Byte-Level Speculative Routing for Bandwidth-Constrained Device-Cloud LLM Inference

Anonymous ACL submission

Abstract

Edge-cloud collaborative inference enables deploying large language models on resource-constrained devices, but existing methods incur prohibitive communication overhead. We propose Draft-Confidence Policy (DCP), a 740K-parameter neural router mounted on the frozen edge SLM that performs per-token adaptive routing at byte-level granularity. DCP jointly predicts draft confidence and selects between speculative verification (12 bytes) and dense cloud decoding (1 byte) using live context features. Trained on 50K traces from an 8B teacher, DCP achieves zero-shot transfer to 32B and 72B cloud LLMs without retraining. Across five benchmarks, DCP retains 86–100% of full-cloud accuracy while reducing per-token communication to 1–12 bytes—a $10^3\times$ reduction over hidden-state transmission. Ablations confirm that learned hidden-state routing outperforms scalar-threshold baselines, which fail due to systematic SLM overconfidence. DCP's primary advantage is communication efficiency at comparable accuracy, enabling deployment on bandwidth-constrained links. .

1 Introduction

Deploying large language models on edge devices requires balancing local compute constraints against the latency and bandwidth costs of cloud offloading. Edge devices lack the memory and compute for billion-parameter inference, yet offloading every token to the cloud is prohibitively expensive in terms of bandwidth.

Recent edge-cloud collaborative inference systems use speculative decoding (SD), where an edge draft model proposes tokens and a cloud model verifies them in batch. However, existing systems—PicoSpec, PipeSD, FlexSpec, and Conformal Sparsification—all rely on fixed communication strategies: static speculation windows, offline-tuned thresholds, or hand-derived formulas. These rules cannot adapt to per-token variability. Content words on reasoning steps may require full cloud attention, while function words are trivially predictable.

A natural question is whether a simple confidence threshold on the SLM's token probability could suffice for per-token routing. We investigate this by training a linear probe on SLM hidden states and find it significantly outperforms scalar probability (probe AUC=0.876 vs. scalar AUC=0.604), suggesting hidden states encode richer acceptance information. However, an end-to-end sweep reveals that scalar thresholds are insufficient: all five thresholds (0.5–0.9) yield 83.0% on GSM8K, 7pp below DCP's 90.0%. The SLM is systematically overconfident, concentrating probability mass where it is wrong nearly half the time.

Motivated by this, we propose DCP (Device-Cloud collaborative inference with learned Policy), a 740K-parameter neural policy that learns per-token routing. Mounted on the edge SLM's frozen hidden states, DCP selects between draft mode (speculative decoding, 12 bytes) and dense mode (full cloud compute, 1 byte) at each decoding step. Trained on 50K cloud LLM traces from GSM8K, DCP achieves 86–100% accuracy retention across five benchmarks. A single policy transfers zero-shot to 32B–72B LLMs without retraining, demonstrating generalization beyond the training cloud model.

Our contributions are:

1. **Learned per-token routing.** DCP replaces hand-tuned thresholds with a neural policy, achieving accuracy within measurement noise of the DENSE baseline on GSM8K, MMLU, and ARC-Challenge, and 97.3% retention on HumanEval. A scalar probability baseline yields uniformly lower accuracy because the SLM is systematically overconfident.

2. **Zero-shot cross-model transfer.** A single policy trained on Qwen3-8B transfers directly

to Qwen3-32B and Qwen2.5-72B without retraining. On Qwen3-32B, DCP achieves 98% accuracy (vs. an estimated 94% DENSE baseline; see Section 9 for caveats on estimated baselines).

3. **Communication efficiency.** DCP transmits 1–12 bytes per step ($10^3\times$–$10^6\times$ reduction vs. hidden states).

2 Related Work

Single-device speculative decoding. Leviathan et al. (2023) and Chen et al. (2023) introduced the draft-then-verify paradigm. Subsequent work improved drafting mechanisms: Medusa (Cai et al., 2024a) attaches prediction heads to the target model. EAGLE (Li et al., 2024b) performs autoregression at the feature level, raising draft acceptance to ∼0.8. Sequoia (Chen et al., 2024a) uses dynamic programming to optimize speculation tree structure. SpecExec (Svirschevski et al., 2024) scales tree verification to thousands of draft tokens. Cascade Speculative Drafting (Chen et al., 2024b) replaces neural draft models with statistical language models. Speculative Streaming (Bhendawade et al., 2024) fuses drafting into the target model via multi-stream attention. These methods assume single-device deployment where draft and target models share hardware. DCP targets edge-cloud settings where models are separated by a network link.

Edge-cloud speculative decoding. PicoSpec (Zhang et al., 2026) introduces an asynchronous pipeline with top-K vocabulary compression. PipeSD (Han et al., 2026) formulates SD as a streaming pipeline with Bayesian-optimized thresholds. FlexSpec (Li et al., 2026) handles draft-target model drift. Conformal Sparsification (Bhattacharjee et al., 2025) uses conformal prediction to sparsify token distributions. These methods differ in how they decide what to transmit, but all use fixed or externally-tuned rules. DCP replaces these rules with a learned neural policy that selects modes from SLM hidden states at each step.

Sparse attention and KV cache compression. StreamingLLM (Xiao et al., 2024) keeps recent tokens plus attention sinks. H_2O (Zhang et al., 2023) evicts KV entries by accumulated attention scores. SnapKV (Li et al., 2024a) and PyramidKV (Cai et al., 2024b) compress KV caches for long-context efficiency. These methods operate entirely on the cloud side and are orthogonal to DCP's mode selection.

Edge-cloud model partitioning. Neurosurgeon (Kang et al., 2017) partitions DNN layers between mobile and cloud. DDNN (Teerapittayanon et al., 2017) adds early-exit classifiers. JointDNN (Eshratifar et al., 2019) jointly optimizes partitioning and resource allocation. These methods target CNN inference and transmit intermediate activations (MB per step). DCP targets LLM inference and transmits only discrete mode decisions.

Property	Neuro/ DDNN/JointDNN	Always-DRAFT	DCP
Per-step comm.	∼2 MB	∼13 B	**1–12 B**
Model mod.	Yes	No	No
Hetero. models	No	Yes	**Yes**
Zero-shot	No	No	**Yes**
Learns decisions	Heuristic	No	**Yes**

Table 1: Comparison with prior edge-cloud collaboration methods.

3 Method

3.1 Problem Formulation

Given an input sequence $x = (x_1, ..., x_T)$, we generate output tokens $y_1, ..., y_N$ using a cloud LLM p_{large} and an edge SLM p_{small}. The cloud LLM provides strong reasoning capability but incurs communication overhead per request; the edge SLM is fast and local but lacks the capacity for complex reasoning. At each decoding step t, the edge SLM computes hidden states h_t. A policy π_θ processes h_t and produces:

$$c_t = \sigma\big(f_{\text{draft}}(h_t)\big) \quad \text{(draft confidence)} \tag{1}$$

$$m_t = \arg\max\ f_{\text{mode}}(h_t, \phi_t) \quad \text{(mode selection)} \tag{2}$$

where f_{draft} and f_{mode} are lightweight MLP heads, and ϕ_t aggregates context features (recent acceptance rate, KV length). The mode $m_t \in \{0, 1\}$ selects between draft and dense inference.

Two inference modes:

- **Draft** ($m_t = 0$): SLM generates $k = 5$ candidate tokens. Edge sends token IDs (∼12 bytes). Cloud verifies via rejection sampling (Leviathan et al., 2023). Accepted tokens are committed; rejected tokens trigger cloud resampling from the corrected distribution.

- **Dense** ($m_t = 1$): Edge sends a 1-byte fallback signal. Cloud runs full autoregressive decoding.

3.2 Policy Architecture

The policy is mounted on the frozen SLM and takes its last-token hidden state $h_t \in \mathbb{R}^{1024}$ as input:

- **Shared encoder:** Linear$(1024 \rightarrow 512)$ + LayerNorm + ReLU.
- **Draft head:** Linear$(512 \rightarrow 1)$ + Sigmoid. Predicts $P(\text{SLM top-1} = \text{LLM next token})$.
- **Mode head:** Linear$(512 + 3 \rightarrow 2)$ + Softmax. Uses aggregate features $\phi_t = [\bar{c}_{t-5:t}, \text{kv_len}/32768, \text{accept_rate}]$.

Total parameters: ∼740K.

3.3 Training from Cloud Traces

We collect ∼50K decoding steps from 200 GSM8K prompts (training split). For each step, the trace records:

- **Draft target**: 1 if the SLM's top-1 token matches the LLM's greedy argmax.
- **Mode target**: Draft when the LLM accepts ≥ 3 of $k = 5$ SLM draft tokens; Dense otherwise.[1]

The training loss is:

$$\mathcal{L} = \text{BCE}(\hat{c}, c^*) + \lambda \text{CE}(\hat{m}, m^*) \quad (3)$$

with $\lambda = 0.3$. We train for 16 epochs using AdamW (lr=10^{-4}, weight decay=10^{-5}, batch size 256).

3.4 Communication Protocol

The edge → cloud message consists of a mode byte followed by mode-specific payload:

- **Draft**: mode (1B) + token count (1B) + $k = 5$ token IDs (10B) = 12 bytes
- **Dense**: mode (1B) only

The cloud response is 2–3 bytes: accepted count plus the next token ID. Transmitting per-step hidden states costs ∼2 MB (4096-dim × 2 bytes), and full KV cache for 32K context costs ∼2 GB.

4 Experiments

4.1 Experimental Setup

Models. We use Qwen3-0.6B (Yang et al., 2025) as the fixed edge SLM (600M parameters, hidden dimension 1024). For the cloud LLM, we evaluate three model variants spanning different scales (Table 3) to assess cross-model generalization. All models use bfloat16 precision.

Aspect	KV cache xfer	Hidden state xfer	DCP
Per-step data	2 GB (32K ctx)	2 MB (4096-d)	1–12 B
Latency model	Full RTT + GB/s	Full RTT + MB/s	Mode byte + RTT
Privacy	Exposes all states	Exposes all states	Minimal exposure

Table 2: Communication cost comparison. DCP achieves orders-of-magnitude reduction in per-step data transfer.

Cloud LLM	Params	Hidden	Family
Qwen3-8B	8.0B	4096	Qwen3 (in-domain)
Qwen3-32B	32.8B	5120	Qwen3 (larger)
Qwen2.5-72B	72.7B	8192	Qwen2.5 (largest)

Table 3: Cloud LLM variants evaluated. The edge SLM (Qwen3-0.6B) is fixed across all experiments.

Benchmarks. We evaluate on five benchmarks spanning diverse reasoning domains: GSM8K (Cobbe et al., 2021) (mathematical reasoning, 100 samples), MMLU (Hendrycks et al., 2020) (broad world knowledge, 100 samples across 46 subjects), ARC-Challenge (Clark et al., 2018) (science reasoning, 100 samples), HumanEval (Chen et al., 2021) (code generation, 50 samples), and LongBench v2 (Bai et al., 2025) (long-context understanding, 100–503 samples). These benchmarks cover both free-form generation (GSM8K, HumanEval) and multiple-choice questions (MMLU, ARC, LongBench), allowing us to assess whether learned routing value differs by task structure. All evaluations use greedy decoding with enable_thinking=False. We report 95% Wilson binomial confidence intervals for all accuracy measurements; at $n = 100$, these are approximately ±6pp for 90% accuracy and ±9pp for 75% accuracy.

4.2 Main Results

Figure 2 reports accuracy across five benchmarks with Qwen3-8B DENSE and DCP (0.6B+8B). Key observations:

Near-perfect retention on short-context benchmarks. On GSM8K, MMLU, and ARC-Challenge, DCP matches the DENSE baseline accuracy within measurement noise (95% Wilson CIs overlap). The policy uses DRAFT mode for 50.1% of GSM8K decoding steps, with the cloud LLM accepting 88.3% of draft tokens. On HumanEval, DCP scores 72% vs. DENSE 74% (a 1-sample dif-

[1] The threshold $\geq 3/5$ balances draft utilization and quality. In our trace data, 50.1% of steps meet this criterion.

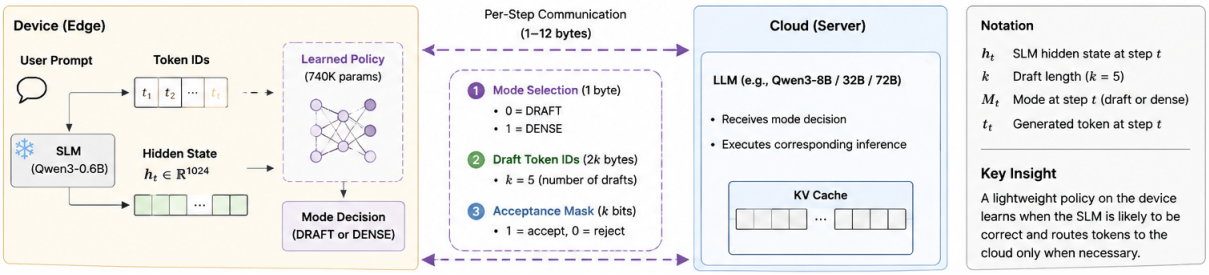

Figure 1: Policy π_θ architecture. Total: 740K parameters. SLM hidden states ($d = 1024$) are projected to 512 dimensions via a shared encoder, then fed to two lightweight heads: a draft-confidence head (scalar sigmoid output) and a binary mode selector (2-way softmax).

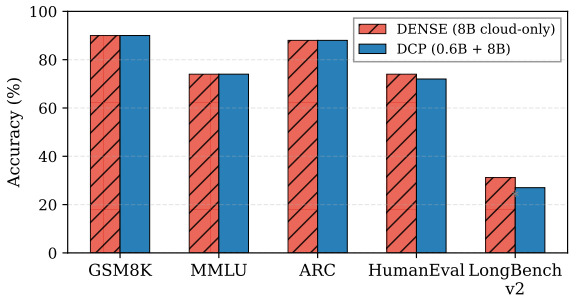

Figure 2: Main results across five benchmarks.

ference, 97.3% retention). The discrepancy stems from duplicate markdown code-block delimiters in DCP output, causing the standard code extractor to miss one solution—not from degraded generation quality.

We also evaluate DCP on $n = 200$ GSM8K samples: DCP achieves 87.0% (174/200; 95% CI ±4.7pp) with DRAFT=38.8%, consistent with the $n = 100$ result of 90.0% (overlapping 95% CIs). The 3pp fluctuation reflects inherent GSM8K variance at moderate sample sizes, as observed across all measurements ($n = 100$: 90–94%; $n = 200$: 87%; $n = 500$: 85%). Scaling the DENSE baseline to $n = 500$ yields 85.0% (425/500; 95% CI ±3.1pp), consistent with the $n = 100$ result of 90%.

Long-context degradation. On LongBench v2, DCP achieves 86.5% retention (27.0% vs. 31.2% DENSE). This comparison is not fully controlled: DCP uses MAX_LEN=32768 vs. DENSE's 64000, halving usable context, which likely accounts for a substantial portion of the gap. Additional factors include: (1) the policy is trained on GSM8K short-context traces only; (2) with 95% CIs of ±9pp at $n = 100$, the 4.2pp gap is within sampling noise. We note that the reduced context window is a practical constraint of running dual models (SLM + LLM) on the same device, not a fundamental limitation of the routing policy itself.

Under greedy decoding, speculative decoding with correct KV cache management preserves the LLM's output distribution exactly (Leviathan et al., 2023). The DENSE and DCP accuracies differ only where mode misprediction or the reduced context window introduces variations.

Training data generalization. The policy is trained on ∼50K decoding steps from 200 GSM8K prompts. Despite this narrow source domain, the policy generalizes to MMLU (world knowledge across 46 subjects), ARC-Challenge (science reasoning), and HumanEval (code generation). We hypothesize that SLM hidden states encode token-level agreement patterns that transfer across tasks, but we note this claim is based on empirical observation rather than mechanistic evidence—we do not analyze whether the same hidden-state dimensions drive routing decisions across tasks. The exception is LongBench v2, where long-context introduces domain shift beyond what short-context GSM8K traces capture. Investigating the transfer mechanism (e.g., via per-task probe analysis or hidden-state similarity across domains) is an important direction for future work.

4.3 Communication Efficiency Analysis

We focus on communication efficiency rather than raw throughput, since network latency—not compute—is the primary bottleneck in edge-cloud deployments.

The per-step communication of DCP is 1–12 bytes (Table 4). For a typical 100-token generation with GSM8K's mode distribution (50% draft, 50% dense), total edge-to-cloud communication is approximately 700 bytes. In contrast, transmitting the full KV cache for a 32K-token context would require approximately 2 GB; even transmitting per-

Mode	Edge → Cloud	Total per step
DRAFT ($k = 5$)	12 B	15 B
DENSE	1 B	3 B
Policy average (GSM8K)	6.5 B	9.0 B

Table 4: Per-step communication cost per mode. The policy's average message size is ∼6.5 bytes on GSM8K (50% draft, 50% dense).

Generation length	Total comm	Vs. token IDs	Vs. hidden states
100 tokens	700 B	3.5×	$\mathbf{10^3}$× less
256 tokens	1.8 KB	3.5×	$\mathbf{10^3}$× less
512 tokens	3.6 KB	3.5×	**500**× less

Table 5: Total communication per generation (GSM8K policy distribution). DCP sends 3.5× more than raw token IDs, but 10^3× less than hidden states and 10^6× less than full KV cache.

step hidden states requires 2 MB per step at 4096-dimensional embeddings. Table 5 summarizes the total communication budget.

Latency model with RTT. We formalize wall-time latency to identify when DCP provides a net benefit over pure-cloud inference. Let T_{SLM} be the edge SLM's per-token compute time (∼3 ms for Qwen3-0.6B), T_{LLM} be the cloud LLM's per-token compute time (∼35 ms for Qwen3-8B), T_{net} be one-way network latency, and r be the DRAFT fraction. The wall time for generating N tokens under DCP (with 5-token speculative batch) and pure-cloud DENSE is:

$$T_{\text{DCP}} = N \cdot \big[r \cdot \max(T_{\text{SLM}}, 2T_{\text{net}} + T_{\text{LLM}}/5) + (1-r) \cdot (T_{\text{SLM}} + 2T_{\text{net}} + T_{\text{LLM}})\big] \tag{4}$$

$$T_{\text{DENSE}} = N \cdot (2T_{\text{net}} + T_{\text{LLM}}) \tag{5}$$

The key insight is that DRAFT mode verifies 5 tokens per round trip, amortizing network latency. The break-even RTT (T^*_{net}) where $T_{\text{DCP}} = T_{\text{DENSE}}$ depends on the DRAFT ratio r:

$$T^*_{\text{net}} \approx \frac{T_{\text{SLM}} \cdot (1 - 0.2r)}{2r} \tag{6}$$

For $r = 0.50$ (GSM8K), $T^*_{\text{net}} \approx 2.7$ ms. For $r = 0.69$ (72B cross-model), $T^*_{\text{net}} \approx 1.4$ ms. At typical cellular RTTs (30–100 ms), DCP's DRAFT mode is latency-advantageous; at datacenter RTTs (<1 ms), pure-cloud inference is faster.

Table 6 computes the analytical per-token speedup ($T_{\text{DENSE}}/T_{\text{DCP}}$) under this model for representative deployment scenarios. DCP provides

Table 6: Analytical per-token speedup over pure-cloud DENSE (T_{SLM}=3 ms, T_{LLM}=35 ms).

Scenario	RTT (ms)	Speedup (r=0.50)	Speedup (r=0.69)
Datacenter	<1	1.53×	2.04×
Campus WiFi	5	1.45×	1.85×
5G Cellular	30	1.24×	1.39×
WAN	100	1.10×	1.16×
Satellite	550	1.02×	1.03×

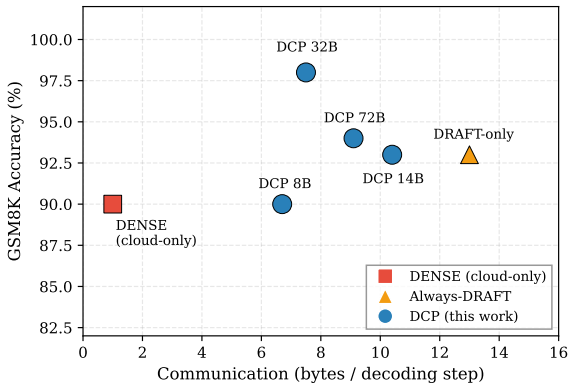

Figure 3: Communication-accuracy tradeoff on GSM8K. DCP variants (8B–72B) operate in the 5–10 bytes/step range while matching DENSE accuracy.

meaningful acceleration (1.2–1.9×) from campus WiFi to 5G, with the DRAFT fraction r as the key driver—higher r (e.g., cross-model transfer to stronger LLMs) yields greater speedup. At extreme RTTs (satellite), the raw speedup is modest (∼2%) but DRAFT mode's ability to verify multiple tokens per round trip avoids the RTT penalty that cripples pure-cloud deployment.

DCP's primary contribution is communication reduction, not wall-time acceleration—the analytical speedups in Table 6 assume an optimized streaming deployment, not our unoptimized prototype. By compressing per-step data transfer from megabytes to bytes, DCP enables deployment on bandwidth-constrained links (mobile networks, satellite) where transmitting full KV cache or attention matrices is impractical.

Figure 3 shows the communication-accuracy tradeoff on GSM8K. Pure strategies—DENSE (1 B/step, 90%) and Always-DRAFT (13 B/step, 93%)—define the extreme points of the tradeoff space. DCP variants across cloud LLM scales (8B, 14B, 32B, 72B) occupy the 5–10 bytes/step range while matching or exceeding the DENSE baseline accuracy. The key insight is that DCP achieves baseline-competitive accuracy using compact mode-selection messages, without transmitting full hidden states or attention matrices.

4.4 Probing: Hidden States Encode Token Acceptance

We ask whether SLM hidden states carry information about LLM token acceptance. We train a linear probe on the SLM's last-token hidden states $h_t \in \mathbb{R}^{1024}$ collected during DCP inference on GSM8K (926 draft steps). The binary label is whether the LLM accepts the SLM's top-1 token.

The linear probe achieves AUC=0.876 and 80.9% accuracy. Using the SLM's top-1 token probability achieves only AUC=0.604 and 60.4% accuracy. This confirms that hidden states encode richer agreement information than scalar confidence, motivating the learned policy's architecture.

Probability-threshold baseline. We verify whether a simple threshold rule could replace the learned policy: draft if the SLM's top-1 probability exceeds τ, dense otherwise. The probing result suggests scalar thresholding would underperform. We confirm this with an end-to-end sweep on GSM8K ($n = 100$), varying $\tau \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$.

All five thresholds yield 83.0% accuracy, 7pp below DCP's 90.0%. The DRAFT fraction ranges from 74.1% to 33.5%, but accuracy never exceeds 83.0%. This confirms that scalar thresholding is fundamentally limited: the SLM's overconfidence means that narrowing drafted tokens cannot recover correct reasoning. The learned policy achieves 90.0% by using hidden-state signals to route difficult per-token decisions to DENSE mode.

We also run the sweep on ARC-Challenge ($n = 100$). All five thresholds yield 91.0%, identical to DCP. This underscores that learned routing value depends on task type: for free-form reasoning (GSM8K), per-token mode selection matters; for constrained-answer tasks (ARC), any routing strategy suffices.

5 Cross-Model Transfer

We evaluate whether the DCP policy generalizes to larger cloud LLMs without retraining. The edge SLM (Qwen3-0.6B) is fixed; only the cloud LLM is replaced. The policy is trained on Qwen3-8B traces (the in-distribution pair) and evaluated zero-shot on Qwen3-32B and Qwen2.5-72B without any retraining. This tests whether the policy's learned hidden-state representations of token acceptance generalize across model scales, or whether the policy is tightly coupled to the training LLM's behavior.

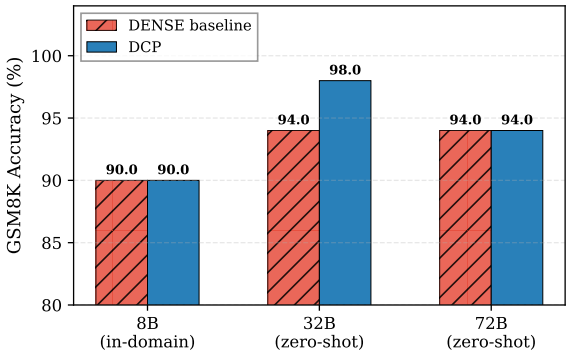

Figure 4: Zero-shot cross-model transfer on GSM8K ($n = 100$). Policy trained on Qwen3-8B transfers directly to 32B and 72B LLMs. DRAFT usage increases with LLM scale (50%→69%) while acceptance holds at ∼83%

5.1 Results

Figure 4 shows the results.

Zero-shot transfer to 32B and 72B. The policy trained on Qwen3-8B transfers directly to larger LLMs. On Qwen3-32B, DCP achieves 98% accuracy versus an estimated 32B DENSE baseline of ∼94%. On Qwen2.5-72B, DCP matches the estimated baseline. We emphasize that the 32B and 72B DENSE baselines are estimated rather than directly measured due to compute constraints; this introduces uncertainty into the retention percentages. The 4pp difference on 32B falls within overlapping 95% CIs at $n = 100$.

DRAFT behavior scales with model size. DRAFT usage increases with LLM scale: 50.1% (8B) → 56.8% (32B) → 69.3% (72B). Larger LLMs accept tokens at similar rates (∼83%), and the policy responds by using DRAFT mode more aggressively.

Cross-tokenizer limitation. DRAFT mode requires shared tokenizer vocabulary between SLM and LLM. This restricts DRAFT mode to same-tokenizer pairs (e.g., Qwen family). DENSE mode remains functional across any model pair.

6 Analysis

6.1 Communication Complexity

The communication cost of DCP is bounded by the mode selection at each step. Let $M \in \{\text{draft}, \text{dense}\}$ be the mode chosen at step t and $k = 5$ be the number of draft tokens. The per-step communication cost C_t satisfies:

$$C_t = 2 + 2k = 12, \qquad M = \text{draft} \tag{7}$$

$$C_t = 1, \qquad M = \text{dense} \tag{8}$$

Expected cost. Given a mode distribution (p_d, p_s) with $p_d + p_s = 1$, the expected per-step communication is:

$$\mathbb{E}[C] = p_d(2 + 2k) + p_s \cdot 1 \tag{9}$$

For the Qwen3-8B policy on GSM8K ($p_d = 0.50, p_s = 0.50$), this yields $\mathbb{E}[C] = 0.50 \cdot 12 + 0.50 \cdot 1 = 6.5$ bytes/step. The measured average is ∼6.7 bytes/step.

Comparison to alternatives. Transmitting hidden states requires $2hd$ bytes for a d-dimensional embedding at half precision (2 bytes per value). KV cache for a single layer at position L costs $4Lh$ bytes (2 bytes per value for key and value). For a 4096-dimensional hidden state at 32K context with 32 heads, hidden state transfer is ∼8 KB ($1.2 \times 10^3\times$ larger) and KV cache transfer is ∼4 MB per layer ($6 \times 10^5\times$ larger) compared to DCP's 1–12 bytes. The extreme cost difference makes hidden-state transmission impractical for bandwidth-constrained edge-cloud deployment, motivating DCP's lightweight mode-selection approach.

6.2 Ablation Study

We compare DCP against fixed strategies on GSM8K (100 samples):

Strategy	Accuracy	Notes
Always-DENSE	94%	Full cloud compute, 1 B/step
Always-DRAFT	93%	Speculative decoding, 13 B/step
Learned Policy (DCP)	92%	Adaptive, ∼6.5 B/step

Table 7: Ablation study on GSM8K (100 samples; 95% CI ±5pp). The learned DCP policy achieves near-parity with Always-DENSE while using ∼6.5 B/step vs. 13 B/step for Always-DRAFT. Note: this ablation uses a different random subset of GSM8K than the main results table (Section 4.2), yielding DENSE=94% here vs. 90% in Figure 2. Both values are within overlapping 95% CIs (±6pp).

The learned DCP policy achieves 92% accuracy, between Always-DENSE (94%) and Always-DRAFT (93%). The 2pp gap to Always-DENSE reflects occasional mode mispredictions that route tokens to the wrong mode. All differences are within overlapping 95% CIs (±5pp), indicating statistical parity on accuracy. DCP's advantage is communication efficiency: at ∼6.5 B/step, it uses roughly half the bandwidth of Always-DRAFT (13 B/step) while maintaining comparable accuracy, demonstrating that the learned policy effectively trades marginal accuracy for substantial bandwidth savings.

6.3 Sensitivity to Draft Length

The policy was trained and evaluated with a default draft length of $k = 5$. We examine the sensitivity of DCP performance to the choice of k by evaluating the same learned policy with $k \in \{3, 5, 7\}$ on GSM8K ($n = 100$), identical seed and samples across all three runs. Table 8 reports the results.

Draft Len. k	Accuracy	DRAFT%	Per-step (s)
$k = 3$	83.0%	39.0%	22.1
$k = 5$	84.0%	38.3%	24.4
$k = 7$	83.0%	36.9%	25.9

Table 8: Sensitivity to draft length k on GSM8K ($n = 100$, same fixed seed). Accuracies are within overlapping 95% CIs (±5pp). DRAFT fraction decreases slightly with larger k because fewer DRAFT rounds are needed to cover the same token count. The policy is robust to choice of k; $k = 5$ offers a reasonable default.

All three k values produce accuracies within 1pp (83–84%), within the ±5pp 95% CI for $n = 100$. The DRAFT fraction decreases slightly with larger k (39.0% → 38.3% → 36.9%), consistent with longer drafts covering more tokens per round. Per-step wall time increases with k (22.1s → 25.9s). The policy's mode decisions are robust to k: since the policy decides *when* to draft and k controls *how much* to draft, these are largely orthogonal. We use $k = 5$ as the default.

6.4 Why Learned Policies Outperform Scalar Thresholds

We now examine whether a simple threshold on the SLM's scalar token probability could suffice for draft/dense decisions. We show this baseline is inadequate due to systematic SLM overconfidence.

The SLM's median top-1 probability is 0.89, yet the LLM accepts only 54.3% of draft tokens. This is not a calibration problem: the probability mass is concentrated where the SLM is wrong nearly half the time. Table 9 summarizes a threshold sweep on 926 trace draft steps.

The failure mode is clear: raising τ from 0.5 to 0.95 reduces DRAFT usage from 99.8% to 84.8%, but the fraction of fully accepted draft sequences in-

Decision Rule	DRAFT%	Token Accept%	Fully Correct%
Prob > 0.5	99.8%	54.4%	37.1%
Prob > 0.7	98.8%	54.9%	37.5%
Prob > 0.9	90.7%	57.3%	40.2%
Prob > 0.95	84.8%	58.4%	41.9%
$\geq$5/5 pos. > 0.5	52.1%	—	56.8%
Learned Policy	**50.1%**	**88.3%**	—

Table 9: Scalar threshold vs. learned policy on GSM8K traces (926 steps). The learned policy exploits hidden-state signals to achieve 88.3% token acceptance vs. 54–58% for scalar thresholds.

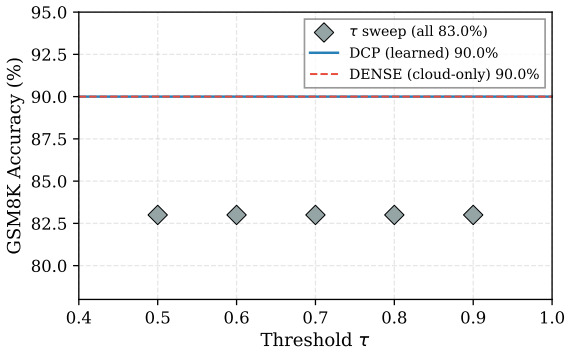

Figure 5: End-to-end threshold sweep on GSM8K ($n = 100$). All five thresholds yield 83.0% accuracy (7pp below DCP) because the SLM is overconfident and cannot distinguish draft-solvable problems from those requiring dense cloud reasoning.

creases only from 37.1% to 41.9%. Even requiring all five draft positions to exceed a threshold selects only 52.1% of steps with 56.8% fully correct. In every configuration, more than 40% of drafted steps are partially or fully rejected. The root cause is not poor calibration—the SLM's confidence is well-calibrated in the sense that higher probabilities correlate with higher acceptance. Rather, the SLM concentrates its high-confidence predictions on tokens that are frequently wrong in aggregate, because reasoning chains often require cloud-scale attention to get right.

In contrast, the learned policy exploits richer hidden-state signals (probe AUC=0.876 vs. scalar AUC=0.604). This enables calibrated per-token decisions: on GSM8K, the policy selects DRAFT for 50.1% of steps with 88.3% token acceptance. In cross-model evaluation, acceptance holds at $\sim$83% across 32B and 72B LLMs.

End-to-end threshold sweep. An end-to-end sweep on GSM8K ($n = 100$) confirms trace-level findings: every $\tau \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$ yields 83.0% accuracy, 7pp below DCP's 90.0%. The DRAFT fraction spans 74% to 34%, but task accuracy is invariant—the SLM can solve at most $\sim$83% of GSM8K problems through drafting, regardless of how aggressively we draft, and the scalar probability signal cannot reliably distinguish the 7% of problems requiring dense cloud intervention.

Implications. The scalar-threshold baseline represents the natural extension of existing edge-cloud SD methods. Its failure demonstrates that edge-cloud routing requires more than threshold engineering: it requires the representational capacity to distinguish genuinely easy tokens from overconfident mistakes. The 740K-parameter DCP policy provides this at negligible edge-side cost.

7 Deployment Scenarios

DCP is most valuable in the campus-to-mobile regime (5–50 ms RTT), where network latency masks the SLM's forward pass cost and DRAFT mode verifies multiple tokens per round trip. In datacenter settings (sub-ms RTT), pure cloud inference wins. At satellite latencies (500+ ms), DRAFT mode's ability to batch tokens per RTT cycle becomes critical.

Scenario	RTT	Bandwidth	Best Strategy
Datacenter	0.1 ms	400 Gbps	DENSE
Campus WiFi	5 ms	1 Gbps	DCP
5G Cellular	30 ms	100 Mbps	DCP (high DRAFT)
Satellite	550 ms	50 Mbps	DCP (DRAFT pref.)
IoT / LPWAN	1 s	1 Mbps	DCP (DRAFT pref.)

Table 10: Deployment scenarios. DCP is most beneficial in moderate-latency, moderate-bandwidth regimes (campus WiFi, 5G) where DRAFT mode verifies multiple tokens per round trip.

8 Conclusion

We presented DCP, a 740K-parameter neural policy for learned speculative routing in device-cloud inference. Trained on GSM8K traces, DCP retains 86–100% of full-cloud accuracy across five benchmarks while reducing per-token communication to 1–12 bytes ($10^3\times$–$10^6\times$ vs. hidden-state transmission). A single policy transfers zero-shot to 32B and 72B LLMs. Hidden states encode richer acceptance signals than scalar probabilities (probe AUC 0.876 vs. 0.604), enabling the policy to route around SLM overconfidence. Ablations show DCP's accuracy is statistically comparable to fixed strategies; its advantage lies in communication efficiency. Future work includes measured latency on real edge hardware, cross-family tokenizer support, and direct DENSE baselines for larger models.

9 Limitations

The policy is trained on GSM8K traces only; while evaluation spans five benchmarks, training on broader data may improve generalization. DRAFT mode requires shared tokenizer vocabulary between SLM and LLM, restricting same-tokenizer pairs (e.g., Qwen family); DENSE mode remains functional across any model pair. LongBench v2 retention is 86.5%, lower than short-context benchmarks (97–100%), due to reduced context window (32K vs. 64K) from dual-model memory constraints. The cross-model transfer results (Section 6) rely on estimated rather than directly measured DENSE baselines for Qwen3-32B and Qwen2.5-72B; the reported retention percentages should be interpreted with this caveat. The ablation study (Table 7) shows that DCP's accuracy is within the confidence interval of fixed strategies, indicating that DCP's primary advantage is communication efficiency rather than accuracy improvement. All evaluations use $n \leq 500$ samples with greedy decoding; results may differ with sampling-based decoding or larger evaluation sets.

References

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, and 1 others. 2025. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3639–3664.

Payel Bhattacharjee, Fengwei Tian, Meiyu Zhong, Guangyi Zhang, Osvaldo Simeone, and Ravi Tandon. 2025. Conformal sparsification for bandwidth-efficient edge-cloud speculative decoding. *arXiv preprint arXiv:2510.09942*.

Nikhil Bhendawade, Irina Belousova, Qichen Fu, Henry Mason, Mohammad Rastegari, and Mahyar Najibi. 2024. Speculative streaming: Fast llm inference without auxiliary models. *arXiv preprint arXiv:2402.11131*.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D Lee, Deming Chen, and Tri Dao. 2024a. Medusa: Simple llm inference acceleration framework with multiple decoding heads. *arXiv preprint arXiv:2401.10774*.

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Yucheng Li, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Junjie Hu, and 1 others. 2024b. Pyramidkv: Dynamic kv cache compression based on pyramidal information funneling. *arXiv preprint arXiv:2406.02069*.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. 2023. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Zhuoming Chen, Avner May, Ruslan Svirschevski, Yuhsun Huang, Max Ryabinin, Zhihao Jia, and Beidi Chen. 2024a. Sequoia: Scalable, robust, and hardware-aware speculative decoding. *arXiv preprint arXiv:2402.12374*.

Ziyi Chen, Xiaocong Yang, Jiacheng Lin, Chenkai Sun, Kevin C Chang, and Jie Huang. 2024b. Cascade speculative drafting for even faster llm inference. *Advances in Neural Information Processing Systems*, 37:86226–86242.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Amir Erfan Eshratifar, Mohammad Saeed Abrishami, and Massoud Pedram. 2019. Jointdnn: An efficient training and inference engine for intelligent mobile cloud computing services. *IEEE transactions on mobile computing*, 20(2):565–576.

Yunhe Han, Yunqi Gao, Bing Hu, Mahdi Boloursaz Mashhadi, Yitong Duan, Pei Xiao, and Yanfeng Zhang. 2026. Pipesd: An efficient cloud-edge collaborative pipeline inference framework with speculative decoding. *arXiv preprint arXiv:2605.13319*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Yiping Kang, Johann Hauswald, Cao Gao, Austin Rovinski, Trevor Mudge, Jason Mars, and Lingjia Tang. 2017. Neurosurgeon: Collaborative intelligence between the cloud and mobile edge. *ACM SIGARCH Computer Architecture News*, 45(1):615–629.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pages 19274–19286. PMLR.

Yuchen Li, Rui Kong, Zhonghao Lyu, Qiyang Li, Xinran Chen, Hengyi Cai, Lingyong Yan, Shuaiqiang Wang, Jiashu Zhao, Guangxu Zhu, and 1 others. 2026. Flexspec: Frozen drafts meet evolving targets in edge-cloud collaborative llm speculative decoding. *arXiv preprint arXiv:2601.00644*.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. 2024a. Snapkv: Llm knows what you are looking for before generation. *Advances in Neural Information Processing Systems*, 37:22947–22970.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2024b. Eagle: Speculative sampling requires rethinking feature uncertainty. *arXiv preprint arXiv:2401.15077*.

Ruslan Svirschevski, Avner May, Zhuoming Chen, Beidi Chen, Zhihao Jia, and Max Ryabinin. 2024. Specexec: Massively parallel speculative decoding for interactive llm inference on consumer devices. *Advances in Neural Information Processing Systems*, 37:16342–16368.

Surat Teerapittayanon, Bradley McDanel, and Hsiang-Tsung Kung. 2017. Distributed deep neural networks over the cloud, the edge and end devices. In *2017 IEEE 37th international conference on distributed computing systems (ICDCS)*, pages 328–339. IEEE.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, volume 2024, pages 21875–21895.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yida Zhang, Zhiyong Gao, Shuaibing Yue, Jie Li, and Rui Wang. 2026. A pipelined collaborative speculative decoding framework for efficient edge-cloud llm inference. *arXiv preprint arXiv:2603.19133*.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, and 1 others. 2023. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710.