How Extractable Are Edge LLMs?

Anonymous authors
Paper under double-blind review

Abstract

Edge large language models (Edge LLMs) are commonly quantized for deployment on devices with limited memory and compute. At the same time, their text interfaces allow an attacker to collect responses and train a student model, creating a risk of black-box behavioral extraction. For such attacks, collecting more target responses provides additional supervision; however, slow on-device inference makes large-scale data collection costly. We therefore investigate whether an attacker can extract instruction-following behavior more effectively under a limited target-query budget. To assess this risk, we introduce Coordinated Linguistic Instruction Querying (**CLIQ**), a structured probing framework designed to reduce repeated probing of similar behaviors by coordinating queries across semantic regions. Specifically, CLIQ partitions a public instruction pool and generates region-conditioned queries before observing any target response. We compare it with Original Queries (**OQ**), which are randomly sampled from the same pool and used without modification. Under a nominal 500-query budget, CLIQ improves a 4B student's BERT-F1 from **77.92** with OQ to **82.44** on Alpaca evaluation references. Furthermore, experiments with two phone-deployed Gemma teachers show improved student performance over OQ while target-side generation speeds remain comparable. Taken together, these results demonstrate that functional instruction-following behavior remains transferable from quantized edge models under the evaluated query budgets: the cost of collecting target responses **does not, by itself, establish resistance to behavioral extraction**.

1 Introduction

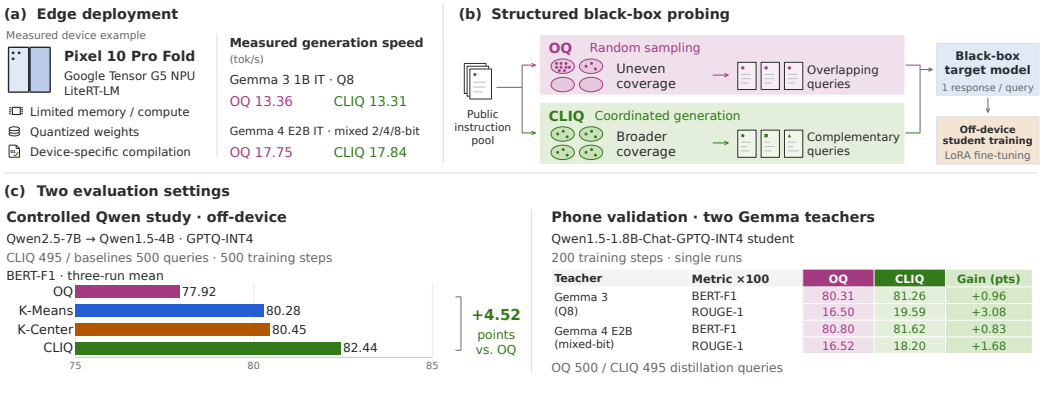


Figure 1: Overview. (a) Edge deployment makes target-response collection costly, motivating the study of behavioral extraction under limited query budgets. (b) Unlike random OQ sampling, CLIQ coordinates region-conditioned probes to improve semantic coverage and reduce redundancy; collected responses supervise off-device student training. (c) Controlled experiments compare query strategies under a shared nominal budget, while phone validation tests transfer from deployed models. Improved transfer at comparable phone generation speeds shows that costly target access does not, by itself, establish extraction resistance. Scores use original Alpaca references.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Edge large language models (Edge LLMs) are becoming important components of mobile applications, enabled by model compression and efficient local inference (Zheng et al., 2025). In particular, Apple's Foundation Models framework and Google's ML Kit GenAI Prompt API give developers access to on-device models (Apple, 2025; Google, 2026b), making their capabilities reusable assets for application development. Reflecting the importance of controlling this reuse, Apple prohibits training other models on Apple-model outputs, while Google's ML Kit GenAI terms restrict competitive model development and model extraction (Apple, 2026; Google, 2025).

These deployments also expose a potential route for black-box behavioral extraction: an attacker may collect textual responses and use them to train an unauthorized student through knowledge distillation (Hinton et al., 2015). Google and Anthropic have reported related extraction and distillation campaigns against cloud services (Google Threat Intelligence Group, 2026; Anthropic, 2026). However, assessing this risk on edge devices requires accounting for local resource constraints. Although quantization reduces memory requirements (Dettmers et al., 2023; Nagel et al., 2021; Gholami et al., 2021), collecting additional supervision still requires target inference on devices with limited compute, making large-scale response collection costly. Google's AICore additionally imposes per-application quotas and foreground-use restrictions (Google, 2026a). We therefore ask: *can an attacker extract instruction-following behavior more effectively under a limited target-query budget?*

To assess this risk, we introduce Coordinated Linguistic Instruction Querying (**CLIQ**). Our intuition is to coordinate *where queries are allocated* and *what they ask* to reduce redundant probing. As illustrated in Figure 1, CLIQ partitions public instructions into semantic regions, filters sparsely supported source clusters, and uses representative textual examples to guide an attacker-side generator. Each retained region contributes an equal number of new probes, constructed before any target response is observed. We compare CLIQ with Original Queries (**OQ**), randomly sampled and used without modification, and with diversity-aware selection methods.

In the main comparison using 495 CLIQ queries and 500 queries for each baseline, CLIQ improves a 4B student's reference-based BERT-F1 from **77.92** with OQ to **82.44**, exceeding K-Center Greedy, the strongest evaluated selection-only baseline, by 1.99 points. Furthermore, phone validation with two Gemma teachers and 0.5B/1.8B quantized Qwen students shows gains across all six reported metrics under a common 200-step protocol, while target generation speeds remain comparable across query strategies. Scores use original references for 5,000 Alpaca evaluation examples, measuring functional transfer rather than exact target imitation. These open-model experiments do not test the commercial interfaces discussed above. Within the evaluated settings, they show that **target-response collection costs do not, by themselves, establish resistance to behavioral extraction**.

Contributions. This paper makes the following contributions:

(i) **Real-device evidence of edge-model behavioral extraction.** Using two Gemma teachers deployed through LiteRT-LM on a Pixel 10 Pro Fold's Tensor G5 NPU, we demonstrate functional behavior transfer to quantized students under limited target access, grounding the extraction risk in actual edge deployment.

(ii) **Structured probing under limited target access.** We introduce CLIQ, which coordinates both *where to query* and *what to query* by organizing public instructions into semantic regions and generating region-conditioned probes before observing target responses.

(iii) **Controlled evaluation across model configurations.** With 495 CLIQ queries versus 500 for the baselines and matched training budgets, CLIQ outperforms OQ and diversity-aware selection, including a 1.99-point BERT-F1 gain over K-Center Greedy. Additional student sizes, model families, and quantization settings test the robustness of this finding.

2 Related Work

2.1 Knowledge Distillation and Black-box Extraction

Knowledge distillation transfers behavior from a larger teacher to a smaller student, with early work addressing classification (Hinton et al., 2015) and sequence modeling (Kim & Rush, 2016), and later studies extending the approach to instruction and response generation (Honovich et al., 2023; Gu et al., 2024). Data-free and API-based variants obtain supervision from black-box model

outputs without access to the original training data (Zhang et al., 2023; Chen et al., 2024). From a security perspective, the same mechanism can transfer useful instruction-following behavior to an unauthorized student. Our work focuses on how much functional behavior is recoverable under a limited target-query budget.

2.2 Query-efficient Distillation and Extraction

Data-efficient learning spans active distillation for recommendation (Du et al., 2025), core-set active learning (Sener & Savarese, 2018), and synthetic instruction generation (Honovich et al., 2023; Wang et al., 2023). Shu et al. (2026) study representation redundancy in instruction-data selection. We compare CLIQ with K-Means Representative and K-Center Greedy to test whether CLIQ improves on diversity-aware selection under a shared nominal query budget and matched training budgets.

Work on model extraction has also developed defenses based on query-distribution monitoring, response perturbation, and watermarking (Juuti et al., 2019; Orekondy et al., 2020; Zhao et al., 2022). Our results motivate considering query organization alongside query count when assessing behavioral exposure.

2.3 Edge-oriented and Quantized LLMs

Post-training quantization methods such as GPTQ and AWQ reduce model memory requirements (Frantar et al., 2022; Lin et al., 2025), while QLoRA enables memory-efficient adaptation of quantized models (Dettmers et al., 2023). While these techniques enable local deployment, our focus is behavioral extraction under limited target access. We evaluate quantized, edge-oriented checkpoints across model families and quantization settings, with phone inference measurements grounding the cost of collecting target responses.

3 Method

Given public instructions $\mathcal{Q}_0$ and an attacker-side generator G, CLIQ constructs probes for extracting behavior from a quantized target T. Figure 2 separates pre-access query construction (Steps 1–4) from shared response collection and student training (Steps 5–6).

3.1 Threat Model and Extraction Objective

We consider text-only black-box access. The attacker can submit instructions and observe textual responses, but cannot access target parameters, gradients, logits, hidden representations, training data, or privileged device instrumentation. Public instructions and an independent query generator are available for constructing probes.

The nominal budget B_{nominal} bounds target queries for training supervision, reflecting limited compute and slow on-device inference (measured in Section 4.3). Attacker-side query generation is accounted for separately.

The objective is to transfer instruction-following behavior to an unauthorized student S_ϕ. We evaluate functional transfer against original Alpaca references (Section 4.1).

3.2 CLIQ Query Construction

OQ randomly samples unmodified instructions from the same public pool that CLIQ partitions to construct region-conditioned probes.

Semantic partitioning. Let $\mathcal{Q}_0$ contain N instructions. We encode each $q_i \in \mathcal{Q}_0$ with a sentence encoder E and partition the embeddings into K semantic regions $\{\mathcal{Q}_1, \ldots, \mathcal{Q}_K\}$. Writing $n_k = |\mathcal{Q}_k|$, uniform random sampling from the public pool gives OQ the expected regional allocation

$$\mathbb{E}\left[b_k^{\text{OQ}}\right] = B_{\text{nominal}} \frac{n_k}{N}, \tag{1}$$

where b_k^{OQ} is the number of sampled queries from region k. Random querying therefore favors frequent regions, whose prevalence in the pool need not reflect their value for extraction.

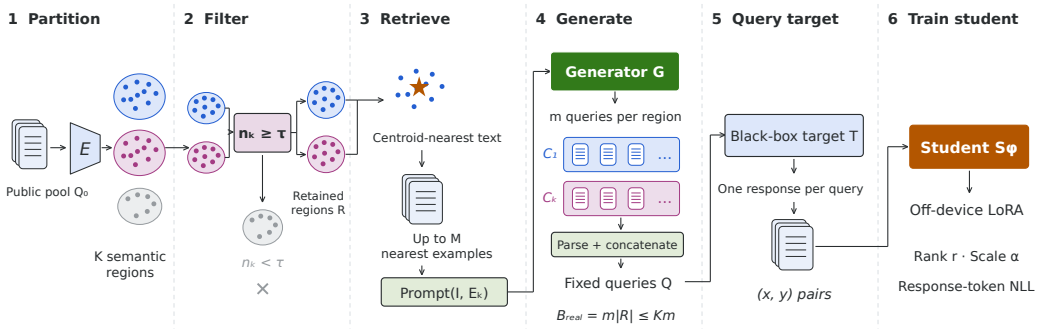


Figure 2: CLIQ query construction and extraction. Steps 1–4 construct region-conditioned probes; Steps 5–6 collect target responses and train the student.

Source-region filtering and budget allocation. CLIQ assigns m queries to each retained region. To avoid overemphasizing narrowly supported regions, we filter source clusters immediately after partitioning, retaining

$$\mathcal{R} = \{k \in \{1, \dots, K\} : n_k \geq \tau\}, \tag{2}$$

where τ is the minimum number of original instructions required in a source cluster.

This trades tail-region coverage for minimum source support before generation or target access; retained regions receive equal allocations.

Representative textual context. For each retained region $\mathcal{Q}_k$, we compute its embedding centroid $c_k = n_k^{-1} \sum_{q_i \in \mathcal{Q}_k} E(q_i)$ and retrieve up to M in-region instructions nearest to it by cosine similarity. These instructions form a textual example set $\mathcal{E}_k$.

The retrieved text conditions generation; neither centroids nor these examples are submitted to the target as probes.

Coordinated generation within regions. We combine $\mathcal{E}_k$ with a shared generation instruction $\mathcal{I}$ to form $P_k = \mathrm{Prompt}(\mathcal{I}, \mathcal{E}_k)$. Each request asks the attacker-side generator G to produce a block of m instruction–input pairs:

$$\mathcal{C}_k = (q_{k,1}, \dots, q_{k,m}) \sim p_G(\cdot \,|\mathcal{I}, \mathcal{E}_k), \qquad k \in \mathcal{R}. \tag{3}$$

Each block coordinates probes under a shared regional context.

Query assembly and budget accounting. We parse the generated instruction–input pairs and concatenate the regional blocks into the final query collection $\mathcal{Q}$. With K and m chosen such that $B_{\text{nominal}} = Km$, the realized query count is

$$B_{\text{real}} = |\mathcal{Q}| = m|\mathcal{R}| \leq Km = B_{\text{nominal}}. \tag{4}$$

Any shortfall from the nominal budget comes from excluding small source clusters before generation.

The complete query collection is fixed before any target response is observed and reused across student-training runs. Appendix A.1 provides the prompting and output-parsing details.

3.3 Target Response Collection and Student Training

For each query $x \in \mathcal{Q}$, we collect one textual response y_x from T, forming the extraction dataset $\mathcal{D}_{\mathcal{Q}} = \{(x, y_x) : x \in \mathcal{Q}\}$. We fine-tune the student off-device using LoRA (Hu et al., 2022) and token-level cross-entropy on the target-generated response tokens, conditioned on the query and preceding response tokens.

Within each setting, response collection, student initialization, and training settings are matched; collected datasets are fixed across seeds.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Experimental settings. The main off-device study uses Qwen2.5-7B-GPTQ-INT4 (Yang et al., 2025b) as the target and Qwen1.5-4B/7B-GPTQ-INT4 (Qwen Team, 2024) as students. Robustness experiments use Llama-3.1-8B-Instruct (Grattafiori et al., 2024) under GPTQ-INT4 or AWQ-INT4 (Lin et al., 2025) with a Qwen1.5-4B-Chat-GPTQ-INT4 student, and examine training sensitivity using Qwen3-1.7B (Yang et al., 2025a) under GPTQ-INT4/INT8. For phone validation, Gemma 3 1B IT (Q8) (Google DeepMind, 2025) and Gemma 4 E2B IT (mixed 2/4/8-bit) run on the Pixel 10 Pro Fold's Tensor G5 NPU through precompiled LiteRT-LM packages (Google AI Edge, 2026). Each supervises Qwen1.5-0.5B/1.8B-Chat-GPTQ-INT4 students trained off-device.

Data and evaluation. OQ and CLIQ share a pool of 52,001 Alpaca instructions (Taori et al., 2023). We evaluate deterministic student outputs on 5,000 fixed instructions (seed 42) against original Alpaca references using BERT-F1 (Zhang et al., 2020), BLEU (Papineni et al., 2002), and ROUGE-1/2/L/Lsum (Lin, 2004). These metrics assess reference-based instruction-following transfer, not direct student–target output matching.

Query strategies and budgets. The controlled 4B comparison includes **Original Queries (OQ)**, which randomly samples eligible instructions without modification; **K-Means Representative**, which selects centroid-nearest instructions; **K-Center Greedy** (Sener & Savarese, 2018), which iteratively maximizes the minimum embedding distance to selected instructions; and **CLIQ**, which generates region-conditioned queries. Other comparisons use OQ and CLIQ.

CLIQ partitions the pool into $K = 100$ clusters and filters those with fewer than $\tau = 5$ original instructions. Qwen3-30B-A3B-Instruct-2507 (Qwen Team, 2025) generates $m = 5$ probes per retained region. Both main and phone collections retain 99 regions, yielding 495 CLIQ queries under the nominal budget $Km = 500$; OQ and the selection-only baselines use 500. Attacker-side generation and retries are separate.

Student training and reporting. One response per query forms a fixed training set across runs. Main experiments use 500 steps with LoRA $r = 64, \alpha = 16$; all four phone configurations use 200 steps with $r = 4, \alpha = 1$. Both use $\alpha/r = 0.25$; full settings appear in Supplementary Section A.1.3. Controlled results report mean $\pm$ standard deviation over seeds $\{42, 43, 44\}$ unless specified otherwise. Qwen3-1.7B training-step results report mean $\pm$ standard error over three runs; phone results are single-run estimates, and query-budget curves are point estimates.

4.2 CONTROLLED EXTRACTION UNDER LIMITED QUERY BUDGETS

Deterministic evaluation gives Qwen2.5-7B 83.69 BERT-F1 in FP16 and 84.22 in GPTQ-INT4 on the same Alpaca references. The latter provides the target-quality reference.

Table 1: Extraction from Qwen2.5-7B-GPTQ-INT4 using 500 OQ / 495 CLIQ queries and 500 training steps. Scores use Alpaca evaluation references (%). Student rows report mean $\pm$ SD over three runs; teacher and base rows are point estimates.

Model	Role	Query	BERT-F1	BLEU	R-1	R-2	R-L
Qwen2.5-7B FP16	Teacher	–	83.69	2.94	22.08	9.36	16.85
Qwen2.5-7B INT4	Teacher	–	84.22	3.15	21.92	9.58	16.56
Qwen1.5-4B INT4	Base	–	75.04	0.83	14.07	5.08	9.46
Qwen1.5-4B INT4	Student	OQ	77.92 ± 0.09	1.21 ± 0.17	16.53 ± 0.20	5.71 ± 0.08	10.87 ± 0.11
Qwen1.5-4B INT4	Student	CLIQ	$\mathbf{82.44 \pm 0.07}$	$\mathbf{2.56 \pm 0.05}$	$\mathbf{18.74 \pm 0.20}$	$\mathbf{6.40 \pm 0.08}$	$\mathbf{12.42 \pm 0.11}$
Qwen1.5-7B INT4	Base	–	78.02	1.04	17.03	5.51	10.87
Qwen1.5-7B INT4	Student	OQ	79.72 ± 0.10	1.48 ± 0.14	17.88 ± 0.20	6.04 ± 0.08	11.77 ± 0.12
Qwen1.5-7B INT4	Student	CLIQ	$\mathbf{83.04 \pm 0.07}$	$\mathbf{2.72 \pm 0.05}$	$\mathbf{19.01 \pm 0.20}$	$\mathbf{6.38 \pm 0.08}$	$\mathbf{12.55 \pm 0.12}$

Table 1 shows that CLIQ raises BERT-F1 from 77.92 to 82.44 for the 4B student and from 79.72 to 83.04 for the 7B student, with consistent BLEU and ROUGE gains.

Selection-only baselines. Table 2 compares all four strategies for the 4B student.

Table 2: 4B-student query strategies under a nominal 500-query budget and at 500 training steps. BERT-F1 is mean ± SD over three runs; Δ is the gain over OQ in points.

Strategy	Query source	Target queries	BERT-F1	Δ vs. OQ
OQ	Existing pool	500	77.92 ± 0.09	–
K-Means Representative	Existing pool	500	80.28 ± 0.07	+2.36
K-Center Greedy	Existing pool	500	80.45 ± 0.08	+2.53
CLIQ	Generated	495	$\mathbf{82.44 \pm 0.07}$	**+4.52**

Both selection baselines improve on OQ, but CLIQ remains 2.16 points above K-Means and 1.99 above K-Center.

To contextualize these scores relative to student initialization, we compute

$$\text{GapClosure} = \frac{\text{Score}_{\text{extracted}} - \text{Score}_{\text{base}}}{\text{Score}_{\text{teacher}} - \text{Score}_{\text{base}}}. \tag{5}$$

For 4B, OQ, K-Means, K-Center, and CLIQ close 31.4%, 57.1%, 58.9%, and 80.6% of the base-to-teacher gap, respectively; for 7B, OQ and CLIQ close 27.4% and 81.0%. This normalizes reference-based BERT-F1 rather than measuring direct target agreement.

4.3 Real-Device Evidence

Each Gemma teacher provides 500 OQ and 495 CLIQ responses. Latency and generation speed are arithmetic means over successful Tensor G5 NPU records.

OQ and CLIQ have similar latency, and generation rates differ by less than 1%. CLIQ improves all six metrics across teachers and student sizes at 200 steps (Table 3). Comparable token rates do not imply equal total collection time.

Shorter phone responses motivate lighter adaptation: across 993 paired non-empty outputs, Qwen1.5-tokenized mean/median lengths are 409.4/512 for Qwen, 58.4/63 for Gemma 3, and 60.7/65 for Gemma 4. Qwen generation is capped at 512 tokens (Supplementary Section A.1.4).

Beyond the main 200-step phone comparison, Gemma 4's 1.8B student improves in CLIQ BERT-F1 through 500 steps, but ROUGE peaks at 200 steps and ROUGE-Lsum falls below OQ at 300–500 steps (Supplementary Section A.2.2).

4.4 Robustness beyond the Main Configuration

We further test off-device Llama-to-Qwen transfer under a nominal budget of 500 target queries and at 500 training steps.

CLIQ improves BERT-F1 in both settings (Table 4), with gains exceeding the observed training SDs. For AWQ, ROUGE-L also increases from 8.38 ± 0.09 to 16.06 ± 0.29. Each comparison fixes the target configuration and varies the query strategy.

4.5 Query Budget and Student-Training Sensitivity

Target-query budget. Figure 3 reports point estimates at 100–400 queries. Smaller CLIQ budgets allocate queries equally across retained regions whenever possible, using the fixed region and within-region ordering of the 495-query collection.

CLIQ improves between 100 and 300 queries and then changes little, while OQ remains nearly flat. These points characterize the tested range, not an exact reduction in required queries.

Training steps and student quantization. With the query collection fixed, Table 5 compares Qwen3-1.7B at 100, 300, and 500 steps under INT4 and INT8.

Table 3: Phone validation on the Pixel 10 Pro Fold (Tensor G5 NPU, LiteRT-LM). Student scores are single-run results at 200 steps, scaled by 100; bold marks the better strategy for each student.

(a) Gemma 3 1B IT (Q8)

Device inference	**OQ**	**CLIQ**
Target queries	500	495
First-token latency (ms)	206.34	207.95
Generation speed (tok/s)	13.36	13.31

Student	**0.5B**		**1.8B**	
Metric	OQ	CLIQ	OQ	CLIQ
BERT-F1	78.87	**80.51**	80.31	**81.26**
BLEU	0.99	**1.39**	1.36	**2.05**
ROUGE-1	14.62	**16.67**	16.50	**19.59**
ROUGE-2	4.27	**4.90**	5.10	**6.70**
ROUGE-L	9.95	**11.25**	11.18	**13.59**
ROUGE-Lsum	11.45	**12.14**	13.13	**15.06**

(b) Gemma 4 E2B IT (mixed 2/4/8-bit)

Device inference	**OQ**	**CLIQ**
Target queries	500	495
First-token latency (ms)	125.00	126.31
Generation speed (tok/s)	17.75	17.84

Student	**0.5B**		**1.8B**	
Metric	OQ	CLIQ	OQ	CLIQ
BERT-F1	75.35	**76.21**	80.80	**81.62**
BLEU	0.59	**0.66**	1.39	**2.00**
ROUGE-1	11.91	**12.79**	16.52	**18.20**
ROUGE-2	3.72	**3.94**	5.05	**5.77**
ROUGE-L	8.09	**8.71**	11.02	**12.42**
ROUGE-Lsum	9.35	**10.02**	13.05	**13.86**

Table 4: Off-device robustness under a nominal 500-query budget and at 500 training steps. BERT-F1 is mean ± SD over three runs; gains compare CLIQ with OQ within each setting.

Setting	**OQ**	**CLIQ**	**Gain**
Llama-GPTQ → Qwen-GPTQ	80.46 ± 0.17	$\mathbf{82.82 \pm 0.18}$	+2.36
Llama-AWQ → Qwen-GPTQ	82.19 ± 0.05	$\mathbf{85.13 \pm 0.13}$	+2.94

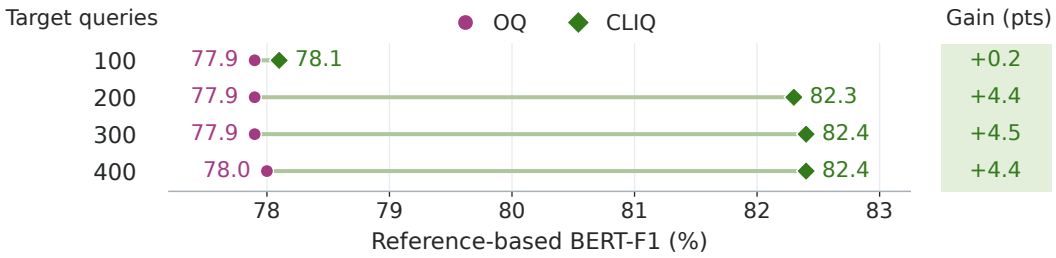


Figure 3: Reference-based BERT-F1 across target-query budgets. Segments connect OQ and CLIQ scores; gains are in points. Estimates have no uncertainty bounds.

Table 5: Qwen3-1.7B training dynamics under GPTQ-INT4 and GPTQ-INT8 under a nominal 500-query budget. Scores are mean ± standard error (SE) over three runs.

(a) GPTQ-INT4

Strategy	**Steps**	**BERT-F1**	**BLEU**	**R-L**
OQ	100	80.32 ± 0.19	1.55 ± 0.16	13.83 ± 0.33
	300	80.44 ± 0.20	1.49 ± 0.17	14.12 ± 0.33
	500	80.03 ± 0.21	1.42 ± 0.21	13.41 ± 0.32
CLIQ	100	78.20 ± 0.23	1.08 ± 0.36	12.45 ± 0.34
	300	82.86 ± 0.19	2.56 ± 0.13	14.22 ± 0.32
	500	$\mathbf{82.92 \pm 0.17}$	$\mathbf{2.55 \pm 0.11}$	$\mathbf{14.54 \pm 0.34}$

(b) GPTQ-INT8

Strategy	**Steps**	**BERT-F1**	**BLEU**	**R-L**
OQ	100	80.14 ± 0.19	1.50 ± 0.17	13.49 ± 0.32
	300	80.17 ± 0.20	1.41 ± 0.21	13.51 ± 0.32
	500	80.08 ± 0.19	1.43 ± 0.15	13.49 ± 0.32
CLIQ	100	78.18 ± 0.23	1.07 ± 0.36	12.53 ± 0.34
	300	82.70 ± 0.16	2.44 ± 0.13	14.48 ± 0.33
	500	$\mathbf{82.74 \pm 0.18}$	$\mathbf{2.47 \pm 0.09}$	$\mathbf{14.51 \pm 0.33}$

CLIQ starts below OQ at 100 steps but surpasses it at 300 and 500 steps, reaching final BERT-F1 gains of 2.89 and 2.66 points, respectively.

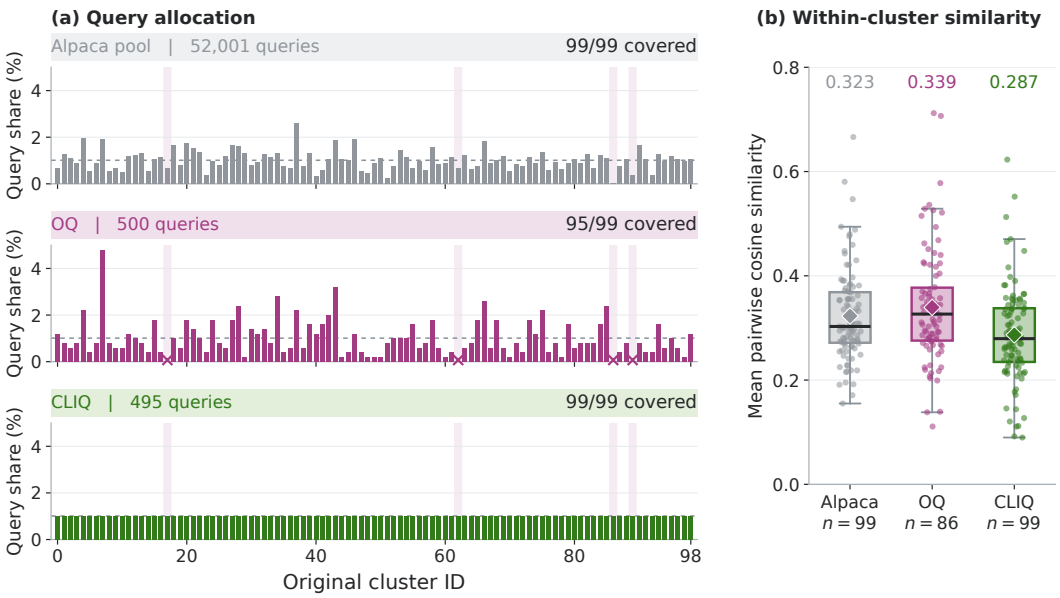


Figure 4: Alpaca pool, OQ, and CLIQ under the original 99-cluster partition. (a) Query shares; shading marks OQ's missing clusters and the dashed line marks equal allocation. (b) Within-cluster similarity; n counts clusters with at least two queries, and diamonds and numbers mark means.

5 Discussion

5.1 Coverage and Complementary Supervision

The query diagnostics suggest two explanations for CLIQ's gains under limited target access: broader regional allocation and greater within-region variety.

Across-region allocation. Figure 4 compares the 52,001-instruction Alpaca pool, 500 OQ queries, and 495 CLIQ queries under the saved 99-cluster partition. OQ uses original pool labels; CLIQ uses generation-source labels. OQ covers 95 clusters, missing four and reaching nine only once; CLIQ assigns five queries to every source region. Thus, regional allocation differs beyond coverage alone.

Within-region variety. Mean pairwise cosine similarity of normalized MiniLM embeddings, computed for clusters with at least two queries, is 0.323 for the pool, 0.339 for OQ, and 0.287 for CLIQ. Despite overlapping distributions, CLIQ's lower mean and median suggest less within-region repetition.

Query examples. The concept examples vary the explanatory focus within a topic; the haiku example adds sensory and contextual constraints to an open-ended task.

Embedding dissimilarity alone need not improve supervision. Semantic relevance combined with distinct response requirements may explain why CLIQ outperforms diversity-aware selection (Section 4.2). The diagnostics support this interpretation without isolating the contributions of allocation and generation. Supplementary Section A.3 provides further coverage and projection analyses.

5.2 Supervision Utilization during Training

The delayed CLIQ gains in Table 5 suggest that broader supervision may require more optimization. With response sets fixed, improvements use existing supervision rather than additional target access. Later plateaus indicate diminishing returns. Query efficiency therefore depends on both query construction and the student's ability to use the resulting responses. The trajectories complement the allocation and similarity diagnostics by showing when these benefits emerge.

Concept explanation

Pool examples
"Explain what an API is."
"Explain the concept of a bubble sort algorithm to a non-technical audience."

CLIQ examples
"Describe how a transformer architecture works in natural language processing, focusing on attention mechanisms."
"Explain the difference between supervised and unsupervised learning with one practical example for each."

Haiku writing

Pool examples
"Generate a haiku poem."
"Compose a haiku about cats."

CLIQ example
"Create a haiku using only sensory details from the natural world, focusing on sound and touch."

Input: Forest at twilight, wind rustling through dry leaves

6 Conclusion

Costly local inference does not establish resistance to behavioral extraction. CLIQ improves instruction-following transfer over random querying and, in the controlled 4B study, diversity-aware selection. Phone validation shows gains across two Gemma teachers and two student sizes at comparable generation speeds. Security assessments should therefore consider query organization alongside volume.

7 Limitations

Gemma validation covers one phone and its LiteRT-LM stack, with single-run, near-matched-budget comparisons and no selection-only baselines; Qwen and Llama experiments run off-device. The instruction-following experiments isolate neither quantization effects through matched full-precision controls nor the contributions of individual CLIQ components. These controls and purpose-built defenses remain future work.

8 Ethical Considerations

Behavioral extraction can enable unauthorized model imitation. Our controlled experiments use public data and licensed open-source models, without probing third-party services or bypassing access controls. Applications must respect licenses, service terms, and applicable regulations. To support risk assessment and defense research, planned releases cover code and configurations for licensed open-source models.

Use of AI Assistants

CLIQ uses Qwen3-30B-A3B-Instruct-2507 to generate region-conditioned queries. AI assistants (ChatGPT and Codex) were also used to polish the manuscript's wording and assist with formatting and layout adjustments. The authors take responsibility for the final manuscript.

References

Anthropic. Detecting and Preventing Distillation Attacks. Anthropic, February 2026. URL https://www.anthropic.com/news/detecting-and-preventing-distillation-attacks. Published February 23, 2026. Vendor-reported incidents and attribution. Accessed September 24, 2026.

Apple. Apple's Foundation Models Framework Unlocks New App Experiences Powered by Apple Intelligence. Apple Newsroom, September 2025. URL https://www.apple.com/newsroom/2025/09/apples-foundation-models-framework-unlocks-new-intelligent-app-experiences/. Published September 29, 2025. Accessed September 24, 2026.

Apple. Apple Developer Program License Agreement. Apple Developer, 2026. URL https://developer.apple.com/support/terms/apple-developer-program-license-agreement/. Section 3.2, item 8, clause 2. Online version accessed September 24, 2026.

Hongzhan Chen, Ruijun Chen, Yuqi Yi, Xiaojun Quan, Chenliang Li, Ming Yan, and Ji Zhang. Knowledge distillation of black-box large language models, 2024. URL https://arxiv.org/abs/2401.07013. arXiv:2401.07013.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 10088–10115. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1feb87871436031bdc0f2beaa62a049b-Paper-Conference.pdf.

Yingpeng Du, Zhu Sun, Ziyan Wang, Haoyan Chua, Jie Zhang, and Yew-Soon Ong. Active large language model-based knowledge distillation for session-based recommendation. In *Proceedings of the 39th Annual Conference on Artificial Intelligence (AAAI-25)*, pp. 11607–11615. AAAI Press, 2025. doi: 10.1609/AAAI.V39I11.33263. URL https://doi.org/10.1609/aaai.v39i11.33263.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: accurate post-training quantization for generative pre-trained transformers. *CoRR*, abs/2210.17323, 2022. doi: 10.48550/ARXIV.2210.17323. URL https://doi.org/10.48550/arXiv.2210.17323.

Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W. Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference. *CoRR*, abs/2103.13630, 2021. URL https://arxiv.org/abs/2103.13630.

Google. ML Kit GenAI API Additional Terms of Service. Google for Developers, 2025. URL https://developers.google.com/ml-kit/genai-terms. Use Restrictions. Terms marked last modified May 14, 2025. Accessed September 24, 2026.

Google. Overview of the ML Kit GenAI APIs. Google for Developers, 2026a. URL https://developers.google.com/ml-kit/genai. Sections on per-application quotas and background usage. Accessed September 26, 2026.

Google. Send Custom Requests to Gemini Nano with GenAI Prompt API. Google for Developers, 2026b. URL https://developers.google.com/ml-kit/genai/prompt/android. Beta API documentation. Page marked last updated July 15, 2026. Accessed September 24, 2026.

Google AI Edge. LiteRT-LM. GitHub repository, 2026. URL https://github.com/google-ai-edge/LiteRT-LM. Repository accessed September 26, 2026; year denotes the consulted version.

Google DeepMind. Gemma 3 1B IT model card. Hugging Face, 2025. URL https://huggingface.co/google/gemma-3-1b-it. Accessed September 26, 2026.

Google DeepMind. Gemma 4 E2B IT: Model configuration, 2026a. URL https://huggingface.co/google/gemma-4-E2B-it/raw/main/config.json.

Google DeepMind. Gemma 4 E2B IT model card, 2026b. URL https://huggingface.co/google/gemma-4-E2B-it.

Google Threat Intelligence Group. GTIG AI Threat Tracker: Distillation, Experimentation, and (Continued) Integration of AI for Adversarial Use. Google Cloud Blog, February 2026. URL https://cloud.google.com/blog/topics/threat-intelligence/distillation-experimentation-integration-ai-adversarial-use. Published February 12, 2026. Vendor-reported incidents. Accessed September 24, 2026.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=5h0qf7IBZZ.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015. URL http://arxiv.org/abs/1503.02531.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. Unnatural instructions: Tuning language models with (almost) no human labor. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 14409–14428. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.ACL-LONG.806. URL https://doi.org/10.18653/v1/2023.acl-long.806.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Mika Juuti, Sebastian Szyller, Samuel Marchal, and N Asokan. PRADA: protecting against DNN model stealing attacks. In *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 512–527. IEEE, 2019.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In Jian Su, Xavier Carreras, and Kevin Duh (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pp. 1317–1327. The Association for Computational Linguistics, 2016. doi: 10.18653/V1/D16-1139. URL https://doi.org/10.18653/v1/d16-1139.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Guangxuan Xiao, and Song Han. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. *GetMobile: Mobile Computing and Communications*, 28(4):12–17, 2025. doi: 10.1145/3714983.3714987. URL https://doi.org/10.1145/3714983.3714987.

LiteRT Community. Gemma 4 E2B IT for LiteRT-LM, 2026. URL https://huggingface.co/litert-community/gemma-4-E2B-it-litert-lm.

Leland McInnes and John Healy. UMAP: uniform manifold approximation and projection for dimension reduction. *CoRR*, abs/1802.03426, 2018. URL https://arxiv.org/abs/1802.03426v1.

Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart van Baalen, and Tijmen Blankevoort. A white paper on neural network quantization. *CoRR*, abs/2106.08295, 2021. URL https://arxiv.org/abs/2106.08295.

Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. Prediction poisoning: Towards defenses against DNN model stealing attacks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SyevYxHtDB.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pp. 311–318. ACL, 2002. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040/.

Qwen Team. Introducing Qwen1.5. Qwen Blog, February 2024. URL https://qwenlm.github.io/blog/qwen1.5/. Published February 4, 2024. Accessed September 26, 2026.

Qwen Team. Qwen3-30B-A3B-Instruct-2507 model card. Hugging Face, 2025. URL https://huggingface.co/Qwen/Qwen3-30B-A3B-Instruct-2507. Accessed September 26, 2026.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pp. 3982–3992. Association for Computational Linguistics, 2019. doi: 10.18653/V1/D19-1410. URL https://doi.org/10.18653/v1/D19-1410.

D. Sculley. Web-scale k-means clustering. In Michael Rappa, Paul Jones, Juliana Freire, and Soumen Chakrabarti (eds.), *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, pp. 1177–1178. ACM, 2010. doi: 10.1145/1772690.1772862. URL https://doi.org/10.1145/1772690.1772862.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1708.00489.

Sentence Transformers. all-MiniLM-L6-v2 model card. Hugging Face, 2026. URL https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2. Model card accessed September 26, 2026; year denotes the consulted version.

Youwei Shu, Shaomian Zheng, Dingnan Jin, Wenjie Qu, Ziyao Guo, Qing Cui, Jun Zhou, and Jiaheng Zhang. On representation redundancy in large-scale instruction tuning data selection. *arXiv preprint arXiv:2602.13773*, 2026.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford Alpaca: An instruction-following LLaMA model. GitHub repository, 2023. URL https://github.com/tatsu-lab/stanford_alpaca. Accessed September 26, 2026.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-Instruct: Aligning language models with self-generated instructions. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 13484–13508. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.ACL-LONG.754. URL https://doi.org/10.18653/v1/2023.acl-long.754.

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, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025a. URL https://arxiv.org/abs/2505.09388.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025b. URL https://arxiv.org/abs/2412.15115.

Jie Zhang, Chen Chen, and Lingjuan Lyu. IDEAL: query-efficient data-free learning from black-box models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=ConT6H7MWL.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. BERTScore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=SkeHuCVFDr.

Xuandong Zhao, Lei Li, and Yu-Xiang Wang. Distillation-resistant watermarking for model protection in NLP. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 5044–5055, 2022.

Yue Zheng, Yuhao Chen, Bin Qian, Xiufang Shi, Yuanchao Shu, and Jiming Chen. A review on edge large language models: Design, execution, and applications. *ACM Comput. Surv.*, 57(8): 209:1–209:35, 2025. doi: 10.1145/3719664. URL https://doi.org/10.1145/3719664.

A Supplementary Material

A.1 CLIQ Pipeline Details

A.1.1 Semantic Query Representation and Partitioning

Query Pool Construction. Given a raw instruction dataset in JSON format, we construct a text query for each sample by concatenating the `instruction` field and the optional `input` field:

$$q = \texttt{instruction} \oplus \texttt{input}.$$

If `input` is empty, we use `instruction` only. OQ sampling and CLIQ semantic partitioning use the same public pool of 52,001 Alpaca instructions. OQ submits sampled instructions without modification; CLIQ uses representative examples to generate new queries. We fix 5,000 Alpaca evaluation instructions using seed 42 and compare student outputs with their original reference responses.

Sentence Embeddings. Each query is encoded using Sentence Transformers (Reimers & Gurevych, 2019), specifically `sentence-transformers/all-MiniLM-L6-v2` (Sentence Transformers, 2026). We apply mean pooling with attention masking followed by ℓ_2 normalization:

$$\mathbf{e}(q) = \text{Normalize}\big(\text{MeanPool}(\text{LM}(q))\big).$$

The maximum sequence length is set to 512 with a batch size of 64.

Semantic Partitioning. All query embeddings are partitioned into semantic regions using MiniBatchKMeans with $K = 100$ and a fixed random seed of 42. MiniBatchKMeans (Sculley, 2010) is adopted for scalability and memory efficiency under large query pools.

A.1.2 Coordinated Linguistic Query Construction

Semantic Region-conditioned Prompting. Before generation, we retain source regions with at least $\tau = 5$ original instructions, defining $\mathcal{R} = \{k : n_k \geq \tau\}$ as in Section 3.2. For each retained region, we retrieve up to $M = 1000$ in-region examples nearest to its centroid and use them as textual context for the generator. Each prompt requests $m = 5$ instruction–input pairs. The nominal budget is $Km = 500$, and the realized count is $B_{\text{real}} = m|\mathcal{R}|$. Both the main Qwen and phone-based CLIQ collections contain five queries from each of 99 retained source regions, giving 495 actual distillation queries. OQ and the main selection-only baselines each use 500 queries. Device inference measurements are reported in Section 4.3. The query set is constructed once without target-response feedback and is held fixed across student-training runs. In all analyses, m denotes the number of generated queries per retained source region.

Query Generator and Decoding. We use Qwen3-30B-A3B-Instruct-2507 as the offline query-generation model through an OpenAI-compatible endpoint. The decoding temperature is set to 0.7, and the maximum token budget is set to 16384 to reduce truncation. The configured timeout is 300 seconds, with up to five retries and exponential backoff. There is one initial generation request per retained source region; these requests and any retries are separate from target queries.

Robust Output Parsing. The query-generation model is instructed to output a JSON array of {instruction, input} pairs. We apply a robust parser that extracts JSON content from markdown blocks and repairs truncated outputs by retaining the last complete object and closing the array. The parsed blocks are concatenated into the fixed query collection, with m pairs per retained region. Parsing recovers the generated text; source-cluster size determines membership in $\mathcal{R}$ before generation. The main and phone collections' 495-query count therefore reflects source-region filtering, not discarded generation outputs.

A.1.3 Implementation Details and Hyperparameters

Algorithm 1 outlines query construction. Table 6 summarizes the key hyperparameters used in semantic partitioning and coordinated linguistic query construction. Table 7 reports the student-training, extraction-response generation, and deterministic evaluation settings. The table distinguishes

the main controlled protocol from the phone protocol. The latter uses 200-step checkpoints with LoRA $r = 4, \alpha = 1$ for all four teacher–student configurations.

Table 6: Hyperparameters for semantic partitioning and coordinated linguistic query construction.

Component	**Setting**
Sentence encoder	`sentence-transformers/all-MiniLM-L6-v2`
Max length	512
Embedding batch size	64
Pooling	mean pooling w/ attention mask
Normalization	ℓ_2 normalization
Semantic partitioning	MiniBatchKMeans
Semantic regions K	100
Seed	42
MiniBatch size	$\min(1000, \lvert\mathcal{Q}_0\rvert/10)$
Query generator model	`Qwen/Qwen3-30B-A3B-Instruct-2507`
Temperature	0.7
Max tokens	16384
Timeout	300s
Retries	5 (exponential backoff)
Examples per region M	1000
Requested queries per region m	5
Source-region support threshold τ	5 original instructions
Region retention rule	$n_k \geq \tau$, before generation
Nominal target-query budget Km	500
Retained CLIQ regions (main / phone)	99 / 99
Realized CLIQ queries (main / phone)	495 / 495

Table 7: Student-training, target-response generation, and evaluation settings.

Setting	Value
Query construction seed	42
Extraction-response seed	42 (fixed)
Student training runs	Main: seeds $\{42, 43, 44\}$; phone: one run per strategy
Training steps (main / phone comparison)	500 / 200
Optimizer	Paged AdamW (32-bit states)
Learning rate	2×10^{-5}
LR scheduler	Linear
Warmup	100 steps
Per-device batch size	2
Gradient accumulation steps	8
Effective batch size	16
LoRA rank / alpha (main / phone)	64 / 16 (main); 4 / 1 (phone)
LoRA dropout	0.05
LoRA target modules	`q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj`
Maximum sequence length	1,024
Training precision	FP16
Weight decay / max grad norm	0 / 1.0
Main Qwen extraction target decoding	$T = 0.7$, top-$p = 0.9$, rep.=1.05, sampling, 512 tokens
Evaluation decoding	Deterministic, `do_sample=False`, 256 tokens

Algorithm 1: CLIQ Pipeline

Input: Public query pool $\mathcal{Q}_0$; semantic regions K; source-support threshold τ; examples per region M; queries per retained region m; generator G
Output: Fixed target-query collection $\mathcal{Q}$ and retained source regions $\mathcal{R}$
Encode each query $q \in \mathcal{Q}_0$ into $\mathbf{e}(q)$;
Partition embeddings into K semantic regions via MiniBatchKMeans;
Compute source-cluster sizes $n_k \leftarrow |\mathcal{Q}_k|$;
$\mathcal{R} \leftarrow \{k \in \{1, \ldots, K\} : n_k \geq \tau\}$;
$\mathcal{Q} \leftarrow ()$;
foreach *retained source region* $k \in \mathcal{R}$ **do**
  Compute its centroid c_k;
  Retrieve up to M in-region queries nearest to the centroid;
  Construct a semantic region-conditioned prompt;
  Request m instruction–input pairs from G with the configured retry policy;
  Parse and repair JSON outputs to obtain $\mathcal{C}_k = (q_{k,1}, \ldots, q_{k,m})$;
  Append the query block $\mathcal{C}_k$ to $\mathcal{Q}$;
$B_{\text{real}} \leftarrow |\mathcal{Q}| = m|\mathcal{R}| \leq Km$;
return $\mathcal{Q}, \mathcal{R}$;

A.1.4 Response-Length Characterization

We use the same 993 paired non-empty outputs from the OQ and CLIQ workloads and tokenize each response with the Qwen1.5 tokenizer. Table 8 summarizes response lengths. The main Qwen target has a configured maximum generation length of 512 tokens. Its median equals this limit; the reported lengths reflect this generation cap.

Table 8: Response lengths in Qwen1.5 tokens on the 993 paired non-empty outputs.

Target	Mean	Median
Qwen2.5-7B-GPTQ-INT4	409.4	512
Gemma 3 1B IT (Q8)	58.4	63
Gemma 4 E2B IT (mixed-bit)	60.7	65

A.2 Additional Results and Sensitivity Analyses

A.2.1 Comprehensive Evaluation Results

Main-text Table 1 reports the full available metric set for the Qwen teacher, base students, and extracted students. Main-text Table 3 reports all six metrics for both Gemma teachers and both phone-supervised student sizes. The additional Gemma 4 checkpoint results are reported separately in Supplementary Section A.2.2.

Table 9: Additional quantization configurations. Scores are reference-based point estimates (%); OQ and CLIQ use FP16 and INT4 teachers, respectively.

Model	Role	Teacher quantization	Student quantization	Query	BERT-F1	BLEU	R-1	R-2	R-L
Qwen2.5-7B	Teacher	FP16	–	–	83.69	2.94	22.08	9.36	16.85
		INT4	–	–	84.22	**3.15**	21.92	9.58	16.56
Qwen3-1.7B	Base	–	FP16	–	81.53	1.70	12.35	4.46	8.92
		–	INT8	–	81.40	1.73	12.23	4.36	8.70
	Student	FP16	INT8	OQ	83.75	2.50	22.30	9.49	16.68
		INT4	INT8	CLIQ	**84.35**	3.04	**23.42**	**10.60**	**17.60**

Table 9 reports additional heterogeneous configurations spanning teacher quantization, student quantization, and query organization. The OQ rows use an FP16 teacher and the CLIQ rows use

an INT4 teacher. The table therefore describes individual configurations; the main-text strategy comparisons hold teacher quantization fixed.

A.2.2 GEMMA 4 E2B PHONE VALIDATION

Teacher and deployment. The instruction-tuned teacher is `google/gemma-4-E2B-it`. The model card reports 2.3B effective parameters and approximately 5.1B including per-layer embeddings (Google DeepMind, 2026b). Its text configuration specifies 35 layers, hidden size 1,536, eight attention heads and one KV head (Google DeepMind, 2026a). Our text-only queries use the Tensor-G5-compiled package `gemma-4-E2B-it_Google_Tensor_G5.litertlm` on the Pixel 10 Pro Fold. LiteRT mobile uses mixed 2/4/8-bit weights (LiteRT Community, 2026).

Protocol and checkpoint results. The student is Qwen1.5-1.8B-Chat-GPTQ-INT4, trained off-device using 500 OQ or 495 CLIQ target responses. The two strategies each have one training run; Table 10 reports checkpoints at 100–500 steps from those runs. Evaluation follows the Alpaca-reference evaluation protocol.

Table 10: Gemma 4 E2B IT to Qwen1.5-1.8B-Chat-GPTQ-INT4: single-run checkpoint scores against Alpaca references (%). Bold marks the better strategy at each step and metric.

Steps	Strategy	BERT-F1	BLEU	R-1	R-2	R-L	R-Lsum
100	OQ	**81.1897**	**1.5611**	**17.0519**	**5.4394**	**11.2980**	**13.4970**
100	CLIQ	81.1015	1.5289	16.9661	5.4000	11.2189	13.4361
200	OQ	80.7959	1.3923	16.5242	5.0509	11.0219	13.0515
200	CLIQ	**81.6238**	**2.0009**	**18.2019**	**5.7745**	**12.4161**	**13.8559**
300	OQ	80.7959	1.3923	16.4995	5.0516	11.0222	**13.0637**
300	CLIQ	**81.9225**	**1.9650**	**17.6753**	**5.3957**	**11.9591**	12.9668
400	OQ	80.7959	1.3923	16.5235	5.0506	11.0168	**13.0603**
400	CLIQ	**82.1615**	**1.9684**	**17.6621**	**5.4004**	**11.9260**	12.8916
500	OQ	80.7959	1.3923	16.5016	5.0528	11.0147	**13.0621**
500	CLIQ	**82.3068**	**2.0011**	**17.6667**	**5.3747**	**11.8920**	12.8811

The main phone table uses the 200-step checkpoint for both 1.8B teacher configurations; later checkpoints below are supplementary training-dynamics results.

At 100 steps, CLIQ is slightly below OQ on all six metrics. Its BERT-F1 then rises from 81.1015 to 82.3068, while OQ decreases to 80.7959 at 200 steps and remains at that reported BERT-F1 through 500 steps. The final BERT-F1 advantage is 1.5109 points. CLIQ's ROUGE metrics peak at 200 steps and subsequently decline; ROUGE-Lsum falls below OQ at 300–500 steps.

Additional 0.5B student at 200 steps. Table 11 reports the added Gemma 4 E2B to Qwen 0.5B results on the 5,000-example validation split. CLIQ improves BERT-F1 by 0.8648 points and improves all five other reported metrics. The student is Qwen1.5-0.5B-Chat-GPTQ-INT4 and reuses the 500 OQ / 495 CLIQ Gemma 4 phone-response pools. These are single-run point estimates at 200 steps with LoRA $r = 4, \alpha = 1$, consistent with the common phone protocol.

Table 11: Gemma 4 E2B to Qwen 0.5B at 200 steps on the 5,000-instruction Alpaca evaluation set. All metrics are scaled by 100; Δ is CLIQ minus OQ in points.

Strategy	BERT-F1	BLEU	R-1	R-2	R-L	R-Lsum
OQ	75.3464	0.5913	11.9057	3.7204	8.0922	9.3515
CLIQ	**76.2112**	**0.6581**	**12.7921**	**3.9378**	**8.7126**	**10.0189**
Δ	+0.8648	+0.0668	+0.8864	+0.2174	+0.6204	+0.6674

Device inference measurements. Both teachers use the Pixel 10 Pro Fold's Tensor G5 NPU. Relative to the Gemma 3 configuration, Gemma 4's reported first-token latency decreases by approximately 39.4% for OQ and 39.3% for CLIQ, while generation speed increases by 32.9% and 34.0%,

respectively. These descriptive percentages are calculated from the rounded means in Table 3; they compare complete model/deployment configurations. Each teacher supplies 500 OQ and 495 CLIQ supervision responses, with one response per query retained for student training.

A.2.3 Reference-Based Gap Closure

Figure 5 visualizes the four 4B strategies in Table 2 using Equation 5. The base score of 75.04 (Table 1) and teacher score of 84.22 are fixed, so both means and standard deviations are normalized by the same 9.18-point gap.

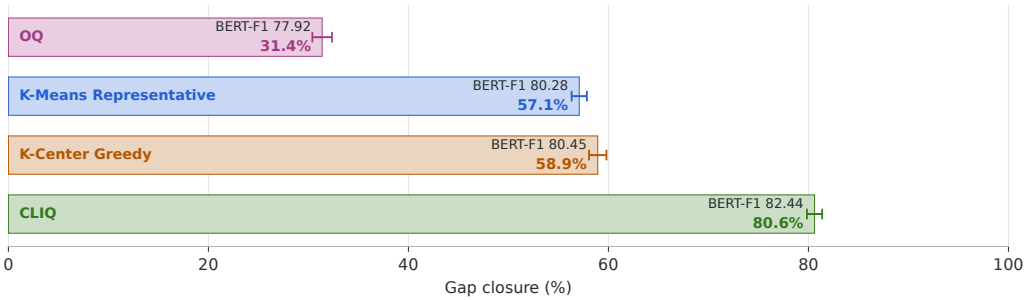


Figure 5: 4B-student gap closure from Table 2. Bars label strategies, BERT-F1 means, and gap closure; error bars show normalized SD.

A.2.4 Training Dynamics under a Fixed Query Budget

Table 5 and Figure 6 report the Qwen3-1.7B student at 100, 300, and 500 training steps under a fixed nominal budget of 500 target queries, with GPTQ-INT4 and GPTQ-INT8 student configurations. Results are mean ± standard error (SE) over three training runs. CLIQ initially remains below OQ at 100 steps, improves sharply by 300 steps, and changes little thereafter. At 500 steps, its BERT-F1 gains over OQ are 2.89 points under INT4 and 2.66 under INT8. Across both student-precision settings, CLIQ's advantage emerges with additional training steps.

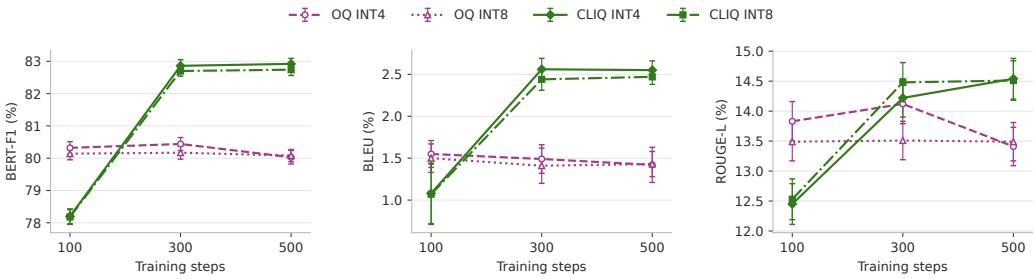


Figure 6: Qwen3-1.7B training dynamics under GPTQ-INT4 and GPTQ-INT8 under a nominal 500-query budget. Error bars show ± one standard error (SE) over three runs.

Figure 7 provides a supplementary point-estimate view of student performance on the Alpaca evaluation set over the course of training.

A.2.5 Clustering Granularity and Query Allocation

Table 12 varies the number of semantic regions K and the number of generated target queries per retained source region m, with evaluation on Dolly and Alpaca.

Clustering Granularity. Results vary with both the dataset and query allocation. Moderate granularities of 25–50 regions often perform competitively, while $K = 100$ gives both strong and weak outcomes depending on m.

Table 12: Clustering and query-allocation sensitivity for the Qwen1.5-4B-GPTQ-INT4 student. Nominal allocations equal Km; m is the number of generated queries per retained source region. Scores are point estimates (%).

Dataset	Clusters K	Queries/region m	Nominal queries	BERT-F1	BLEU	R-1	R-2	R-L	R-Lsum
Dolly	25	4	100	81.384	2.092	17.905	6.196	12.089	14.402
	25	8	200	82.272	2.325	18.325	6.263	12.397	14.686
	25	10	250	82.267	2.473	18.546	6.258	12.550	15.017
	50	4	200	82.558	2.643	18.335	6.377	12.258	14.825
	50	10	500	82.345	2.608	18.399	6.268	12.500	14.818
	100	8	800	78.707	1.476	16.895	5.895	11.298	13.499
	100	10	1000	**83.042**	**2.972**	**18.831**	**6.556**	**12.667**	**15.138**
Alpaca	25	4	100	81.582	1.861	17.749	5.873	11.563	14.039
	25	8	200	81.451	2.010	17.749	5.873	11.563	14.039
	25	10	250	81.936	2.256	18.177	6.088	11.977	14.508
	50	4	200	80.914	1.885	17.701	6.060	11.669	14.165
	50	10	500	**82.518**	**2.460**	18.133	6.084	11.980	14.525
	100	4	400	82.056	2.236	**18.379**	**6.160**	**12.052**	**14.617**
	100	8	800	78.714	1.307	16.691	5.608	10.928	13.239

Queries per Retained Region. Scores vary non-monotonically as m increases from 4 to 8 or 10, with particularly low scores in some $K = 100$ configurations.

Interpreting the Allocation Sweep. The table reports the nominal allocation $B_{\text{nominal}} = Km$. The realized target-query count is $B_{\text{real}} = m|\mathcal{R}|$, where $\mathcal{R} = \{k : n_k \geq \tau\}$ contains source regions retained before generation. Per-configuration retained-region counts are unavailable for this sweep; the table reports nominal allocations. At fixed m, increasing K changes both the nominal budget and the partition; at fixed K, increasing m changes both the budget and within-region sampling. These results describe sensitivity to the joint choice of partition granularity and query budget.

A.3 Semantic Analysis of Query Space

This section provides supplementary query-space diagnostics for the interpretation in Section 5.1. We examine semantic coverage, cluster-level allocation, intra-cluster redundancy, and within-region diversity.

A.3.1 Semantic Coverage under Limited Budgets

Figure 8 measures cluster coverage using saved cluster labels and shuffled query sequences (seed 42). OQ uses prefixes of the shuffled public pool; CLIQ uses prefixes of its shuffled generated collection. At the largest displayed budget, CLIQ uses all 495 queries. This is a separate coverage diagnostic from the region-balanced query-budget experiment in Figure 3.

The 500-query OQ prefix covers 97 clusters, whereas the fixed OQ distillation collection in Figure 4 covers 95; the two figures use different sampled subsets. CLIQ covers more clusters at every displayed budget in this diagnostic, consistent with broader coverage under limited query access.

A.3.2 Query Allocation across Semantic Regions

Figure 4(a) in the main text compares query shares for the 52,001-instruction Alpaca pool, the actual 500-query OQ distillation set, and the 495-query CLIQ set. OQ covers 95 of the original 99 clusters; CLIQ assigns five queries to each generation-source cluster. The plot uses the original saved labels. For CLIQ, these labels identify the source regions used to generate each query block.

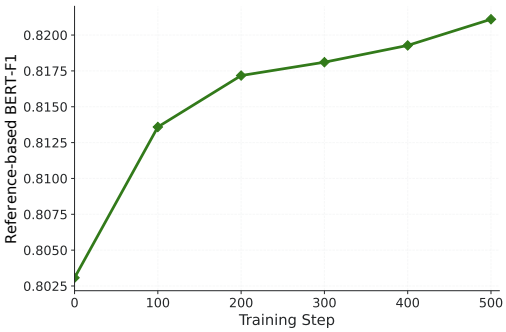


Figure 7: CLIQ training progress under a nominal 500-query budget, measured against original Alpaca references. Values are point estimates.

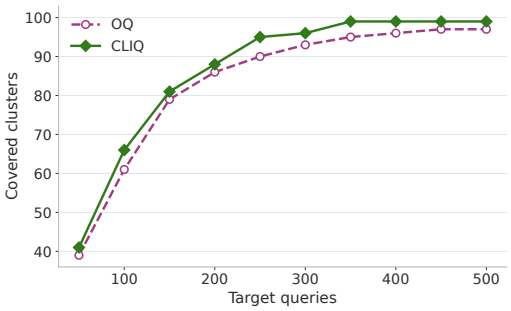


Figure 8: Cluster coverage in the shuffled-query diagnostic. Dashed circles denote OQ; solid diamonds denote CLIQ.

A.3.3 Within-region Query Similarity

Figure 4(b) in the main text shows the distribution of mean pairwise cosine similarity within semantic regions containing at least two queries (99 for the Alpaca pool, 86 for OQ, and 99 for CLIQ). All distinct pairs are included within each eligible cluster. OQ's nine singleton clusters are covered but cannot contribute a pairwise similarity, and are excluded rather than assigned zero. CLIQ's lower mean and median are consistent with less semantic redundancy, although the distributions overlap.

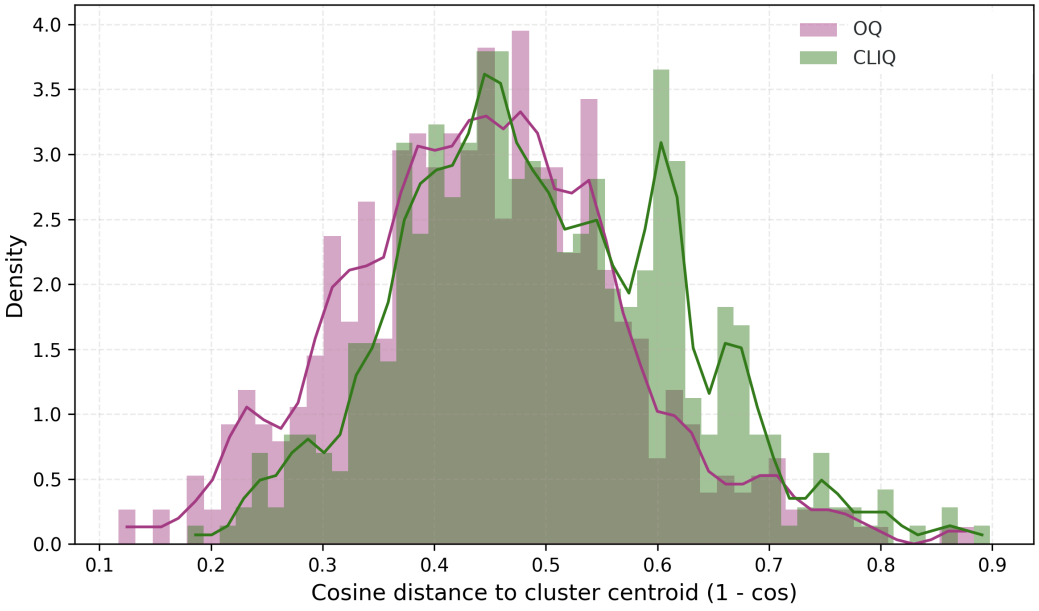


Figure 9: Cosine distances from queries to cluster centroids. CLIQ queries cover both central and peripheral regions.

A.3.4 Global Semantic Distribution

Figure 10 shows UMAP (McInnes & Healy, 2018) projections of sentence embeddings for the original Alpaca pool and CLIQ-generated query sets. Each point represents one query. All sets use the same embedding model; the panels show different cluster counts and per-cluster allocations.

In the displayed projections, both the original instructions and CLIQ queries form several groups. Their apparent density varies with the query set and allocation. These two-dimensional projections

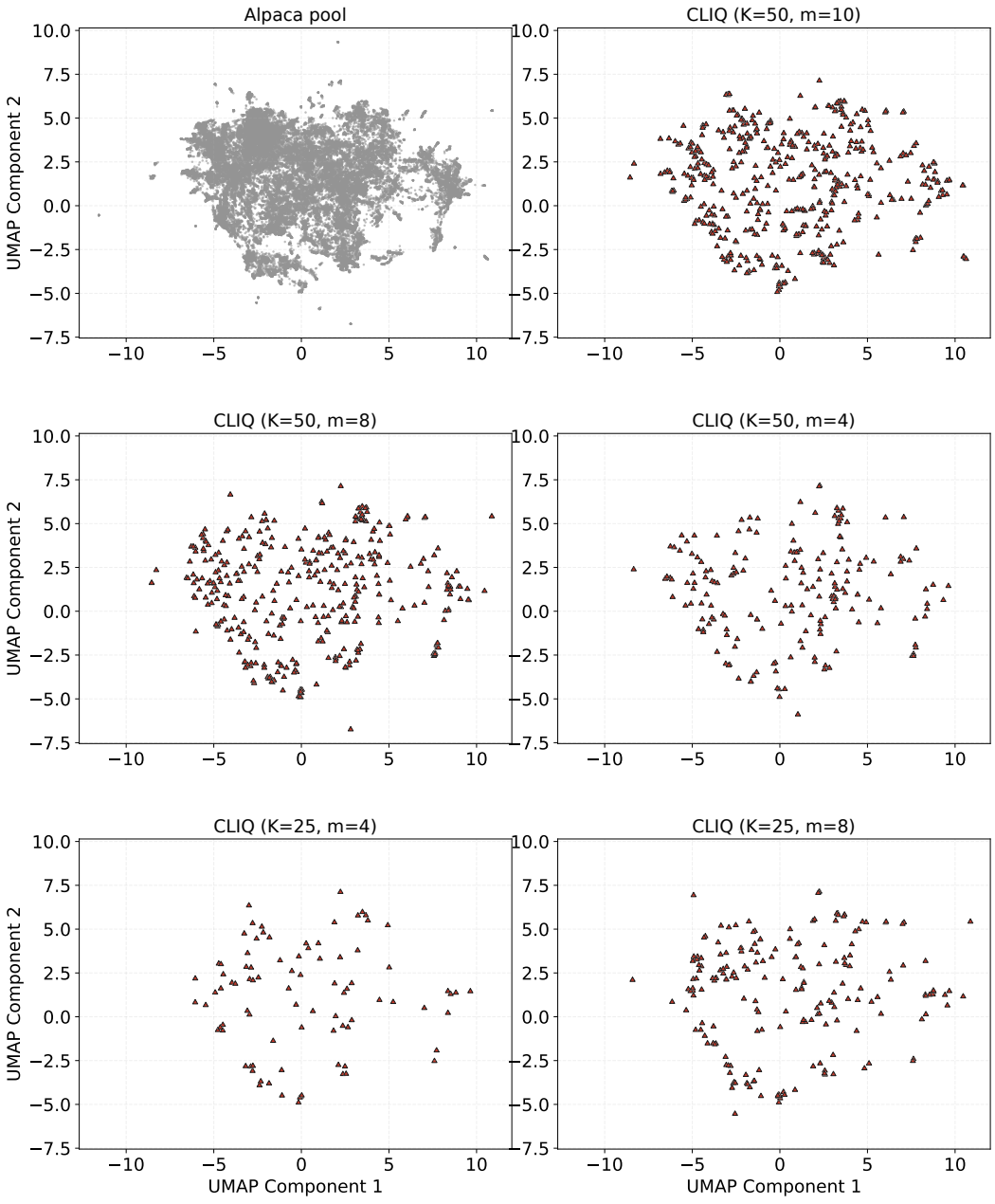


Figure 10: UMAP projections of the Alpaca pool and CLIQ query sets across cluster counts K and per-cluster allocations m. Gray circles show pool queries; outlined triangles show CLIQ.

provide a qualitative view of query distribution. Table 12 reports extraction performance for different allocations.

A.3.5 WITHIN-CLUSTER QUERY DISTRIBUTION

We also examine the positions of generated queries within their source regions. Figure 9 shows that CLIQ queries occur at a range of distances from their corresponding centroids, rather than only near the centers.

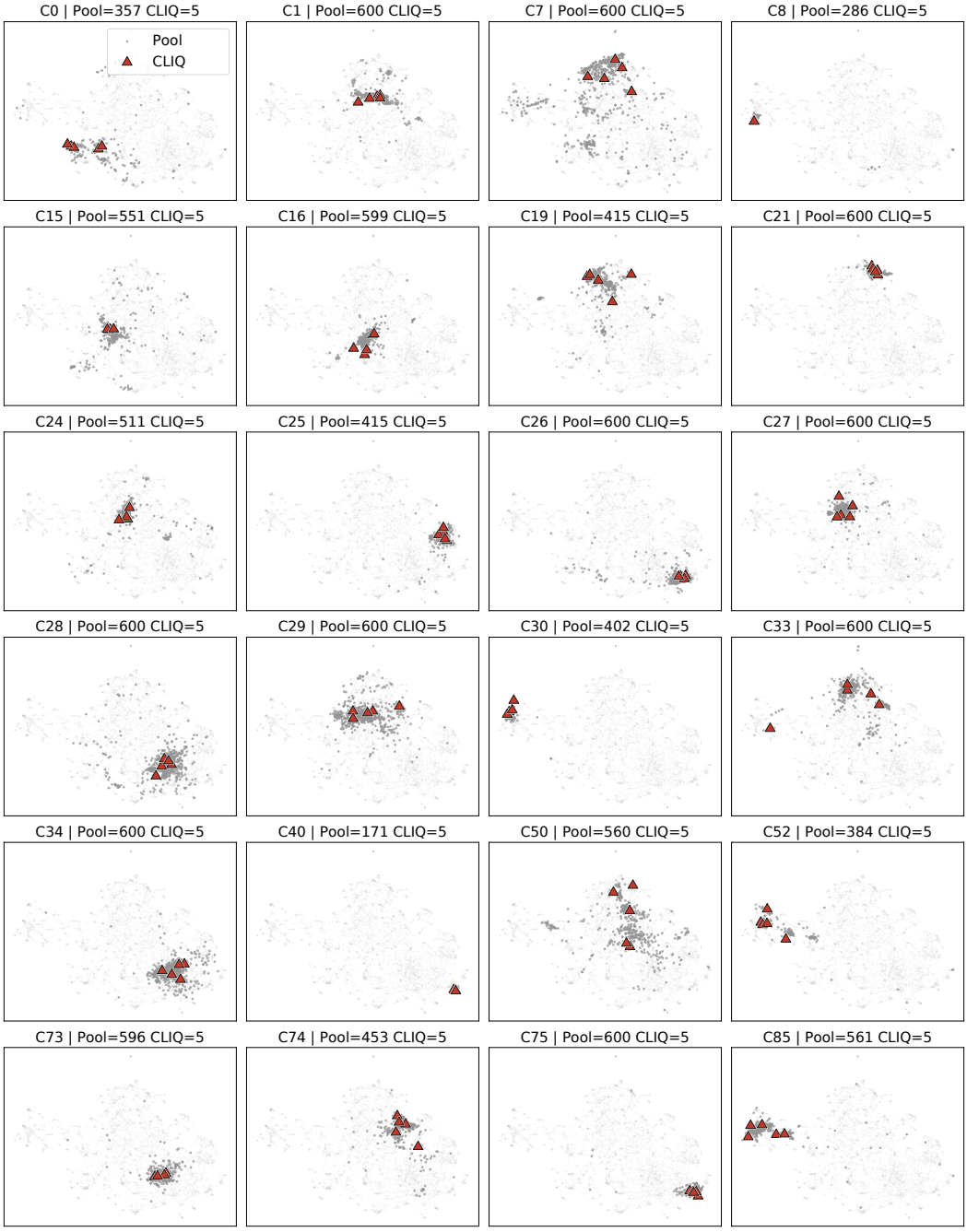


Figure 11: Per-cluster projections of Alpaca pool queries and CLIQ-generated queries, with $m = 5$ generated queries per displayed region. Gray circles show original Alpaca instructions; larger outlined triangles show CLIQ.

Figure 11 displays five CLIQ queries alongside the original pool queries for each selected region. The generated queries occupy different positions across these panels, illustrating within-region variation.

Taken together, the allocation and similarity diagnostics are consistent with lower query redundancy under CLIQ. The projections illustrate query distribution, while the extraction experiments measure student performance.

A.4 Cluster-conditioned Query Examples

Figure 12 presents queries from semantic regions visualized in Figures 10 and 11.

Concept Explanation Cluster (Cluster 7)

Cluster Theme. Definition-oriented and explanatory instructions involving scientific concepts, algorithms, and general knowledge.

Original Queries	CLIQ-generated Queries
What does DNA stand for?	**Explain the concept of a zero-knowledge proof in cryptography using a real-world analogy.** *A scenario where one person proves to another that they know a secret without revealing the secret itself*
Explain the concept of a bubble sort algorithm to a non-technical audience.	**Describe how a transformer architecture works in natural language processing, focusing on attention mechanisms.**
Explain the concept of a linear regression model using simple language.	**Explain the difference between supervised and unsupervised learning with one practical example for each.**
Explain what an API is.	**What is the purpose of a firewall in network security, and how does it protect against cyber threats?**
Explain the concept of entropy in thermodynamics.	**Summarize the concept of edge computing and its advantages over traditional cloud computing in IoT applications.**

Mathematical Reasoning Cluster (Cluster 30)

Cluster Theme. Mathematical reasoning tasks involving algebraic manipulation, derivatives, and symbolic computation.

Original Queries	CLIQ-generated Queries
Convert the given equation into an algebraic expression. $3x + 5y = 9$	**Find the vertex of the parabola defined by the quadratic equation** $y = 2x^2 - 8x + 5$**.** $y = 2x^2 - 8x + 5$
Find the solution to the quadratic equation. $3x^2 + 7x - 4 = 0$	**Solve the system of equations using substitution method:** $3x + y = 10$ **and** $2x - 3y = -5$**.** $3x + y = 10,\ 2x - 3y = -5$
Compute the following expression. $2^3 + 3^2$	**Evaluate the limit of the function** $f(x) = (x^2 - 9)/(x - 3)$ **as** x **approaches** 3**.** $f(x) = (x^2 - 9)/(x - 3),\ x \to 3$
Find the slope of the following line. $y = 3x - 2$	**Determine the area under the curve** $y = x^2$ **from** $x = 0$ **to** $x = 3$ **using integration.** $y = x^2,\ x = 0$ *to* $x = 3$
Compute the derivative of $2x^2 + 5x$**.**	**Find the magnitude and direction of the vector** $v = 4i - 3j$**.** $v = 4i - 3j$

Haiku Writing Cluster (Cluster 40)

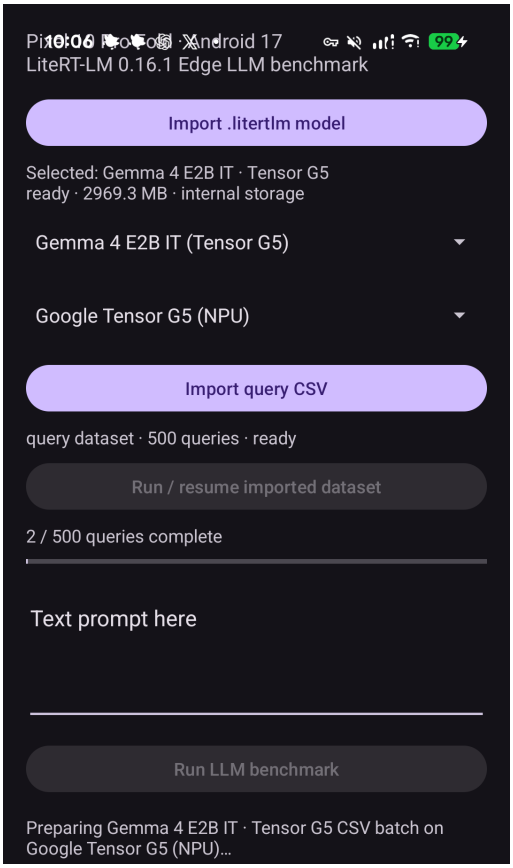
Cluster Theme. Short-form creative writing tasks centered on haiku generation and nature-oriented imagery.

Original Queries	CLIQ-generated Queries
Generate a haiku poem.	**Compose a haiku that captures the essence of a solitary traveler at dawn.** *Traveler walking through misty hills at sunrise*
Compose a haiku about cats.	**Create a haiku using only sensory details from the natural world, focusing on sound and touch.** *Forest at twilight, wind rustling through dry leaves*
Design a haiku about snow.	**Write a haiku that metaphorically represents the passage of time using seasonal imagery.** *Cherry blossoms fall like time slipping through fingers*
Write a Haiku of 3 lines, with the first containing 5 syllables and the remaining lines containing 7 syllables.	**Generate a haiku about a moment of unexpected beauty in an ordinary setting.** *A cracked sidewalk sprouting green through concrete*
Write a haiku with the following phrase: "frog in the pond".	**Craft a haiku that reflects on inner peace using elements of water and stillness.** *Still pond after rain, ripple fades into silence*

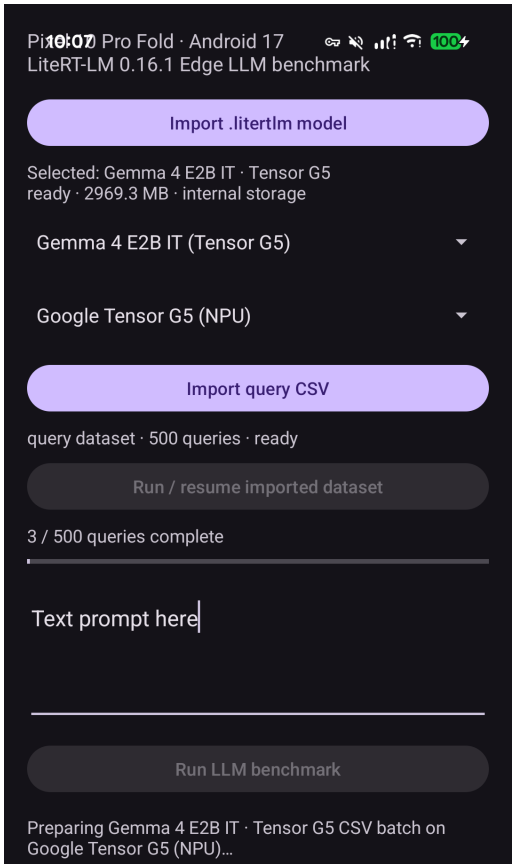
Figure 12: Representative original and CLIQ-generated queries from different semantic clusters.

A.5 PHONE EXPERIMENT SCREENSHOTS

Figure 13 shows the response-collection interface used on the Pixel 10 Pro Fold. The app displays LiteRT-LM 0.16.1 and the Google Tensor G5 NPU backend, together with the imported query count and batch progress. These in-progress captures document the device-side workflow; aggregate measurements and student results are reported in Table 3.



(a) Gemma 4 E2B IT: 2/500 completed.



(b) Gemma 4 E2B IT: 3/500 completed.

Figure 13: Phone-side response collection on the Tensor G5 NPU. Both screenshots identify Gemma 4 E2B IT and show progress through a 500-query batch: (a) 2/500 and (b) 3/500 completed. Screenshots are cropped only below the app content.