Disentangling Topology and Diversity in Multi-Agent LLMs for Multilingual Low-Resource Emotion Detection

Ulugbek Shernazarov[1] **Charitha Ruwansiri Weerakon Basnayake**[2]
Abdelkhaleq El Jarjini[1] **Noel Crespi**[1] **Praboda Rajapaksha**[1,3]
[1]Samovar, Telecom SudParis, Institut Polytechnique de Paris, 91120 Palaiseau, France
[2]Cardiff Metropolitan University, Cardiff, United Kingdom
[3]Deparment of Computer Science, Aberystwyth University, Aberystwyth, United Kingdom
ulugbek.shernazarov@telecom-sudparis.eu

Abstract

Multi-agent LLM systems combine multiple inference calls, but prior work often confounds how calls are connected with how they are diversified. We study these factors independently: inference topology and source of inter-agent diversity. In a controlled 2×3 matrix, we cross parallel aggregation and sequential refinement with stochastic sampling, role prompting, and learned QLoRA specialization, under a fixed three-call budget and output protocol within each backbone. Using Qwen2.5-14B-Instruct and Llama-3.1-8B-Instruct, we evaluate all six configurations on multilingual low-resource emotion detection across nine languages. Parallel learned specialization is strongest on Qwen at 52.83 Macro-F1 and reaches 52.94 on Llama. On Qwen it also exceeds same-backbone zero-shot, few-shot, CoT, and seven-call self-consistency baselines. The preferred topology depends on diversity source: sequential refinement helps stochastic and prompted settings, while the learned Width advantage shrinks from 2.83 points on Qwen to 0.17 on Llama. Depth-wise analysis suggests that later learned specialists can overwrite correct early predictions, although the aggregate effect is backbone-dependent. Overall, how agents are differentiated produces larger performance shifts than topology, which should be evaluated jointly with specialization.

1 Introduction

Multi-agent LLM systems improve prediction quality by allocating multiple inference calls to the same input. A system may sample several independent answers and vote, assign different roles to parallel agents, or pass an answer through a sequence of critique and calibration steps (Wang et al., 2023; Du et al., 2024; Madaan et al., 2023; Shinn et al., 2023; Wang et al., 2025). These systems are often grouped under the same "multi-agent" label, but they encode different design choices about why multiple calls help. One choice concerns *topology*: should calls be connected in parallel, so that independent errors can cancel under aggregation, or in sequence, so that later calls can revise earlier reasoning? A second choice concerns *diversity*: do agents differ through stochastic decoding, prompt variation, or trained parameters?

This conflation makes reported gains difficult to interpret. Parallel systems often combine aggregation with sampling variation, role prompting, or heterogeneous models, while sequential systems typically vary both feedback structure and role design. As a result, it is often unclear whether improvements arise from the way calls are connected, from the source of inter-agent diversity, or simply from the strength of the underlying model. The distinction matters especially under a fixed inference budget, where one must decide whether additional calls are better spent on voting, revision, prompt engineering or learned specialization.

Fine-grained emotion detection is a useful setting for this comparison because the task separates evidence gathering from decision aggregation. Emotional meaning may depend on lexical cues, emojis, sarcasm, negation, pragmatic context, and multiple co-occurring emotions. These cases make topology and diversity testable: different calls can attend to different evidence while the final output remains a fixed multi-label decision. Recent multi-agent work on emotion analysis instantiated a single coordination architecture rather than isolating which design choice is responsible for the gain (Dong et al., 2026). We study this question through a controlled 2×3 matrix, (Figure 1 and Section 3). The topology axis compares *Width* - three agents predict independently and are aggregated in parallel, with *Depth* - three calls form a sequential refinement chain. We instantiate the complete six-cell matrix independently on Qwen2.5-14B-Instruct (Yang et al., 2024) and Llama-3.1-8B-Instruct (Grattafiori et al., 2024). Within each

The Proceedings of the Second Workshop for REsearch on Agent Language Models (REALM 2026), pages 1340–1362
October 29, 2026 ©2026 Association for Computational Linguistics

backbone, we hold the model, call budget, output schema, and reasoning protocol fixed, allowing topology and diversity source to be studied as separate experimental factors rather than entangled components of a single architecture.

We evaluate this design on multilingual low-resource emotion detection (Muhammad et al., 2025a) across nine languages. The setting also exposes a failure mode that English-only evaluation can obscure: when the base model has weak language competence, parallel voting among weak agents may reproduce shared errors or abstention rather than create useful diversity. The same fixed six-label task is evaluated across languages with widely different base-model competence, creating a natural stress test for whether additional calls contribute complementary evidence or merely repeat shared cross-lingual failures. Contributions in this paper are within a two-backbone, fixed three-call setting. First, it provides a controlled comparison of topology and diversity source in multi-agent LLM inference under fixed compute within each backbone. Second, it shows across Qwen2.5-14B and Llama-3.1-8B that learned specialization via lightweight adapters is substantially more effective than stochastic or prompt-based diversification, while explicitly separating the effect of pooled supervised adaptation from specialist heterogeneity. Third, it gives a mechanistic account of when sequential refinement helps and when it hurts: depth is useful when later calls can add evidence to weak initial predictions, while the learned Width–Depth ordering is model-dependent and later correction-trained specialists require stronger preservation-aware calibration once the first specialist is already strong.

2 Related Work

Multi-agent LLM inference. Multi-agent LLM systems improve predictions by making multiple calls to the same or related inputs, but prior work often changes the connection pattern and the source of diversity at the same time. Chain-of-thought prompting elicits intermediate reasoning (Wei et al., 2022; Kojima et al., 2022); self-consistency samples several reasoning traces and aggregates final answers (Wang et al., 2023), while debate and mixture-of-agents systems combine critique, aggregation, proposer identity, or heterogeneous models (Du et al., 2024; Wang et al., 2025). Sequential methods such as Self-Refine and Reflexion instead revise a single trajectory through feedback or reflection (Madaan et al., 2023; Shinn et al., 2023). These systems show that multiple calls can help, but they do not isolate whether gains come from topology, from diversity among calls, or from model strength. This distinction is important because later calls are not automatically beneficial: self-correction can degrade performance when feedback is weak or uninformative (Huang et al., 2024). Our work addresses this confound by crossing topology and diversity source while holding the backbone, call budget, output schema, and evaluation setting fixed.

Sources of diversity. Diversity in multi-call inference can be induced by stochastic decoding, by prompts or roles, or by trained parameters. Stochastic diversity underlies self-consistency; prompt and role diversity are common in persona-based and collaborative prompting (Kong et al., 2024; Salewski et al., 2023; Kim et al., 2025); and parameter diversity can be introduced efficiently through LoRA/QLoRA adapters (Hu et al., 2022; Dettmers et al., 2023). Ensemble theory suggests that aggregation is useful only when members are both individually competent and meaningfully diverse (Krogh and Vedelsby, 1994; Wood et al., 2023). This motivates our comparison of stochastic, designed and learned diversity under the same three-call budget. In particular, the L3 condition combines parameter diversity with pooled supervised specialization and may therefore improve individual competence as well as inter-agent heterogeneity, whereas prompt-only diversity may mainly vary outputs around the same shared errors. We use the term *learned specialization* when attributing the source of the L3 gain.

Multilingual low-resource emotion detection. Fine-grained emotion detection is a natural testbed for this question because emotional meaning can depend on lexical cues, pragmatic context, sarcasm, script, and culturally specific affective expressions. Multilingual emotion benchmarks such as BRIGHTER extend earlier African and multilingual sentiment resources to multi-label emotion classification (Muhammad et al., 2022, 2023). Performance in this setting is shaped by typology, script overlap, pretraining coverage, and resource level (Pires et al., 2019; Conneau et al., 2020; Wu and Dredze, 2020; Lin et al., 2019; Joshi et al., 2020). Recent work further suggests that multilingual LLM reasoning may transfer unevenly and

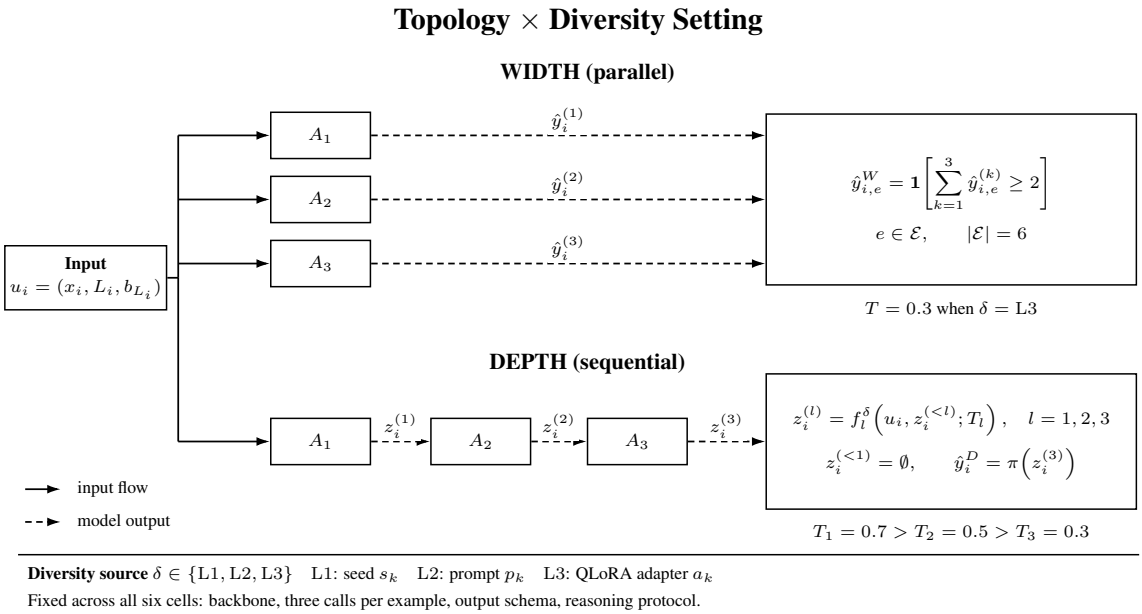

Figure 1: Topology × diversity framework. Width issues three parallel predictions and combines them by per-label majority vote; Depth issues three sequential refinement steps and reads out the final state. The diversity source δ varies across stochastic, prompted, and learned conditions, while the backbone, the three-call budget, the output schema, and the reasoning protocol are held fixed in every cell.

may route through English even for non-English inputs (Shi et al., 2023; Etxaniz et al., 2024). These conditions make low-resource emotion detection useful for testing whether multi-agent diversity creates genuinely complementary evidence or merely repeats shared cross-lingual failure modes.

Multi-agent emotion reasoning and our gap. The closest multi-agent emotion-reasoning work proposes specialized agents for perception, reasoning, and resolution. That line of work motivates modular affective reasoning, but it studies a fixed coordination design rather than separating topology from diversity source. Similarly, prompting work studies role variation without learned parameter diversity, while adapter work usually evaluates specialization under one aggregation regime. The gap is therefore not the absence of multi-call inference, prompts, or adapters; it is the absence of a controlled comparison that asks whether stochastic, prompted, and learned diversity should be aggregated in parallel or chained sequentially under the same inference budget.

3 Methodology

3.1 Task and Evaluation Setting

We formulate emotion detection as multi-label classification over the six BRIGHTER emotions: ANGER, DISGUST, FEAR, JOY, SADNESS, and SURPRISE. Let $\mathcal{E}$ denote this label set, with $|\mathcal{E}| = 6$. Each example is a triple (x_i, L_i, y_i), where x_i is the input text, L_i is the language identifier, and $y_i \in \{0,1\}^{|\mathcal{E}|}$ is the gold multi-label emotion vector. Each system emits a JSON object containing one binary decision per emotion and a short reasoning field.

The headline metric is Macro-F1 averaged uniformly over languages and emotions. For language L and emotion e, let $\mathrm{F1}_{L,e}$ be the binary F1 score for emotion e on examples from language L. We report

$$\mathrm{MacroF1} = \frac{1}{|\mathcal{L}|\,|\mathcal{E}|} \sum_{L \in \mathcal{L}} \sum_{e \in \mathcal{E}} \mathrm{F1}_{L,e}$$

Uniform language averaging prevents languages with larger test sets from dominating the aggregate score and makes language-level failures visible.

We evaluate on the nine-language low-resource BRIGHTER subset: Naija Pidgin, Mandarin Chinese, Marathi, Moroccan Arabic, Tatar, Emakhuwa, Mozambican Portuguese, Zulu, and Indonesian. We use the standard splits: 11,214 public training instances pooled over the seven languages with public training data, 1,975 development instances, and 8,597 held-out test instances (Muhammad et al., 2025b). All systems are evaluated as a single pooled multilingual system. L1/L2 and the prompting baselines are inference-only, whereas

L3 adapters are trained once per backbone on the pooled public training set; no model is selected or tuned separately per target language. Within each backbone, all matrix cells use the same output schema, reasoning-bank policy, and decision protocol. Additional dataset, split, and reasoning-bank details are given in Appendix B.

3.2 Topology–Diversity Matrix

We study two topologies: *Width* (parallel aggregation) and *Depth* (sequential refinement), with three diversity sources: stochastic sampling (L1), designed prompts (L2), and learned specialization (L3), yielding the six cells in Figure 1.

We instantiate the full matrix independently on Qwen2.5-14B-Instruct and Llama-3.1-8B-Instruct, served with vLLM (Kwon et al., 2023). Within each backbone, every cell makes exactly three inference calls per input, uses top-$p = 0.95$, and shares the same output schema and decision protocol. Each call receives the same shared input package $u_i = (x_i, L_i, b_{L_i})$ where b_{L_i} is the language-specific reasoning bank for language L_i. This fixed-budget design makes within-backbone comparisons interpretable: observed differences can be attributed to topology, diversity source or their interaction rather than to a different number of calls or model identity inside the matrix.

Width topology. In Width, three agents receive the same input package u_i and predict independently. For diversity level $\delta \in \{\mathrm{L1}, \mathrm{L2}, \mathrm{L3}\}$, the three agent predictions are

$$\hat{y}_i^{(k)} = f_k^{\delta}(u_i), \qquad k \in \{1, 2, 3\},$$

where $\hat{y}_i^{(k)} \in \{0, 1\}^{|\mathcal{E}|}$. For each emotion label, the final prediction is obtained by majority vote:

$$\hat{y}_{i,e}^{W} = \mathbf{1}\left[\sum_{k=1}^{3} \hat{y}_{i,e}^{(k)} \geq 2\right], \qquad e \in \mathcal{E}.$$

Thus, a label is emitted iff at least two of the three agents emit it. Width has no inter-agent conditioning or ordering, and the three outputs are symmetrically aggregated.

Depth topology. In Depth, three layers are executed sequentially. Let $z_i^{(\ell)}$ denote the full JSON output of layer ℓ, including both the binary emotion decisions and the reasoning field, and let $\pi(z)$ extract the binary emotion vector from a JSON output. Layer 1 predicts from the input alone: $z_i^{(1)} = f_1^{\delta}(u_i; T_1)$. Layer 2 receives the original input together with Layer 1's full JSON output: $z_i^{(2)} = f_2^{\delta}(u_i, z_i^{(1)}; T_2)$. Layer 3 receives the original input and the JSON outputs from both previous layers: $z_i^{(3)} = f_3^{\delta}(u_i, z_i^{(1)}, z_i^{(2)}; T_3)$. The final Depth prediction is $\hat{y}_i^{D} = \pi(z_i^{(3)})$. The temperature schedule descends as $(T_1, T_2, T_3) = (0.7, 0.5, 0.3)$, so that the amount of decoding randomness matches the layer's role in the chain. The first layer uses the highest temperature to allow alternative interpretations of ambiguous affective cues; the second layer reduces randomness while reviewing the first output; and the third layer uses the lowest temperature to make the final prediction more conservative and stable. This implements an exploration-to-consolidation pattern: early layers may explore alternative interpretations, while later layers are encouraged to revise and commit.

3.3 Diversity Sources

L1: Stochastic diversity. Calls share the generic CoT prompt and differ only through decoding stochasticity and, for Depth, sequential conditioning. Width-L1 uses seeds 42, 123, and 456 at $T = 0.7$; Depth-L1 uses the shared prompt under the $0.7/0.5/0.3$ schedule. Seeds are fixed a priori rather than selected by development performance.

L2: Designed prompt diversity. Width-L2 uses three additive analytical lenses emphasizing surface, contextual, and discourse-structural cues at $T = 0.7$ and top-$p = 0.95$. Depth-L2 instead assigns sequential ANALYST, CRITIC, and CALIBRATOR roles. The prompts bias attention without restricting agents from using other evidence types; full templates are given in Appendices K–L.

L3: Learned specialization (parameter diversity). L3 trains three QLoRA adapters over each frozen backbone. Unlike L1/L2, L3 introduces pooled supervised adaptation in addition to inter-agent heterogeneity, so the raw L3–L1/L2 difference cannot be attributed to diversity alone. Section 5.5 separates these effects using a shared-adapter control and a temperature-matched specialist control.

Width-L3 uses general, ambiguity, and contrastive specialists followed by the same majority vote used throughout Width. Depth-L3 uses position-specific analyst, critic, and calibrator adapters trained for successive correction. Adapters are hot-swapped through one vLLM en-

gine; objectives and QLoRA details are provided in Appendix M.

3.4 Shared Protocol and Ablations

Within each backbone, all six cells share the same language-specific reasoning banks, precision-prior decision policy, JSON schema, and three-call budget. This prevents language conditioning, thresholding, or formatting differences from being mistaken for topology or diversity effects.

We test shared-adapter controls, temperature and prompt variants, specialist leave-one-out, alternative aggregation, paired-bootstrap uncertainty, cross-backbone replication, and Depth truncation. The complete ablation suite is reported in Appendix G.

4 Experiments

We evaluate the complete six-cell matrix on the same nine-language BRIGHTER development and held-out test splits for Qwen2.5-14B-Instruct and Llama-3.1-8B-Instruct. Topology, prompts, reasoning banks, adapter objectives, output schema, and the three-call budget are unchanged across backbones.

The Qwen reference systems are zero-shot, few-shot, single-pass CoT, and seven-call CoT with self-consistency. Qwen2.5-72B and multilingual encoder results are reported only as contextual BRIGHTER references.

For key contrasts, we compute 95% paired non-parametric bootstrap intervals using 2,000 shared resamples. For the learned Width–Depth comparison we additionally report a Wilcoxon signed-rank test over the nine paired language scores. Bootstrap intervals measure evaluation-sample uncertainty; targeted W-L3 seed perturbations separately assess sensitivity to training and inference stochasticity. Full compute and configuration details appear in Appendix C.

5 Results

5.1 Topology and Diversity Effects

Table 1 reports the complete Topology $\times$ Diversity matrix on both backbones. On Qwen2.5-14B, W-L3 reaches 53.61% Macro-F1 on development and 52.83% on test, and replacing designed prompt diversity (W-L2, 40.71%) with learned specialization improves test Macro-F1 by 12.12 points. The second backbone preserves the main diversity-source result. On Llama-3.1-8B, W-L3 reaches 52.94% test Macro-F1, exceeding W-L1 by 8.39 points and W-L2 by 7.85 points. Thus, the largest shifts in both matrices come from moving to the learned-specialization regime rather than from changing topology.

The topology effect remains smaller and depends on the source of diversity. On Qwen, Depth outperforms Width for stochastic and designed diversity by 3.22 and 4.36 test points, while W-L3 exceeds D-L3 by 2.83 points. Llama shows the same directional topology advantage at L1 and L2, with Depth exceeding Width by 2.47 and 1.26 points, respectively, but the learned cells are nearly tied: W-L3 exceeds D-L3 by only 0.17 points. Therefore, no topology dominates across diversity sources, and the learned Width advantage observed on Qwen should not be treated as a backbone-independent rule.

The near-zero W-L1 $\rightarrow$W-L2 improvement on Qwen is also informative, but it should not be read as evidence that designed prompts contain no useful signal. Under the canonical majority-vote rule, designed analytical-lens prompts add only 0.05 test points, from 40.66% for W-L1 to 40.71% for W-L2. However, the original aggregation ablation shows that W-L2 rises to 45.12% under confidence-max aggregation, a gain of 4.41 points over majority vote (Appendix Table 13). Prompt diversity therefore creates recoverable signal even when majority vote underuses it. The corresponding Llama Width result is likewise small under majority vote (45.09 versus 44.55), reinforcing that prompt-only diversity changes agent behavior much less than learned specialization.

This is especially plausible in the multilingual setting: surface, context, and structure lenses can redirect attention, but they cannot by themselves add missing language competence when the underlying model parameters remain unchanged. Learned specialization changes this regime by improving the competence and error profile of the individual agents before aggregation, as examined directly in Section 5.5.

5.2 Cross-Backbone Replication

The second backbone separates findings that replicate from those that are model-dependent. The learned-specialization advantage over stochastic and designed Width is large on both models: W-L3 $-$W-L1 is +12.17 points on Qwen and +8.39 on Llama, while W-L3 $-$W-L2 is +12.12 and +7.85 points, respectively. Sequential conditioning also

Topology	Diversity	Qwen2.5-14B		Llama-3.1-8B	
		Dev	Test	Dev	Test
WIDTH	L1 Stochastic	42.00	40.66	45.97	44.55
WIDTH	L2 Designed	42.40	40.71	46.91	45.09
WIDTH	L3 Learned	**53.61**	**52.83**	53.45	**52.94**
DEPTH	L1 Stochastic	44.61	43.88	48.84	47.02
DEPTH	L2 Designed	45.69	45.07	48.52	46.35
DEPTH	L3 Learned	50.74	50.00	**53.78**	52.77

Table 1: Topology $\times$ Diversity matrix averages over nine languages (Macro-F1, %) on two model families/sizes. Values are the canonical runs; targeted seed robustness is reported in Appendix E.

improves the stochastic cell on both backbones: D-L1 $-$W-L1 is +3.22 points on Qwen and +2.47 on Llama.

The learned Width–Depth ordering is less stable. On Qwen, the paired bootstrap difference W-L3 $-$D-L3 is +2.83 points with a 95% interval of [1.91, 3.77]; on Llama it is +0.17 with an interval of [-0.57, 0.94]. A language-level Wilcoxon signed-rank test is also non-significant for this contrast on both Qwen ($p = .10$) and Llama ($p = .73$). We therefore retain the Qwen learned-Width result as a descriptive finding of that backbone, but the cross-backbone evidence supports the narrower conclusion that specialization effects replicate more strongly than a specific learned-topology preference.

Seed robustness. We additionally perturb the stochastic components of the headline learned Width condition. On Qwen, changing the inference seed moves W-L3 test Macro-F1 from 52.83 to 53.38 ($\Delta = +0.55$); on Llama, an alternative QLoRA training seed moves it from 52.94 to 53.62 ($\Delta = +0.68$). These shifts are small relative to the 7.85–12.17 point learned-specialization gains over W-L1/W-L2. Appendix E reports the corresponding robustness results.

5.3 Comparison with Reference Systems

Table 2 compares the Qwen W-L3 result against the averaged reference systems used in our evaluation. The first group contains same-backbone no-multi-agent baselines. These are the primary Qwen comparisons because they share the Qwen2.5-14B-Instruct backbone and output protocol. The prompting baselines and L1/L2 cells are inference-only, whereas L3 uses pooled supervised adapter training; we therefore use these baselines to contextualize the gain rather than to attribute all of it to multi-agent diversity. W-L3 outperforms zero-shot prompting by 8.74 points, few-shot prompting by 8.60 points, single-pass CoT by 4.36 points and CoT+self-consistency at $k = 7$ by 4.02 points, while using three calls. This suggests that the gain is not simply from sampling more reasoning traces: CoT+self-consistency increases stochastic coverage around one prompt and one parameter setting, whereas W-L3 combines three learned specialists with different adapter priors before aggregation. The advantage is strongest as an averaged nine-language result rather than a claim of uniform dominance for every language and every comparator. In the matrix itself, W-L3 is the best cell on seven of the nine languages (Appendix Table 7); against SemEval contextual references, it exceeds the RoBERTa baseline on eight of nine languages but does not match the per-language Top-1 systems on most languages (Appendix Table 10).

Condition	Calls	Test F1	Δ
Zero-shot Qwen2.5-14B	1	44.09	+8.74
Few-shot Qwen2.5-14B	1	44.23	+8.60
CoT, single pass	1	48.47	+4.36
CoT+SC, $k = 7$	7	48.81	+4.02
Qwen2.5-72B few-shot	1	45.87	+6.96
XLM-R cross-lingual	–	33.27	+19.56
mDeBERTa cross-lingual	–	31.23	+21.60
W-L3 (ours)	3	**52.83**	0.00

Table 2: Averaged reference systems on the nine-language test split (Macro-F1, %). Δ is the advantage of W-L3 over each condition. The first block contains same-backbone no-multi-agent baselines; the second block contains external BRIGHTER references.

The second group in Table 2 contains BRIGHTER reference systems. W-L3 exceeds the Qwen2.5-72B few-shot reference by 6.96 points; this comparison is contextual because model scale, supervision, and inference protocol differ. It also exceeds the reported XLM-R and mDeBERTa cross-lingual encoder baselines by 19.56 and 21.60 points, respectively. These en-

coder comparisons are useful as context because they represent non-generative multilingual transfer systems from the same benchmark family. They should not be treated as controlled head-to-head evidence: the BRIGHTER references differ in supervision, model class, scale, and inference protocol, whereas the within-matrix and same-backbone Qwen comparisons remain the most interpretable evidence.

Appendix Table 10 gives the per-language SemEval-2025 Task 11 comparison. We treat it as contextual rather than conclusive because leaderboard systems may use target-language supervision, per-language model selection and larger heterogeneous ensembles. W-L3 is higher than the official RoBERTa baseline on eight of nine languages, with an average advantage of 8.07 points; the main performance gap is in Marathi, where supervised RoBERTa is 10.00 points higher. We use Dong et al. (2026) as reference for multi-agent emotion reasoning rather than as a direct Macro-F1 baseline, since its English benchmarks, model scale and supervision setting differ from this study.

5.4 Per-Language Patterns

The gains from learned Width are not uniform across languages. Relative to W-L2, the Qwen W-L3 improves Emakhuwa from 2.02% to 19.16% test Macro-F1 and Zulu from 13.69% to 27.05% (Appendix Table 7). These languages remain the lowest absolute scores, but the improvement is important because the prompt-only Width cells are dominated by all-zero abstention on the low-resource tail: prediction diagnostics show that W-L1 and W-L2 emit all-zero vectors for roughly 97% of Emakhuwa inputs and 86% of Zulu inputs. Learned specialization therefore improves performance in a regime where prompt-only diversity mainly reproduces shared abstention.

The gains are not limited to these two lowest-resource languages. On Qwen, W-L3 exceeds W-L1 on seven of nine languages; on Llama it exceeds W-L1 on all nine (Appendix Tables 7 and 8). Naija Pidgin, Moroccan Arabic, Tatar, Mozambican Portuguese, and Zulu show substantial gains on both backbones. Marathi remains an informative boundary case: the Qwen W-L3 result is 72.20%, while the per-language RoBERTa contextual reference reaches 82.20% (Appendix Table 10), illustrating the regime where target-language supervised encoder training can still dominate a unified pooled-adapter system.

Condition	Qwen	Llama
W-L1	40.66	44.55
Shared adapter, 3 seeds ($T = .7$)	50.04	49.08
Three specialist adapters ($T = .7$)	51.57	49.83
Canonical W-L3	**52.83**	**52.94**

Table 3: Staged test-set controls for the learned Width gain (Macro-F1, %).

A protocol-matched diagnostic analysis further tests whether the gain from learned specialization is related to base-model competence. Using the single-pass $T = 0.7$ CoT score as a per-language competence proxy, the W-L3 $-$W-L1 gain is negatively associated with competence on both backbones. The relation is strongest on Llama (Spearman $\rho = -0.767$, $p = .016$); Qwen shows the same direction ($\rho = -0.567$, $p = .112$). With only nine languages, this is supporting rather than definitive evidence, but it helps explain why BRIGHTER is diagnostic for the present question: the benchmark supplies a wide natural range of model competence while keeping the task and label space fixed.

5.5 Attributing the W-L3 Gain

The raw L3 gain combines supervised adaptation, specialist heterogeneity, and decoding temperature. We therefore evaluate the staged control in Table 3: W-L1, a shared adapted model with three stochastic calls at $T = 0.7$, three specialist adapters at the same temperature, and canonical W-L3 at $T = 0.3$.

Along this intervention path, Qwen gains $+9.38$ points from shared adaptation, $+1.53$ from specialist differentiation at matched temperature, and $+1.26$ from the canonical lower temperature. The corresponding Llama increments are $+4.53$, $+0.75$, and $+3.11$. Thus, the largest L3 improvement comes from supervised adaptation, with a smaller additional contribution from specialist heterogeneity. We therefore interpret L3 as a *learned-specialization* regime rather than a pure diversity intervention.

Additional Qwen controls support the same interpretation: raising W-L3 to $T = 0.7$ reduces test Macro-F1 to 51.57, removing the system prompt costs about 2.8 points, and adding specialist-specific prompts produces no meaningful improvement (Appendix G).

The one-call adapted comparison shows that individual specialists are already strong. Qwen specialists score 52.42, 50.06, and 51.75 versus 52.83

for the ensemble; Llama specialists score 52.33, 47.50, and 53.52 versus 52.94. Aggregation therefore contributes modest, backbone-dependent value rather than explaining the full L3 gain. Qwen leave-one-out re-aggregation additionally drops by 5.29, 4.78, and 2.01 points when removing the general, contrastive, and ambiguity specialists, respectively (Appendix G).

5.6 Learned Depth vs. Learned Width

The L3 reversal on Qwen is the clearest topology–specialization interaction, but the second backbone shows that its aggregate magnitude is model-dependent. Table 12 reports the Qwen truncation analysis. Depth itself is not generally harmful: for D-L1 and D-L2, the full chain improves over Layer 1 alone by 2.01 and 3.21 test points. The pattern reverses only for Qwen D-L3, where Layer 1 alone already reaches 52.11% and the full learned chain falls to 50.00%. This supports an implementation-specific over-correction interpretation rather than a general weakness of sequential refinement.

The aggregate degradation does not repeat on Llama: the learned Depth Layer-1 output is 52.33%, while the full chain reaches 52.77%. Nevertheless, the decision-transition analysis exposes a related over-revision tendency. From Layer 1 to Layer 2, Qwen fixes 2.47% of label decisions but breaks 3.46% (net -0.99 points at the decision level); Llama fixes 1.05% but breaks 2.46% (net -1.42). Thus, correction-trained later specialists overwrite correct early decisions on both backbones, but whether those changes reduce the final language-averaged Macro-F1 depends on the model and language distribution.

The Qwen skip-layer variant is also informative: removing Layer 2 scores 50.04%, nearly identical to the full D-L3 chain at 50.00%. Layer 2 is therefore not the unique bottleneck in Qwen. The combined evidence supports a narrow practical conclusion: sequential learned specialists need preservation-aware, position-aware calibration so that later layers learn when *not* to revise a strong predecessor. The data do not support a universal rule that learned specialists should always be run in parallel.

Post-hoc aggregation remains a diagnostic factor for prompt-only Width. In the Qwen aggregation run, replacing majority vote with confidence-max improves W-L1 from 40.66% to 44.39% and W-L2 from 40.71% to 45.12%, gains of 3.73 and 4.41 points, respectively. In contrast, W-L3 is comparatively stable under that readout. Together with the paired cross-backbone statistics in Section 5.2, these results reinforce the main distinction: learned specialization is the robust large effect, while topology and aggregation effects are smaller and conditional on how agent differences are created.

6 Conclusion

We set out to ask whether topology or diversity source matters more in multi-agent LLM inference. The central pattern is consistent: under a fixed three-call budget, changing how agents are differentiated produces substantially larger performance shifts than changing topology. On Qwen, moving from designed prompts to learned QLoRA specialists gains 12.12 Macro-F1 points; the corresponding gain on Llama is 7.85 points. The cross-backbone evidence, however, qualifies the topology claim. Sequential depth helps the stochastic and prompted regimes, while the learned Width–Depth ordering is clear on Qwen but nearly tied on Llama. The attribution controls further show that the L3 advantage is driven primarily by pooled supervised adaptation, with a smaller additional contribution from specialist heterogeneity. The practical upshot is therefore to invest first in how agents acquire complementary competence, and to evaluate topology jointly with that specialization rather than treating Width or Depth as universally preferable. Reporting multi-agent systems along these axes separately would make gains in this fast-moving area considerably easier to interpret and build on.

Limitations

The matrix is evaluated on two decoder-only instruction-tuned backbones, Qwen2.5-14B-Instruct and Llama-3.1-8B-Instruct, which does not establish model-independent design rules across architectures or scales. The study uses one task domain, BRIGHTER multilingual multi-label emotion detection; its variation in language competence is useful diagnostically, but the observed topology–specialization interactions may not transfer unchanged to unrelated agent tasks. L3 also differs from L1/L2 in supervision: the learned adapters use pooled public training data from the seven Track-A languages. The target-excluded reasoning-bank policy depends on imperfect proxies: Chinese and Tatar lack close in-pool siblings and, therefore, rely on broader cues such as English pretraining coverage, script overlap, contact, and

emoji anchors. Paired bootstrap intervals quantify evaluation-sample uncertainty rather than training-run variance; targeted W-L3 seed perturbations provide a separate sensitivity check and shift test Macro-F1 by 0.55 points on Qwen and 0.68 points on Llama. Emakhuwa also remains low even in the best Qwen cell (19.16% Macro-F1), showing that multi-agent inference cannot replace missing target-language competence.

Ethical considerations

Emotion detection can support accessibility and crisis-monitoring applications, but it can also enable surveillance, profiling or unfair moderation. These risks are amplified in low-resource languages, where errors are higher and cultural cues are easier to misread. The system should not be used as an individual decision-maker in high-stakes settings; deployment requires human review, uncertainty reporting and validation with language-community expertise. We use publicly released BRIGHTER annotations, collect no new human-subject data, report aggregate results, and avoid reproducing identifiable examples. AI-assisted tools were used for language and LaTeX/submission support; the authors checked all experiments, numerical results, and scientific claims. We use BRIGHTER in accordance with the dataset's stated license and intended research use.

Acknowledgments

We thank the anonymous ARR reviewers for constructive feedback that strengthened the cross-backbone and robustness analyses.

References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*.

Hao Dong, Zuowen Bao, Muze Li, and Zhengfeng Yang. 2026. Emotion meets coordination: Designing multi-agent LLMs for fine-grained user sentiment detection on social media. *PLOS One*, 21(2):e0342053.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multiagent debate. In *International Conference on Machine Learning*.

Julen Etxaniz, Gorka Azkune, Aitor Soroa, Oier Lopez de Lacalle, and Mikel Artetxe. 2024. Do multilingual language models think better in English? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 550–564, Mexico City, Mexico. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024. Large language models cannot self-correct reasoning yet. In *International Conference on Learning Representations*.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Junseok Kim, Nakyeong Yang, and Kyomin Jung. 2025. Persona is a double-edged sword: Rethinking the impact of role-play prompts in zero-shot reasoning tasks. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 848–862, Mumbai, India. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics. Findings.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*.

Aobo Kong, Shiwan Zhao, Hao Chen, Qicheng Li, Yong Qin, Ruiqi Sun, Xin Zhou, Enzhi Wang, and Xiaohang Dong. 2024. Better zero-shot reasoning with role-play prompting. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4099–4113, Mexico City, Mexico. Association for Computational Linguistics.

Anders Krogh and Jesper Vedelsby. 1994. Neural network ensembles, cross validation, and active learning. In *Advances in Neural Information Processing Systems*, volume 7.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the ACM SIGOPS Symposium on Operating Systems Principles*.

Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, Antonios Anastasopoulos, Patrick Littell, and Graham Neubig. 2019. Choosing transfer languages for cross-lingual learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3125–3135, Florence, Italy. Association for Computational Linguistics.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-Refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*.

Shamsuddeen Hassan Muhammad, Idris Abdulmumin, Seid Muhie Yimam, David Ifeoluwa Adelani, Ibrahim Said Ahmad, Nedjma Ousidhoum, Abinew Ali Ayele, Saif Mohammad, Meriem Beloucif, and Sebastian Ruder. 2023. SemEval-2023 task 12: Sentiment analysis for African languages (AfriSenti-SemEval). In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2319–2337, Toronto, Canada. Association for Computational Linguistics.

Shamsuddeen Hassan Muhammad, David Ifeoluwa Adelani, Sebastian Ruder, Ibrahim Sa'id Ahmad, Idris Abdulmumin, Bello Shehu Bello, Monojit Choudhury, Chris Chinenye Emezue, Saheed Salahudeen Abdullahi, Anuoluwapo Aremu, Alípio Jorge, and Pavel Brazdil. 2022. NaijaSenti: A Nigerian Twitter sentiment corpus for multilingual sentiment analysis. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 590–602, Marseille, France. European Language Resources Association.

Shamsuddeen Hassan Muhammad, Nedjma Ousidhoum, Idris Abdulmumin, Jan Philip Wahle, Terry Ruas, Meriem Beloucif, Christine de Kock, Nirmal Surange, Daniela Teodorescu, Ibrahim Said Ahmad, David Ifeoluwa Adelani, Alham Fikri Aji, Felermino D. M. A. Ali, Ilseyar Alimova, Vladimir Araujo, Nikolay Babakov, Naomi Baes, Ana-Maria Bucur, Andiswa Bukula, and 29 others. 2025a. BRIGHTER: BRIdging the gap in human-annotated textual emotion recognition datasets for 28 languages. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8895–8916, Vienna, Austria. Association for Computational Linguistics.

Shamsuddeen Hassan Muhammad, Nedjma Ousidhoum, Idris Abdulmumin, Seid Muhie Yimam, Jan Philip Wahle, Terry Ruas, Meriem Beloucif, Christine De Kock, Tadesse Destaw Belay, Ibrahim Said Ahmad, Nirmal Surange, Daniela Teodorescu, David Ifeoluwa Adelani, Alham Fikri Aji, Felermino Dario Mario Ali, Vladimir Araujo, Abinew Ali Ayele, Oana Ignat, Alexander Panchenko, and 2 others. 2025b. SemEval-2025 task 11: Bridging the gap in text-based emotion detection. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, pages 2558–2569, Vienna, Austria. Association for Computational Linguistics.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Leonard Salewski, Stephan Alaniz, Isabel Rio-Torto, Eric Schulz, and Zeynep Akata. 2023. In-context impersonation reveals large language models' strengths and biases. In *Advances in Neural Information Processing Systems*.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. Quantifying language models' sensitivity to spurious features in prompt design or: How I learned to start worrying about prompt formatting. In *International Conference on Learning Representations*.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. 2023. Language models are multilingual chain-of-thought reasoners. In *International Conference on Learning Representations*.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. 2025. Mixture-of-agents enhances large language model capabilities. In *International Conference on Learning Representations*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Danny Wood, Tingting Mu, Andrew M. Webb, Henry W. J. Reeve, Mikel Luján, and Gavin Brown. 2023. A unified theory of diversity in ensemble learning. *Journal of Machine Learning Research*, 24(359):1–49.

Shijie Wu and Mark Dredze. 2020. Are all languages created equal in multilingual BERT? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 120–130, Online. Association for Computational Linguistics.

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, and 23 others. 2024. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

A Extended Related Work

This appendix expands the shortened Related Work in Section 2. The main paper keeps only the claims needed to motivate the topology–diversity matrix; the additional background here documents the broader literature connections. References for the methods summarized below are given in Section 2; the additional prompt-sensitivity reference is cited locally.

Parallel and sequential multi-agent inference. Parallel multi-call methods usually generate several candidate answers and aggregate them. Self-consistency samples multiple Chain-of-Thought traces from the same prompt and marginalizes over the final answers. Debate-style systems introduce interaction among agents, allowing one model or role to critique another before a final answer is chosen. Mixture-of-agents systems aggregate outputs from multiple proposers, sometimes across heterogeneous models or layers. Sequential methods instead refine a single trajectory: Self-Refine separates generation from feedback and revision, while Reflexion adds evaluative reflection across attempts. These systems demonstrate the usefulness of multiple calls, but they typically change topology, role structure, and diversity source together.

Why topology and diversity must be separated. The distinction matters because an additional call can help for different reasons. In a parallel topology, independent errors may cancel under aggregation. In a sequential topology, later calls may add evidence or correct earlier mistakes. However, the same topology can behave differently depending on how the calls are diversified. If agents differ only by sampling seed, their errors may remain correlated. If they differ by role prompt, they may attend to different evidence but still share the same underlying model limitations. If they differ by trained adapter weights, they may become individually stronger but may also become overconfident specialists. Our matrix separates these possibilities by crossing topology with stochastic, prompted, and learned diversity under the same three-call budget, instantiated independently on two backbones.

Diversity and ensemble competence. Ensemble theory suggests that aggregation is useful only when individual members are both competent and meaningfully diverse. Diversity alone is not sufficient: if all members share the same systematic error, a majority vote can simply reproduce that error. This is especially relevant for multilingual low-resource inference, where all calls may inherit the same weak language competence from the shared backbone. The learned-diversity condition is therefore not only a diversity intervention but also a competence intervention: QLoRA specialists can change the effective parameters used by each call while still preserving a shared base model.

Prompted diversity and role design. Role prompts and persona prompts can provide useful cognitive scaffolding, but they are sensitive to wording and may degrade performance when they over-constrain the model (Sclar et al., 2024). This motivates our additive lens design. The Surface, Context, and Structure agents are asked to pay particular attention to different evidence types, but they are not instructed to ignore the others. The goal is to bias attention without reducing individual competence, since restrictive role prompts can lower the quality of the members that aggregation depends on.

Multilingual transfer and low-resource emotion detection. Multilingual emotion detection exposes topology–diversity interactions that English-only evaluation may hide. Cross-lingual transfer is affected by typological similarity, script overlap,

tokenization, pretraining coverage, and resource level. Recent work further suggests that multilingual LLM reasoning may route through English and transfer unevenly across target languages. In this setting, stochastic and prompt-induced diversity may vary reasoning style without creating genuinely independent evidence for the lowest-resource languages. This motivates the learned-adapter condition and the per-language reporting used throughout the paper.

B Dataset, Split, and Reasoning-Bank Details

This section expands the evaluation-setting details compressed from Section 3.1. We evaluate on the nine-language low-resource BRIGHTER subset: Naija Pidgin (pcm), Mandarin Chinese (chn), Marathi (mar), Moroccan Arabic (ary), Tatar (tat), Emakhuwa (vmw), Mozambican Portuguese (ptmz), Zulu (zul), and Indonesian (ind) (Muhammad et al., 2025a). These languages span five scripts: Latin, Arabic, Chinese characters, Devanagari, and Cyrillic. They also cover several language families, including Niger-Congo, Sino-Tibetan, Indo-European, Turkic, Afro-Asiatic, and Austronesian.

We use the standard BRIGHTER splits: 11,214 public training instances pooled over the seven languages with public training data, 1,975 development instances, and 8,597 held-out test instances. Under the BRIGHTER/SemEval protocol, Zulu and Indonesian are Track-C languages with no target-language training rows. We therefore train learned adapters only on the pooled public training data from the seven Track-A languages and evaluate the resulting single multilingual system on all nine languages.

The reasoning-bank policy is target-excluded. Same-family sources are used when available; otherwise the bank falls back to script overlap, sociolinguistic contact, pretraining coverage, or script-agnostic affect anchors such as emoji. The purpose of the reasoning bank is not to provide target-language labeled supervision, but to supply compact language-specific context, cue reminders, and affect anchors that are held fixed across all topology–diversity cells.

B.1 Design evolution

The current bank design is the fourth iteration in an internal sequence. V1 attached one-line rationales directly to few-shot exemplars. This was auditable but inflated prompt length and made every exemplar edit require a paired rationale edit. V2 moved the rationale into a single shared reasoning seed per language, but this was too under-specified for languages with diverse affective cues. V3 expanded to twelve seeds per language, which improved coverage but increased context length and encouraged verbose reasoning. V4, the canonical design, uses four cue-first reasoning seeds per language. This was selected as a coverage-versus-brevity compromise and is reused unchanged across all six matrix cells.

Reasoning bank file: `{lang}_reasoning_seed42.json`

```
{
  "language_name": "{full name}",
  "script": "{Latin | Arabic | Cyrillic | Devanagari | Han}",
  "family": "{family or proxy class}",
  "seed": 42,
  "language_instructions": "{one-paragraph cue-and-context primer}",
  "reasoning_examples": [
    {
      "id": "{example id}", "emotion": "{emotion label}",
      "text": "{exemplar text}",
      "reasoning": {
        "cues": "{lexical and emoji evidence}",
        "tone": "{discourse-level affect}",
        "interpretations": "{alternative readings}",
        "critique": "{adjudication leading to label}"
      },
      "final_labels": ["{emotion label}"]
    }, ...
  ]
}
```

Table 4: Reasoning bank file schema. One file is used per target language under `data/reasoning_banks/`. The rendered bank is inserted into the CoT block in Table 15.

B.2 Bank JSON schema

Table 4 shows the schema for each language reasoning bank. The same file format is used for all nine languages.

B.3 Family/proxy transfer policy

The reasoning banks follow a target-excluded family/proxy policy. Same-family sources are used when available; otherwise, the bank falls back on script overlap, sociolinguistic contact, pretraining coverage, or script-agnostic affect anchors such as emoji. Table 5 summarizes the policy used for each target language. This table is included to

Target	Family	Script	Proxy basis
pcm	Creole	Latin	Native Pidgin examples plus English-lexifier and substrate cues
chn	Sino-Tibetan	Han	English-dominant proxy with script-agnostic emoji anchors
mar	Indo-European	Devanagari	Broad Indo-European pool, including Hindi as the closest sibling
ary	Afro-Asiatic	Arabic	Maghrebi Arabic sibling and Afro-Asiatic proxies
tat	Turkic	Cyrillic	Russian/Ukrainian proxies via script overlap and contact
vmw	Niger-Congo	Latin	Niger-Congo and Bantu-family proxies
ptmz	Indo-European	Latin	Romance and wider Indo-European proxies
zul	Niger-Congo	Latin	Niger-Congo and Bantu-family proxies
ind	Austronesian	Latin	Austronesian sibling and regional-register cues

Table 5: Family/proxy basis for each of the nine target languages. The policy prioritizes same-family target-excluded sources when available and uses proxy-transfer fallbacks where the available pool lacks close siblings.

make clear that proxy cases such as Chinese and Tatar are treated as bounded approximations, not as typological matches.

C Experimental Details

This section expands the compressed experimental protocol in Section 4. The complete six-cell matrix is evaluated on the same BRIGHTER development and held-out test splits for Qwen2.5-14B-Instruct and Llama-3.1-8B-Instruct. Within each backbone, cells share the same language-specific reasoning banks, JSON output schema, precision-prior decision policy, and three-call inference budget. Llama is used as the second-backbone replication of the matrix.

C.1 Comparison groups

The baselines use the same Qwen2.5-14B-Instruct backbone and the same output protocol as the Qwen matrix cells: zero-shot prompting, few-shot prompting, single-pass CoT, and seven-call CoT with self-consistency. These are the primary comparisons because differences are attributable to inference design rather than backbone, label schema, or output format. We additionally report BRIGHTER reference systems, including Qwen2.5-72B few-shot prompting, XLM-R, and mDeBERTa, as contextual scale and cross-lingual references. SemEval-2025 leaderboard comparisons are reported separately in Appendix F, but are not treated as important because those systems may use target-language supervision, per-language tuning, or larger ensembles.

Cell	Requests	Tokens (M)	Wall (h)
W-L1	5,925	6.74	11.7
W-L2	5,925	7.26	12.8
W-L3	5,925	6.46	11.8
D-L1	5,925	7.84	12.3
D-L2	5,925	8.21	12.9
D-L3	5,925	7.66	12.1

Table 6: Per-cell Qwen2.5-14B inference compute on the development split using a single 40GB GPU.

C.2 Statistical reporting

Table 1 reports the canonical run for each matrix cell. Bootstrap intervals quantify uncertainty over evaluation examples, while Appendix E reports targeted W-L3 training/inference seed perturbations as a separate robustness check. For key contrasts, we compute 95% paired non-parametric percentile bootstrap confidence intervals using 2,000 replicates. Each replicate resamples examples within language using the same draw for both systems and recomputes the language-uniform Macro-F1 difference. For the learned Width–Depth contrast we also report the Wilcoxon signed-rank test over the nine paired language-level scores as a complementary cross-language consistency check.

C.3 Compute reporting

Table 6 reports the per-cell development-split compute budget. All matrix cells issue the same number of requests, while token counts and wall-clock time vary slightly with topology and prompt length. These numbers report inference only and exclude one-time QLoRA adapter training. Across the Qwen matrix, inference used approximately 73.6 GPU-hours on development and 155 GPU-hours on test, excluding one-time adapter training.

D Per-Language Matrix Detail

Tables 7 and 8 expand the averaged matrix results from Table 1 into per-language test Macro-F1 for Qwen and Llama. The tables show that the learned-specialization gain is not driven by a single language and also expose the substantial remaining variation across languages.

E Seed Robustness

This appendix reports targeted seed perturbations for the headline W-L3 condition. The canonical run remains the value reported in Table 1; these perturbations are a sensitivity check rather than a full-matrix estimate of training-run variance. On Qwen2.5-14B, an alternative inference seed produces 54.70 dev and 53.38 test Macro-F1, versus 53.61 and 52.83 canonically. On Llama-3.1-8B, an alternative QLoRA training seed produces 54.73 dev and 53.62 test, versus 53.45 and 52.94 canonically. A further Llama seed-diversity variant reaches 53.67 dev and 52.89 test.

F SemEval Reference Detail

Table 10 provides the per-language reference comparison used in Section 5. These numbers are contextual rather than primary evidence because SemEval systems may use target-language supervision, per-language model selection, and larger ensembles. The table shows that W-L3 improves over the RoBERTa reference on eight of nine languages, while still trailing the per-language Top-1 systems on most languages.

G Ablation Details

This section collects the ablations supporting the mechanism claims in Section 5. The ablations test whether the W-L3 gain is due to learned adapters rather than co-varying temperature or prompt choices, whether the D-L3 deficit is caused by depth itself or by learned-specialist interaction, and whether aggregation or bootstrap uncertainty changes the interpretation.

H Additional Figures

Figure 3 visualizes the depth-layer dynamics behind the learned-depth inversion discussed in Section 5. Figure 4 provides a complementary per-language view, showing that the strongest W-L3 gains appear on the lower-resource end of the evaluation set. Figure 5 gives the ablation summary moved from the main text.

I Aggregation and Per-Emotion Detail

Table 13 reports post-hoc aggregation variants for Width cells. The main paper uses majority vote as the canonical topology-consistent rule for the controlled matrix. The variants here are diagnostic rather than new matrix cells: they show that prompt-only Width contains useful signal that majority vote does not fully exploit, and that W-L3 is mildly aggregation-sensitive under a more permissive `any 3` rule. Table 14 reports per-emotion F1 for canonical W-L3, showing that the best matrix cell remains uneven across emotion labels.

Aggregation notation. Let $\hat{y}_i^{(k)} \in \{0,1\}^{|\mathcal{E}|}$ denote the binary emotion vector emitted by Width agent $k \in \{1,2,3\}$ for example i, and let $c_{i,e}^{(k)} \in [0,1]$ denote the confidence score emitted by agent k for emotion e. The canonical majority rule used in the main matrix is

$$\hat{y}_{i,e}^{\text{maj}} = \mathbf{1}\left[\sum_{k=1}^{3} \hat{y}_{i,e}^{(k)} \geq 2\right].$$

The `any 3` rule emits a label when any agent emits it:

$$\hat{y}_{i,e}^{\text{any}} = \mathbf{1}\left[\sum_{k=1}^{3} \hat{y}_{i,e}^{(k)} \geq 1\right].$$

The `confidence_max` rule selects, for each label independently, the decision of the agent with the highest reported confidence:

$$k_{i,e}^{*} = \arg\max_{k\in\{1,2,3\}} c_{i,e}^{(k)}, \qquad \hat{y}_{i,e}^{\text{conf}} = \hat{y}_{i,e}^{(k_{i,e}^{*})}.$$

J Prompt and Reasoning Protocol

This section documents the prompt and reasoning protocol used by all matrix cells. The goal is to make the fixed components of the experiment auditable: the same reasoning directive, language-specific reasoning bank, decision policy, and JSON output schema are reused across Width and Depth, across all three diversity levels. This is important because the main paper attributes performance differences to topology and diversity source, not to hidden changes in task instructions or output format. The fixed-budget comparison applies to the six matrix cells: each uses exactly three LLM calls per sample. Baselines are reported with their

Cell	pcm	chn	mar	ary	tat	vmw	ptmz	zul	ind	Avg
W-L1	35.65	59.53	74.56	44.71	43.43	4.39	38.60	13.59	51.48	40.66
W-L2	36.69	59.96	73.70	46.59	41.46	2.02	40.27	13.69	52.00	40.71
W-L3	57.02	67.12	72.20	62.44	63.02	19.16	56.40	27.05	51.03	52.83
D-L1	40.43	61.36	76.30	51.21	47.52	6.39	41.79	15.80	54.11	43.88
D-L2	43.18	60.57	76.68	50.49	48.23	7.74	45.13	18.96	54.66	45.07
D-L3	47.52	64.40	75.76	56.92	55.23	18.55	52.15	26.78	52.67	50.00

Table 7: Per-language Qwen2.5-14B test Macro-F1 (%) for every matrix cell; Avg is the uniform mean over the nine languages.

Cell	pcm	chn	mar	ary	tat	vmw	ptmz	zul	ind	Avg
W-L1	38.51	60.14	78.42	45.95	49.23	10.28	51.03	16.53	50.88	44.55
W-L2	39.64	60.29	77.64	46.79	50.63	10.95	51.44	16.58	51.86	45.09
W-L3	55.87	64.77	79.45	55.64	60.65	19.47	57.55	30.45	52.63	52.94
D-L1	43.50	61.47	77.94	49.98	48.20	14.73	49.15	22.34	55.86	47.02
D-L2	44.23	60.46	78.58	48.46	46.31	13.86	48.93	20.59	55.72	46.35
D-L3	54.53	65.39	80.63	54.10	58.92	17.68	58.49	30.46	54.71	52.77

Table 8: Per-language Llama-3.1-8B-Instruct test Macro-F1 (%) for every matrix cell; Avg is the uniform mean over the nine languages.

Backbone	Perturbation	Canonical	Repeat	$\lvert\Delta\rvert$
Qwen2.5-14B	Inference seed	52.83	53.38	0.55
Llama-3.1-8B	Training seed	52.94	53.62	0.68
Llama-3.1-8B	Seed-diversity variant	52.94	52.89	0.05

Table 9: Individual W-L3 seed runs on the test split (Macro-F1, %). The observed shifts are much smaller than the learned-specialization gains over W-L1 and W-L2.

own call counts, so the comparison to CoT+self-consistency at $k = 7$ is intentionally conservative with respect to call budget.

J.1 Shared-call notation

For each example i, let x_i be the input text, $L_i \in \mathcal{L}$ the language identifier, b_{L_i} the rendered reasoning bank for that language, and $y_i \in \{0,1\}^{|\mathcal{E}|}$ the gold multi-label vector over the six emotions. Every call receives the shared input package

$$u_i = (x_i, L_i, b_{L_i}).$$

Each model response is parsed as a JSON object

$$z_i = (\pi(z_i), \kappa(z_i), r_i),$$

where $\pi(z_i) \in \{0,1\}^{|\mathcal{E}|}$ is the binary emotion vector, $\kappa(z_i) \in [0,1]^{|\mathcal{E}|}$ is the confidence vector, and r_i is the short reasoning field. Only $\pi(z_i)$ is used for scoring; $\kappa(z_i)$ is used only in post-hoc aggregation analyses, and r_i is used for prompting and qualitative analysis.

J.2 CoT block and decision policy

Table 15 gives the shared CoT block. The decision policy is intentionally conservative: it asks the model to identify evidence first, default to a single dominant emotion, and assign positive labels only when evidence is compelling. This policy is held fixed across all six matrix cells.

J.3 Task instruction and JSON output schema

Table 16 gives the invariant task instruction and JSON schema. The schema emits one binary decision per emotion, one confidence score per emotion, and a short reasoning field. This lets us re-aggregate saved predictions under alternative aggregation rules without rerunning inference.

K Width Topology Prompt Designs

Every Width cell composes its system prompt as an optional analytical lens followed by the shared CoT block in Table 15. W-L1 and W-L3 use no analytical lens: their system prompt is only the shared CoT block. W-L2 uses three additive analytical lenses. The lenses are additive rather than restrictive: each agent is encouraged to attend to one evidence type, but is not forbidden from using the others.

W-L1 agents differ only through decoding stochasticity at $T = 0.7$. W-L3 agents share the same prompt form but load different QLoRA adapters at $T = 0.3$. Thus, the diversity in W-L3 is parametric rather than prompt-level.

Lang	Track	W-L3	RoBERTa	Δ_R	Top-1	Δ_1
pcm	A	0.570	0.555	+0.015	0.674	-0.104
chn	A	0.671	0.531	+0.140	0.709	-0.038
mar	A	0.722	0.822	-0.100	0.884	-0.162
ary	A	0.624	0.472	+0.152	0.629	-0.005
tat	A	0.630	0.539	+0.091	0.846	-0.216
vmw	A	0.192	0.121	+0.071	0.325	-0.133
ptmz	A	0.564	0.459	+0.105	0.548	+0.016
zul	C	0.271	0.153	+0.118	0.397	-0.126
ind	C	0.510	0.376	+0.134	0.672	-0.162
Avg	–	0.528	0.448	+0.080	0.632	-0.104

Table 10: Per-language comparison with SemEval-2025 references. Δ_R is W-L3 minus the RoBERTa baseline; Δ_1 is W-L3 minus the per-language Top-1 submission.

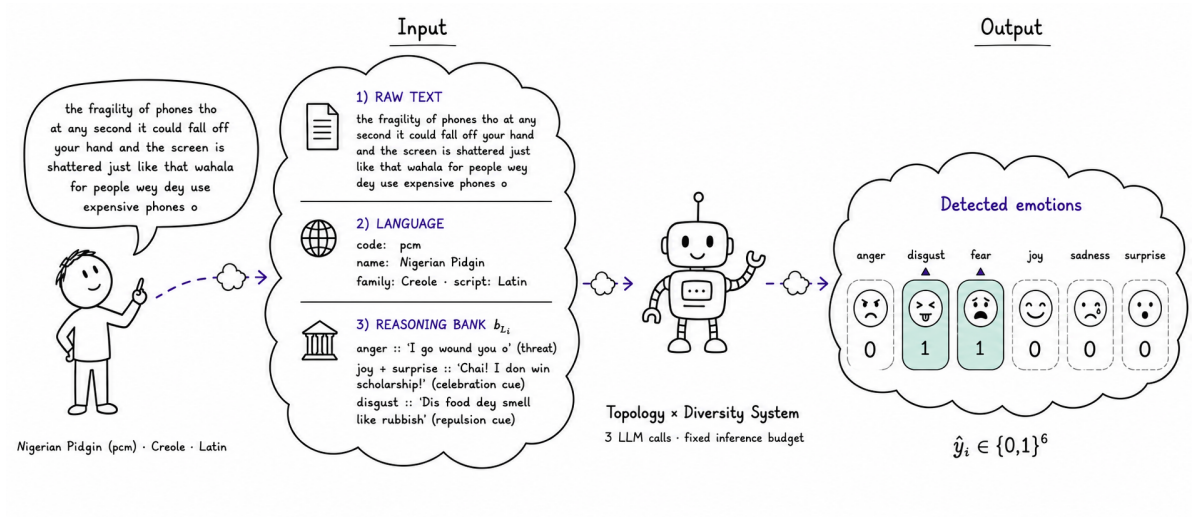

Figure 2: Input–output example for the shared multilingual emotion-detection interface. The input is a Nigerian Pidgin (pcm) post. Each call receives the raw text x_i, language metadata L_i, and the rendered reasoning bank b_{L_i}; the topology–diversity system then connects and diversifies three LLM calls under the fixed inference budget. The output is a JSON-compatible multi-label prediction vector $\hat{y}_i \in \{0,1\}^6$, ordered as ANGER, DISGUST, FEAR, JOY, SADNESS, and SURPRISE. The shown prediction activates DISGUST and FEAR.

L Depth Topology Inter-Layer Prompts

Depth cells emit three sequential calls. Layer 1 sees the input directly. Layer 2 receives the original input and Layer 1's serialized JSON output; Layer 3 receives the original input and the serialized JSON outputs from both previous layers. This makes the earlier prediction an external artifact that later calls can inspect, revise, or confirm. The depth topology also uses a descending temperature schedule, $0.7 \rightarrow 0.5 \rightarrow 0.3$, implementing an exploration-to-consolidation pattern.

L.1 Depth-chain notation

For diversity level $\delta \in \{\mathrm{L1}, \mathrm{L2}, \mathrm{L3}\}$, Depth executes three JSON-conditioned calls:

$$z_i^{(1)} = f_1^{\delta}(u_i; T_1),$$

$$z_i^{(2)} = f_2^{\delta}(u_i, z_i^{(1)}; T_2),$$

$$z_i^{(3)} = f_3^{\delta}(u_i, z_i^{(1)}, z_i^{(2)}; T_3).$$

The final Depth prediction is extracted from the third JSON output:

$$\hat{y}_i^D = \pi(z_i^{(3)}).$$

The fixed temperature schedule is

$$(T_1, T_2, T_3) = (0.7, 0.5, 0.3),$$

which implements the exploration-to-consolidation pattern described in the main paper.

L.2 Inter-layer conditioning template

Table 20 gives the user-turn template used for inter-layer conditioning. For D-L1, all three layers share the same generic role and differ through sequential

Claim tested	Ablation(s)	Intervention	Main result	Interpretation
W-L3 gain is mainly due to learned adapters, not decoding temperature	A1	Run W-L3 at $T = 0.7$, matching the higher prompt-only sampling temperature	W-L3 remains strong; increasing temperature reduces test Macro-F1 by 1.26 points relative to canonical W-L3	The learned-adapter gain is not explained by the lower $T = 0.3$ decoding setting.
W-L3 gain is not primarily caused by prompt engineering	A7, A8	Replace the full system prompt with a minimal prompt, or remove the system prompt entirely	Minimal prompt costs 2.71 points; no-system prompt costs 2.84 points, while W-L3 remains far above W-L2	Adapters carry most of the performance gain; prompt content contributes but is not the main driver.
Explicit per-agent prompts do not add to learned adapter diversity	A9	Add per-agent prompts on top of the three W-L3 adapters	Performance is essentially flat: $+0.08$ dev and -0.12 test	Prompt diversity and adapter diversity do not compose additively in this setting.
All W-L3 specialists are important	A2	Leave-one-out re-aggregation over the three Width-L3 adapters	Removing `general`, `contrastive`, and `ambiguity` drops test Macro-F1 by 5.29, 4.78, and 2.01 points, respectively	The Width-L3 ensemble is not driven by a single adapter; learned diversity is functionally used by aggregation.
Learned Depth underperforms because of sequential specialist interaction, not because depth is generally harmful	A3, A13	Re-score Depth predictions after truncating at Layer 1 or Layer 2 across D-L1, D-L2, and D-L3	Full chains help D-L1 and D-L2, but D-L3 peaks at Layer 1: 52.11 versus 50.00 for the full chain	The learned-depth failure is specific to trained specialists revising one another, not to JSON-conditioned depth itself.
Layer 2 is not the sole cause of D-L3 degradation	A6	Skip Layer 2 in D-L3 and run Layer 1 directly into Layer 3	Skip-L2 scores 50.04, nearly identical to canonical D-L3 at 50.00	The degradation is caused by the later learned calibration process as a whole, not only by the critic layer.
Width aggregation rule is a diagnostic factor, especially for prompt-only systems	A5	Re-aggregate saved Width predictions with confidence-max and any-of-three rules	Confidence-max improves W-L1 by 3.73 and W-L2 by 4.41 points; W-L3 is comparatively stable, with any-of-three adding 1.13 points	Prompt-only Width contains signal that majority vote underuses; W-L3 is less dependent on aggregation choice.
Headline learned-cell comparisons should be read with uncertainty estimates	A4	Bootstrap test predictions with 2,000 paired resamples	The W-L3 vs. W-L2 gap is much larger than the bootstrap uncertainty; the W-L3 vs. D-L3 gap is smaller and should be read more cautiously	The main learned-diversity claim is robust, while the topology gap at L3 is directionally meaningful but less decisive.

Table 11: Ablation summary. Each ablation targets a specific mechanism required by the topology–diversity interpretation: whether W-L3 gains come from adapter weights rather than co-varying prompt or temperature choices, whether the D-L3 deficit is caused by depth itself or by learned specialist interaction, and whether aggregation or bootstrap uncertainty changes the interpretation.

Cell	L1 only	L1+L2	Full	Δ L1–Full
D-L1	41.87	43.69	**43.88**	-2.01
D-L2	41.86	**45.46**	45.07	-3.21
D-L3	**52.11**	49.94	50.00	+2.11

Table 12: Depth truncation results on the test split (Macro-F1, %). For D-L1 and D-L2, sequential refinement improves over Layer 1 alone. For D-L3, Layer 1 alone is strongest, showing that the learned-depth failure is specific to specialist interaction rather than depth itself.

Cell	maj. 2/3	conf. max	any 3
W-L1	40.66	**44.39**	41.21
W-L2	40.71	**45.12**	41.79
W-L3	52.83	52.91	**53.96**

Table 13: Aggregation-rule sensitivity for WIDTH cells (test Macro-F1, %).

conditioning and temperature. For D-L2 and D-L3, Layers 2 and 3 additionally receive the Critic and Calibrator role prompts in Tables 21 and 22.

L.3 D-L2 and D-L3 role prompts

M Adapter Training Details

This appendix records the learned-specialization implementation. The detailed table below gives the Qwen2.5-14B-Instruct (Yang et al., 2024) adapter configuration. The second-backbone experiment instantiates the same Width specialist roles and Depth position-specific objectives on Llama-3.1-8B-Instruct (Grattafiori et al., 2024) using the same pooled BRIGHTER training split; model-specific seed runs are reported in Appendix E. The purpose of L3 is to make specialization parametric while keeping the number of inference calls and output protocol fixed within each backbone.

M.1 Qwen shared training recipe

All Qwen L3 adapters use the same base recipe: QLoRA over the frozen Qwen2.5-14B-Instruct backbone, NF4 quantization with double quantization, LoRA rank $r = 8$, $\alpha = 16$, dropout 0.05, AdamW optimization, bf16 compute, and early stopping on development loss. The training data

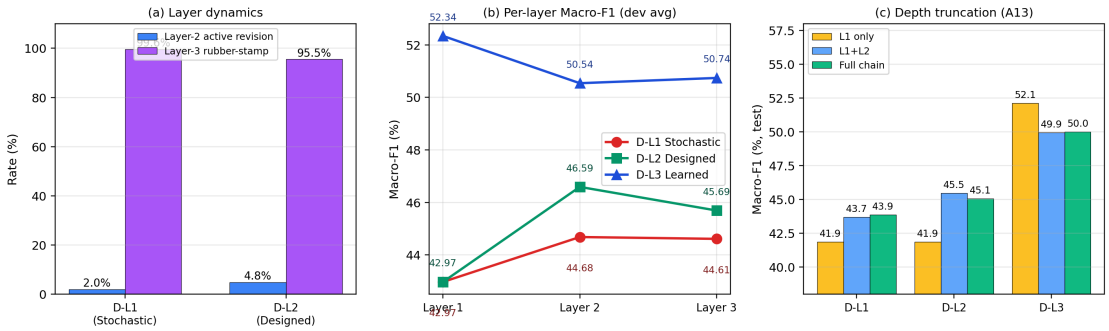

Figure 3: Depth dynamics and truncation analysis. D-L1 and D-L2 gain from sequential layers; D-L3 peaks at Layer 1 and erodes under later learned specialists.

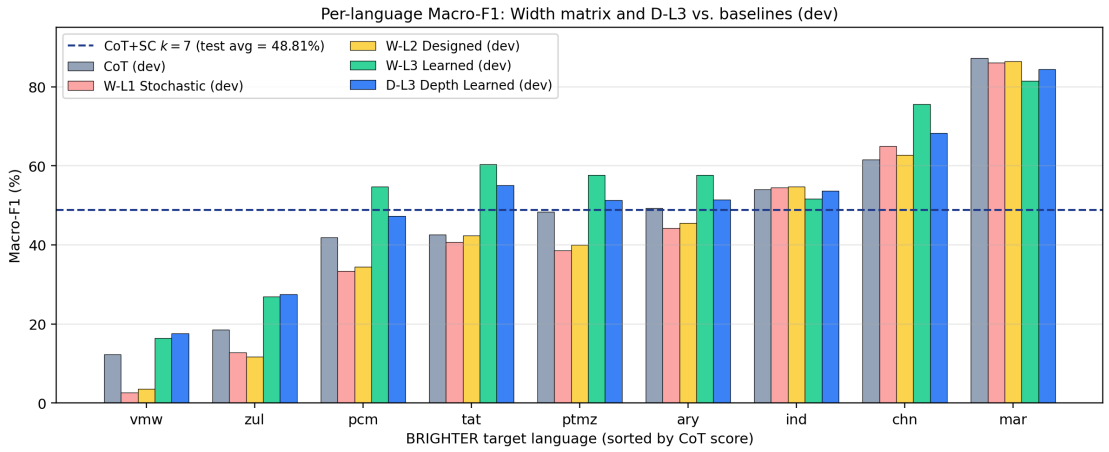

Figure 4: Per-language development results ordered by single-pass CoT score. The largest W-L3 gains appear on the low-resource tail.

Emotion	W-L3 test F1
anger	48.11
disgust	31.42
fear	49.75
joy	71.80
sadness	67.13
surprise	48.76

Table 14: Per-emotion test F1 (%) for W-L3, averaged over languages.

are the pooled public training rows over the seven languages with Track-A training data; Zulu and Indonesian have no train rows under the Track-C protocol. Loss is computed only over the assistant turn, with system and user tokens masked.

Adapter notation. Let M_{θ_0} denote the frozen Qwen2.5-14B-Instruct backbone. For Width-L3, let $\mathcal{S}_W = \{\text{gen}, \text{amb}, \text{ctr}\}$ denote the `general`, `ambiguity`, and `contrastive` adapters. For Depth-L3, let ϕ_1, ϕ_2, ϕ_3 denote the Layer 1, Layer 2, and Layer 3 adapter parameters.

Adapter parameterization. For an adapted linear map with frozen weight W_0, QLoRA uses the effective weight

$$W_s = Q(W_0) + \Delta W_s, \qquad \Delta W_s = \frac{\alpha}{r} B_s A_s,$$

where $Q(W_0)$ is the NF4-quantized frozen weight, A_s and B_s are trainable low-rank matrices, $r = 8$, and $\alpha = 16$. Only the adapter parameters $\phi_s = \{A_s, B_s\}$ are updated; the backbone parameters θ_0 remain frozen.

Assistant-token objective. Each training example is serialized as a chat sequence $c_i = [m_i^{\text{sys}}, m_i^{\text{usr}}, m_i^{\text{asst}}]$, where the assistant turn contains the target JSON object. Let $\mathcal{A}_i$ be the set of token positions belonging to the assistant turn. The weighted supervised loss for specialist s is

$$\mathcal{L}_{\text{sup}}^{(s)} = -\sum_{i=1}^{N} w_i^{(s)} \sum_{t \in \mathcal{A}_i} \log p_{\theta_0, \phi_s}(c_{i,t} \mid c_{i,<t}).$$

System and user tokens are masked out of the loss, so the model is optimized only to generate the

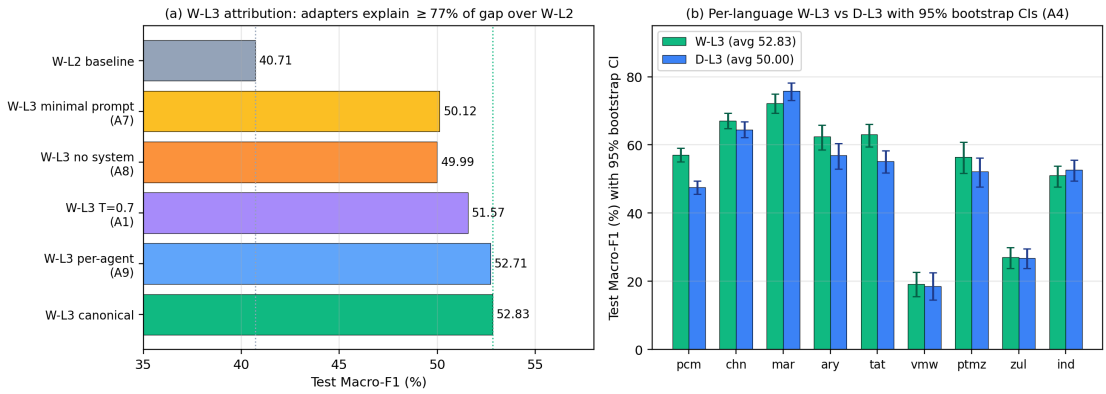

Figure 5: Ablation summary. Left: W-L3 remains strong under prompt and temperature ablations. Right: W-L3 vs. D-L3 per language with bootstrap confidence intervals.

CoT block (verbatim, identical across all six cells)

`Before classifying, reason step by step about the emotions in this text.`
`Language-specific guidance for {lang}:`
`{REASONING_BANK}`
`Decision policy:`
`- Identify emotional cues first (words, phrases, emoji, tone)`
`- Consider cultural and linguistic context for {lang}`
`- Default to a single dominant emotion unless two distinct cue clusters clearly justify multiple labels`
`- Assign 1 only when evidence is compelling; when uncertain, assign 0`

Table 15: The CoT block injected as the system-prompt prefix in every Width and Depth call. `{lang}` is the BRIGHTER language code; `{REASONING_BANK}` is the rendered per-language bank described in Appendix B.

Task instruction (verbatim user-turn template)

`Task: Classify the emotions expressed in the following text. The text may express zero, one, or multiple emotions simultaneously.`
`Emotion categories: anger, disgust, fear, joy, sadness, surprise`
`For each emotion, decide if it is present (1) or absent (0). Provide a confidence score between 0.0 and 1.0, and brief reasoning.`
`Respond ONLY with a valid JSON object in exactly this format, with no additional text before or after:`
`{`
`  "emotions": {"anger": 0, "disgust": 0, "fear": 0, "joy": 0, "sadness": 0, "surprise": 0},`
`  "confidence": {"anger": 0.0, "disgust": 0.0, "fear": 0.0, "joy": 0.0, "sadness": 0.0, "surprise": 0.0},`
`  "reasoning": "Your analysis here."`
`}`
`Text: {INPUT_TEXT}`

Table 16: The task instruction and JSON schema delivered through the user turn. This template is identical across all six matrix cells.

assistant-side JSON response.

M.2 Width-L3 specialists

The Width-L3 cell uses three parallel specialist adapters, abbreviated as gen, amb, and ctr, corresponding to `general`, `ambiguity`, and `contrastive`.

`general`. This adapter is trained uniformly on the pooled training set and acts as the baseline supervised specialist. Its objective is

$$\mathcal{L}_{\text{gen}} = \mathcal{L}_{\text{sup}}^{(\text{gen})}, \qquad w_i^{(\text{gen})} = 1.$$

It is the most important adapter in leave-one-out analysis: removing it reduces test Macro-F1 by 5.29 points.

`ambiguity`. This adapter upweights multi-label and hard examples, with sample weight doubled for multi-label or difficult cases. Let $\mathcal{H}$ denote the set of hard examples identified for this specialist. Its example weight is

$$w_i^{(\text{amb})} = 1 + \mathbf{1}[\|y_i\|_1 > 1 \ \vee \ i \in \mathcal{H}],$$

so examples with multiple gold emotions or hard-case status receive weight 2. The objective is

$$\mathcal{L}_{\text{amb}} = \mathcal{L}_{\text{sup}}^{(\text{amb})}.$$

It is intended to specialize in co-occurring emotions, where a single dominant-label prior may be insufficient. Removing it reduces test Macro-F1 by 2.01 points.

`contrastive`. This adapter is trained on the full data with an additional cosine-hinge contrastive

W-L2 Agent A — *Surface-Priority* lens
`You are an expert emotion detection system. When analyzing text, pay particular attention to explicit emotional signals: emotional keywords, sentiment-bearing phrases, emoji, emoticons, punctuation patterns (exclamation marks, repeated characters, ellipses), capitalization, and interjections. These surface cues should be your primary evidence, though you may also consider contextual meaning where relevant.`

Table 17: Surface-Priority lens system prompt for the W-L2 surface agent. The lens is concatenated with the language-specific CoT block in Table 15.

W-L2 Agent B — *Context-Priority* lens
`You are an expert emotion detection system. When analyzing text, pay particular attention to contextual and pragmatic meaning: implied emotions, overall tone, potential sarcasm or irony, rhetorical questions, cultural and idiomatic expressions, and what the author likely feels even if not stated explicitly. Look beyond literal word meanings, though you may also consider surface cues where relevant.`

Table 18: Context-Priority lens system prompt for the W-L2 context agent.

term over emotion-present and emotion-absent representations. Let h_i denote the representation used for contrastive training, let $P_i = \{e \in \mathcal{E} : y_{i,e} = 1\}$ be the set of present emotions, and let $N_i = \{e \in \mathcal{E} : y_{i,e} = 0\}$ be the set of absent emotions. Let q_e denote the representation associated with emotion e. The contrastive term is

$$\mathcal{L}_{\text{ctr}} = \sum_{i=1}^{N} \sum_{e^+ \in P_i} \sum_{e^- \in N_i} \ell_i(e^+, e^-),$$
$$\ell_i(e^+, e^-) = \max\big(0,\, s_i(e^+, e^-)\big)\,,$$
$$s_i(e^+, e^-) = m - \cos(h_i, q_{e^+}) + \cos(h_i, q_{e^-}).$$

Here m is the hinge margin. The contrastive specialist minimizes

$$\mathcal{L}_{\text{ctr-adapter}} = \mathcal{L}_{\text{sup}}^{(\text{ctr})} + \lambda_{\text{ctr}} \mathcal{L}_{\text{ctr}}.$$

It is intended to sharpen separation between affective categories and non-present labels. Removing it reduces test Macro-F1 by 4.78 points.

M.3 Depth-L3 specialists

The Depth-L3 cell uses three position-specific adapters. Layer 1 uses a uniform analyst adapter trained on the pooled public training set:

$$\mathcal{L}_1 = \mathcal{L}_{\text{sup}}^{(1)}.$$

W-L2 Agent C — *Structure-Priority* lens
`You are an expert emotion detection system. When analyzing text, pay particular attention to how emotional signals interact across the text: sentiment shifts between clauses, contrasts or contradictions, co-occurring emotions that may reinforce or conflict with each other, and the overall emotional arc. Consider whether the text expresses one coherent emotion or multiple interacting emotions, though you may also consider individual cues where relevant.`

Table 19: Structure-Priority lens system prompt for the W-L2 structure agent.

Inter-layer conditioning user-turn template (Layers 2 and 3)

`A previous analysis of the following text produced this result:`
`—`
`{PRIOR_JSON}`
`—`
`Now analyze the same text yourself. You may agree with, modify, or completely revise the previous analysis. Provide your own classification.`
`{TASK_INSTRUCTION}`
`Text: {INPUT_TEXT}`

Table 20: Inter-layer conditioning template prepended to the user message of Layers 2 and 3 in the Depth topology. `{PRIOR_JSON}` contains Layer 1's full JSON output for Layer 2, and both earlier layers' full JSON outputs for Layer 3. The instruction permits agreement, modification, or full revision; the model is not forced to dissent.

Layer 2 uses a critic adapter trained to correct Layer 1 mistakes. Let

$$\epsilon_i^{(1)} = \mathbf{1}\Big[\pi(z_i^{(1)}) \neq y_i\Big]$$

indicate whether Layer 1 makes an example-level prediction error. The Layer 2 training loader gives mistaken Layer 1 examples an effective weight of 3.0:

$$w_i^{(2)} = 1 + 2\epsilon_i^{(1)}.$$

The Layer 2 objective is

$$\mathcal{L}_2 = -\sum_{i=1}^{N} w_i^{(2)} \sum_{t \in \mathcal{A}_i} \log p_{\theta_0,\phi_2}\Big(c_{i,t} \mid c_{i,<t}, z_i^{(1)}\Big)\,.$$

Layer 3 uses a calibrator adapter trained on Layer 2 mistakes, but with a weaker correction

Layer 2 Critic role (D-L2 and D-L3)
You are a critical reviewer of emotion analysis. Your job is to find and correct errors in a previous analysis. When reviewing, specifically check for: (a) Emotions that may have been missed because they are expressed implicitly, culturally, or through understatement. (b) Emotions that may have been incorrectly assigned due to surface-level pattern matching without considering context, sarcasm, or irony. (c) Confidence scores that seem miscalibrated relative to the evidence. (d) Inconsistencies between the reasoning and the predicted labels. If the previous analysis is correct, confirm it and explain why. Do not change predictions without justification.

Table 21: Critic system prompt for Layer 2 in D-L2 and D-L3. The role prompt is concatenated with the shared CoT block in Table 15.

Layer 3 Calibrator role (D-L2 and D-L3)
You are performing final calibration on an emotion analysis that has undergone two rounds of review. Your task is to: (a) Confirm emotions where the evidence is strong and both prior analyses agree. (b) Adjudicate where the two analyses disagree, weighing the evidence for each side. (c) Remove emotions assigned on weak or ambiguous evidence – prioritize precision over recall. (d) Ensure confidence scores accurately reflect the strength of evidence.

Table 22: Calibrator system prompt for Layer 3 in D-L2 and D-L3. The D-L3 truncation results in Table 12 show that this final learned calibration stage can become harmful when the first learned specialist is already strong.

weight. Let

$$\epsilon_i^{(2)} = \mathbf{1}\left[\pi(z_i^{(2)}) \neq y_i\right].$$

The calibrator applies a calibration scaling of 0.2 to the Layer 2 error upweight, giving an additional error weight of 0.6:

$$w_i^{(3)} = 1 + 0.6\epsilon_i^{(2)}.$$

Let $z_i^{(<3)} = (z_i^{(1)}, z_i^{(2)})$ denote the previous Depth outputs available to Layer 3. The Layer 3 objective is

$$\mathcal{L}_3 = -\sum_{i=1}^{N} w_i^{(3)} \sum_{t \in \mathcal{A}_i} \log p_{\theta_0,\phi_3}\left(c_{i,t} \mid c_{i,<t}, z_i^{(<3)}\right).$$

The truncation results in Table 12 show that this learned correction chain is miscalibrated once the first specialist is already strong: D-L3 Layer 1 alone reaches 52.11 test Macro-F1, while the full learned chain reaches 50.00.

M.4 Multi-LoRA serving

All L3 inference uses a single vLLM engine with LoRA support enabled. The frozen base model is loaded once, and adapters are hot-swapped per call through LoRA requests. This means W-L3 preserves the same three-call inference budget as W-L1 and W-L2: it does not require loading or serving three separate full models.

For Width-L3, the three adapter predictions are

$$\hat{y}_i^{(k)} = \pi(f_{\theta_0,\phi_k}(u_i)), \qquad k \in \mathcal{S}_W.$$

The final prediction is the same per-label majority vote used by all Width cells:

$$\hat{y}_{i,e}^{W\text{-}L3} = \mathbf{1}\left[\sum_{k=1}^{3} \hat{y}_{i,e}^{(k)} \geq 2\right].$$

For Depth-L3, the position-specific adapters are applied sequentially:

$$z_i^{(1)} = f_{\theta_0,\phi_1}(u_i),$$
$$z_i^{(2)} = f_{\theta_0,\phi_2}(u_i, z_i^{(1)}),$$
$$z_i^{(3)} = f_{\theta_0,\phi_3}(u_i, z_i^{(1)}, z_i^{(2)}),$$

with final prediction

$$\hat{y}_i^{D\text{-}L3} = \pi(z_i^{(3)}).$$

N Source Code and Configuration Map

The source code, prompts, configurations, evaluation scripts, and experiment utilities are publicly available at https://github.com/eracoding/topologyxdiversity. Table 25 maps the main experimental components to their implementation paths. The repository mirrors this structure, with prompts, configuration files, prediction outputs, and ablation scripts included for reproducibility.

Software environment. The reproducibility package includes a `requirements.txt` file specifying the minimum software versions used by the implementation. The requirements cover model serving, data processing, evaluation, configuration parsing, and utility scripts. QLoRA adapter training additionally depends on the standard Hugging

Cell	Adapter	Input conditioning	Training objective / weighting	QLoRA setup	Optimization setup
W-L3	`general`	Text + language reasoning bank	Uniform supervised BCE on pooled Track-A training data	$r = 8$, $\alpha = 16$, dropout 0.05, 4-bit NF4 double quantization	AdamW; LR $= 10^{-5}$; batch $= 4$; grad. accum. $= 4$; effective batch $= 16$; cosine schedule; warmup ratio $= 0.10$; max epochs $= 5$
W-L3	`ambiguity`	Text + language reasoning bank	Supervised BCE; multi-label or hard examples weighted $\times 2$	$r = 8$, $\alpha = 16$, dropout 0.05, 4-bit NF4 double quantization	AdamW; LR $= 10^{-5}$; batch $= 4$; grad. accum. $= 4$; effective batch $= 16$; cosine schedule; warmup ratio $= 0.10$; early-stopped at 3 epochs
W-L3	`contrastive`	Text + language reasoning bank	Supervised BCE plus cosine-hinge contrastive loss over emotion-present vs. emotion-absent representations	$r = 8$, $\alpha = 16$, dropout 0.05, 4-bit NF4 double quantization	AdamW; LR $= 10^{-5}$; batch $= 4$; grad. accum. $= 4$; effective batch $= 16$; cosine schedule; warmup ratio $= 0.10$; max epochs $= 5$
D-L3	`layer1`	Text + language reasoning bank	Uniform supervised BCE on pooled Track-A training data	$r = 8$, $\alpha = 16$, dropout 0.05, 4-bit NF4 double quantization	AdamW; LR $= 10^{-5}$; batch $= 4$; grad. accum. $= 4$; effective batch $= 16$; cosine schedule; warmup ratio $= 0.10$; max epochs $= 5$; early-stopping patience $= 2$
D-L3	`layer2`	Text + Layer-1 JSON output + language reasoning bank	Supervised BCE; examples mispredicted by Layer 1 weighted $\times 3.0$	$r = 8$, $\alpha = 16$, dropout 0.05, 4-bit NF4 double quantization	AdamW; LR $= 10^{-5}$; batch $= 4$; grad. accum. $= 4$; effective batch $= 16$; cosine schedule; warmup ratio $= 0.10$; max epochs $= 5$; early-stopping patience $= 2$
D-L3	`layer3`	Text + Layer-2 JSON output + language reasoning bank	Supervised BCE; examples mispredicted by Layer 2 receive calibration-scaled additional weight 0.6	$r = 8$, $\alpha = 16$, dropout 0.05, 4-bit NF4 double quantization	AdamW; LR $= 10^{-5}$; batch $= 4$; grad. accum. $= 4$; effective batch $= 16$; cosine schedule; warmup ratio $= 0.10$; max epochs $= 5$; early-stopping patience $= 2$

Table 23: QLoRA adapter training setup for the Qwen learned-specialization cells. All adapters use the same frozen Qwen2.5-14B-Instruct backbone, LoRA applied to `q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, and `down_proj`, bf16 compute, loss masking over non-assistant tokens, and the shared JSON output schema. W-L3 adapters use temperature 0.3, top-$p = 0.95$, and seed 42 at inference. D-L3 uses the layer-wise temperature schedule $(T_1, T_2, T_3) = (0.7, 0.5, 0.3)$, with top-$p = 0.95$ and seed 42 at each layer. Zulu and Indonesian have no Track-C training rows and are not used in adapter training.

Face/PEFT/bitsandbytes stack. The main dependencies are summarized in Table 24. Experiment-specific settings, including decoding parameters, adapter paths, aggregation rules, and evaluation scripts, are provided in the corresponding configuration files and source-code directories.

Component	Version requirement
vLLM serving	`vllm>=0.4.0`
PyTorch	`torch>=2.1.0`
Transformers	`transformers>=4.40.0`
PEFT	`peft>=0.10.0`
bitsandbytes	`bitsandbytes>=0.43.0`
Accelerate	`accelerate>=0.27.0`
pandas	`pandas>=2.0.0`
NumPy	`numpy>=1.24.0`
scikit-learn	`scikit-learn>=1.3.0`
PyYAML	`pyyaml>=6.0`
tqdm	`tqdm>=4.65.0`
jsonlines	`jsonlines>=4.0.0`
matplotlib	optional, `matplotlib>=3.7.0`
seaborn	optional, `seaborn>=0.12.0`
SciPy	optional, `scipy>=1.11.0`

Table 24: Minimum software requirements used for serving, adapter training, data processing, evaluation, and analysis. Optional visualization packages are not required to reproduce the main numerical results.

Component	Source path
Shared base config	`config/base.yaml`
W-L1 / W-L2 / W-L3 configs	`config/config_b_l{1,2,3}.yaml`
D-L1 / D-L2 / D-L3 configs	`config/config_c_l{1,2,3}.yaml`
Width topology orchestrator	`src/topologies/width.py`
Depth topology orchestrator	`src/topologies/depth.py`
Base and prompt agents	`src/agents/base_agent.py`, `prompt_agent.py`
LoRA agent	`src/agents/lora_agent.py`
W-L2 lens prompts	`src/prompts/system_prompts.py`
Depth role prompts	`src/prompts/inter_layer.py`
CoT block builder	`src/prompts/cot_enhancement.py`
Task instruction and JSON schema	`src/prompts/task_template.py`
Per-label majority vote	`src/aggregation/majority_vote.py`
Reasoning banks	`data/reasoning_banks/{lang}_*.json`
Shot banks	`data/shot_banks/{lang}_*.json`
Width-L3 adapters	`models/adapters/b_l3_{1,2,3}/`
Depth-L3 adapters	`models/adapters/c_l3_layer{1,2,3}/`
Adapter training scripts	`training/train_b_adapters.py`, `training/train_c_adapters.py`
Per-cell runners	`experiments/run_{b,c}_l{1,2,3}.py`
Results	`outputs/results/`
Ablation suite	`ablation/{a1,...,a13}/`

Table 25: Source-code and configuration map for the components referenced in the main paper.