts'] represents Class M.

In the sentence 'It is the 16th-largest family of flowering plants, with more than 3,700 species in 434 genera including such well-known and economically important plants such as ajwain, angelica, anise, asafoetida, caraway, carrot, celery, chervil, coriander, cumin, dill, fennel, lovage, cow parsley, parsley, parsnip and sea holly, as well as silphium, a plant whose identity is unclear and which may be extinct.' the entities ['angelica', 'anise', 'caraway', 'carrot', 'celery', 'chervil', 'coriander', 'cumin', 'dill', 'lovage', 'parsley', 'parsnip'] represent Class M.

In the sentence 'Tea bags are made from abacá fibers, a very thin and long fiber manilla hemp.' the entity ['abacá fibers'] represents Class M.

In the sentence 'Ammophila Arenaria, known as European beachgrass, has a similar story, though it has no horticulture benefits.' the entity ['Ammophila Arenaria'] represents Class M.

In the sentence 'It consists of pureed or mashed potatoes and other root vegetables, and can also include cream, bacon, onion or shallot, herbs, and spices.' the entities ['onion', 'shallot'] represent Class M. Which class does the entity ['Alcona County'] in the sentence 'The highway turns northwesterly through forest land to cross into Alcona County.' represent? Think step by step and reply with the correct Class. Reply with 'None of these' if the answer does not fall in any Class.

Okay, let's see. The user is asking about the class of the entity "Alcona County" in the sentence "The highway turns northwesterly through forest land to cross into Alcona County." I need to figure out which class it belongs to based on the previous examples.

Looking at the examples, there are different classes: Class A, O, and M. Class A seems to be for people or entities that are individuals or well-known figures. Class O is for objects or things, like vehicles, machines, or specific items. Class M is for materials or substances, like fibers, plants, or ingredients.

In the given sentence, "Alcona County" is a place name. It's a county in the United States. Now, looking at the examples, the only class that fits a place like a county would be Class O. Wait, but in the earlier examples, places like "Airbus A380" were Class O, which is an object. But "Alcona County" is a geographical area, so maybe it's a place. However, in the examples given, there's no mention of a class for places. The classes are A, O, M.

Wait, looking back, in the example where "Alvis Saladin" was a vehicle, it was Class O. "Airbus A380" was an aircraft, also Class O. "Kayaba Ka-1" was a machine, Class O. So O is for objects, things, or entities that are specific items or machines. "Alcona County" is a geographical area, not an object. But the user hasn't provided an example for a county. Looking at the examples, the only class that wasn't A or M was O. The user didn't have any examples of a county, but the other examples for O were objects, machines, or vehicles. Since "Alcona County" is a place, but none of the examples have a place in the classes, maybe the answer is that it doesn't fit into any of the given classes. However, the user says to reply with 'None of these' if it doesn't fit into any.

In the example where "Ammophila Arenaria" was a plant, it was Class M. "Alvis Saladin" was a vehicle, Class O. "Airbus A380" was a plane, Class O. "Kayaba Ka-1" was a machine, Class O. "Puffing Billy" was a locomotive, Class O. "water chestnuts" were a plant, Class M. "abacá fibers" were a material, Class M. "angelica" etc. were plants, Class M. "Alcona County" is a place, not a person, object, or material. Therefore, it doesn't fit into A, O, or M. So the answer should be "None of these".

None of these.

Figure 11: Multi-step CoT reasoning with Qwen-3-4B (enabled deep thinking mode represented in orange, input prompt displayed in red and model response in green) using NERD. Notice that the model employs deeper chain of thought reasoning traces after decomposing the problem step-by-step (Wei et al., 2022b).

Embedding dimension Qwen-3 embedding models are trained with Matryoshka Representation Learning (MRL) (Kusupati et al., 2022), which nests representations at multiple granularities within a single model so that any leading prefix of the full embedding vector retains strong representational quality. This means that truncating embeddings to a lower dimension is a native, lossless operation rather than a post-hoc approximation. In contrast, methods such as UMAP (McInnes et al., 2020) or the Nyström approximation apply non-linear or random projections that can distort the hyperspherical geometry on which the vMF-DPMM is defined, introducing artefacts that would corrupt the model’s distance computations. We therefore use $d = 512$, a dimension natively supported by MRL training, as a computationally efficient operating point that avoids any such geometric distortion after doing a sweep on our validation set from $d = 32$ to $d = 2048$ with a step size of $\times 2$.

Deep thinking mode in Qwen-3 All autoregressive models in the Qwen-3 family have an option to enable significant enhancement in reasoning, coding and agentic abilities by enabling the *thinking* mode (Yang et al., 2025), albeit at a significantly higher cost due to an increase in the number of “thought” tokens consumed. We compare wGFN and DPMM against this state-of-the-art baseline for few-shot online clustering and present a sample CoT reasoning trace in Figure 11.

F Usage of AI

ChatGPT was used for grammatical correction and compressing original content in the main paper, mainly to fit the page limit. Boilerplate code snippets (such as helper functions, I/O operations, sampling functions etc.) were also written with the help of coding assistants.