Typical: Typed Decisions Read Directly from Pretrained Language Models

Guy Nachshon
OzLabs
Guy@ozlabs.ai

Abstract

Software that asks a language model to route a ticket or rate an alert needs a probability distribution over options it names at request time, plus a usable way to say that none of them fit. Typical reads that distribution directly out of a pretrained causal language model, without generating a token. It encodes the state once into a key-value cache and renders each question with its runtime candidates as a short causal suffix, where a head at 71% of backbone depth scores the terminal decision state against each candidate's own contextual hidden state. Output types are contracts on the distribution: categorical Choice, ordinal Score, a Bernoulli yes/no head (Noul), and an abstention gate outside the candidate softmax. The contextual readout keeps the question-conditioned signal of the standard letter-logit interface (Δ_q^{sh} .106 vs. .109) with a quarter of its log-odds shift under candidate-set edits. We measure that signal with Δ_q, a shuffled-question control that separates question-conditioned capability from candidate-set priors, and use it to show that the capability lives in computation over the candidates: in all six configurations we tested, a decision state computed before the candidates are seen stays within .03 of zero, against .09–.12 for readouts over them. Training on counterfactual rubric groups, where only the rubric determines the answer, lifts held-out rule families by 35–40 points, and the full recipe cuts held-out ordinal NLL threefold at 14B. A decision costs 45 ms at 1.7B and 60 ms at 14B in a research harness, and 15.5–17 ms on the optimized 1.7B serving path; the 14B model reaches .931 on the standard tier of JevBench, a third-party typed-decision benchmark. A train-render $\times$ eval-render design on 605 held-out long documents shows that placing the evidence first in training states adds 11.2 points ($p = 5 \times 10^{-10}$), reproduced at a second seed. We release 1.7B and 4B models and an inference package.

1 Introduction

A growing share of language-model calls in production software ask for a decision: which of forty queues a ticket belongs in, whether a transaction is fraud, how severe an alert is, which tool an agent should call next. The caller needs a probability distribution over a label set it defines at request time, and it needs to know when none of the labels fit. The usual route to that distribution goes through text. The model generates an answer symbol or a JSON object, or the caller constrains decoding, or it compares candidate log-probabilities. Each of these inherits the costs of the generative interface: decoding latency, parsing, and a well-documented sensitivity to how the answer is written, including option-order and option-symbol biases [Zheng et al., 2023, Pezeshkpour and Hruschka, 2024] and first-token probabilities that disagree with the text the model goes on to produce [Wang et al., 2024a].

When a causal language model reads a question followed by a list of options, the winning option is already decodable from its hidden states before the model binds the answer symbol [Wong et al.,

Preprint.

2026]. Typical makes that internal decision the deployed interface. A call has the signature

$$(\text{state},\ \text{typed question},\ \text{runtime candidate set}) \ \longmapsto\ \big(P(a_1), \ldots, P(a_K),\ P(\varnothing)\big),$$

and nothing is generated or parsed. Typical encodes the state (a ticket, a policy document, an agent trace) once into a key-value cache. Each question against it is a short causal suffix containing the question and its candidates, and a head at 71% of the backbone's depth reads the terminal decision state against each candidate's own contextual hidden state. The output type is a contract on the returned distribution: categorical *Choice*, ordinal *Score*, a Bernoulli yes/no decision we call *Noul*, and an abstention event $\varnothing$ computed by a gate outside the candidate softmax (Figure 1).

We settled three design choices by measurement. First, the head reads each candidate's contextual state. Scoring the decision state against the candidates' contextual hidden states keeps the question-conditioned signal of the standard letter-logit interface (Δ_q^{sh} .106 vs. .109, within seed noise across two seeds), reduces the shift in pairwise log-odds under candidate-set edits to a quarter, and adds 2–7 points on natural language inference. Scoring the same decision state against an independent embedding of each option's text loses 12 points of knowledge accuracy (Section 6.1).

Second, question-conditioned capability lives in computation over the candidates. We measure it with Δ_q, the accuracy lost when the real question is replaced by a shuffled one while the state and candidate set stay fixed, which extends the choices-only control of Balepur et al. [2024]. It separates question-conditioned capability from candidate-set priors, a distinction that accuracy and calibration both miss. In every configuration we tested (six factorizations and objectives, two backbone depths), a decision state computed before the candidates are seen stays within .03 of zero, while readouts that process the candidates in context retain .09–.12, matching their listwise teacher (Section 6.2). Reading at 71% of depth yields one model that is strong on both evidence and knowledge; reading the final layer produces a model at the SNLI hypothesis-only baseline (Section 6.3).

Third, each output type is a contract that the head and the training target enforce. Abstention is a gate and never a rendered "none of the above" candidate; removing the rendered option removes its degenerate endpoints (never abstaining at 5 candidates, always abstaining at 150) and lowers MMLU-Pro false abstention from .692 to .003 (Section 6.4). An ordinal target cuts Score's held-out NLL from 2.07 to 1.23 at unchanged accuracy, and the Bernoulli head makes yes/no decisions invariant to label order by construction (Section 7.1).

Training data is as strong a lever as architecture. DecisionMix v2, a rule-engine curriculum in which every state and candidate set appears under several rubrics with different gold answers, makes the rubric the only way to fit the data; training on it lifts held-out families, grammars and rubric styles by 35–40 points. Balancing decision types and adding null-target copies of workflow rows each move a measurable axis, and at 14B the full recipe cuts held-out ordinal NLL from 2.87 to 0.95 (Sections 7.2 and 7.3). The cost stays low across scale: in a single-stream research harness a decision takes 45, 56 and 60 ms at 1.7B, 4B and 14B, so an $8\times$ larger backbone is $1.3\times$ slower, and the optimized serving path decides in 15.5–17 ms warm at 1.7B. The 14B model reaches .931 on the JevBench standard tier (Sections 7.4 and 7.5).

We also treat rendering, the way the state and options are laid out as text, as an experimental variable. A 2×2 design that crosses training render with evaluation render on 605 held-out long documents measures a data effect that a 19-item benchmark cell could not resolve: with training states right-truncated at 1,024 tokens, placing the evidence first adds 11.2 points ($p = 5 \times 10^{-10}$), reproduces at a second seed, and is corroborated at 14B (Section 7.6). The corrected checkpoints are the current public releases, at .947 (1.7B) and .950 (4B) on those documents.

Contributions.

1. A direct decision interface for pretrained causal language models: a cached-state decision pass with a contextual-candidate readout at mid-depth, typed output contracts and a factored abstention gate with exact structural properties, returning distributions over runtime candidates without generating tokens (Section 3).

2. An account of where the decision lives, built on Δ_q: matched comparisons of readouts, decision-state factorizations, depths and abstention designs that fix each component of the interface (Section 6).

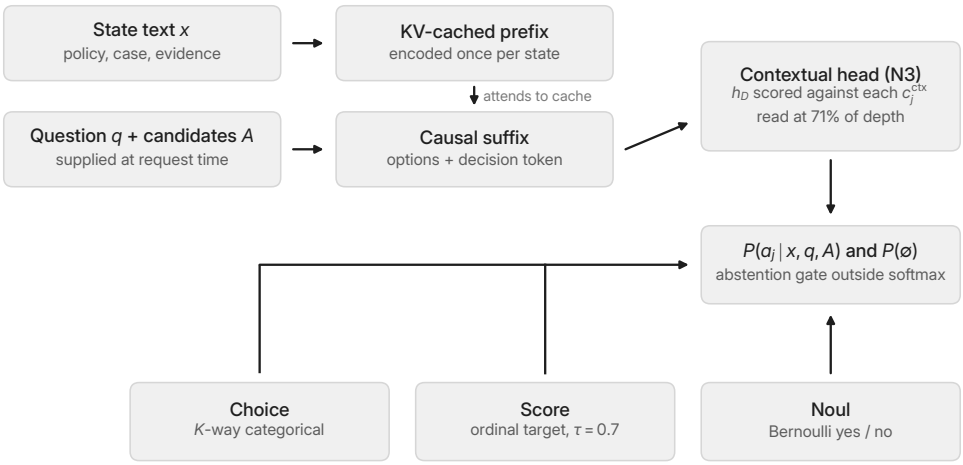

Figure 1: The decision pass. The state is encoded once into a key-value cache. Each question is a short causal suffix holding the question and its runtime candidates; the head reads the terminal decision state h_D against each candidate's pooled contextual state c_j^{ctx} at 71% of backbone depth. One readout serves three typed contracts (Choice, Score, Noul), and abstention is a separate gate outside the candidate softmax.

3. A training recipe for decision models: counterfactual rubric groups, decision-type balancing, a soft-target uncertainty corpus, null augmentation, evidence-first long states and likelihood-based checkpoint selection (Sections 4 and 7).
4. Render-crossed evaluation, which separates data effects from interface compatibility and measures how far rendering alone moves a benchmark score (Sections 7.6 and 7.7).
5. Open models and code: `typical-small` (Qwen3-1.7B) and `typical-medium` (Qwen3.5-4B), with a self-contained inference package (`pip install typical-ai`) whose outputs match the training-time decider exactly.

2 Related work

Decisions in hidden states. Hidden states carry linearly decodable structure [Alain and Bengio, 2016, Hewitt and Liang, 2019, Belinkov, 2022] that varies with depth [Tenney et al., 2019, Belrose et al., 2023] and anticipates tokens not yet emitted [Pal et al., 2023, Wu et al., 2024], and circuit analysis localizes where a chosen option binds to its symbol [Lieberum et al., 2023]. Wong et al. [2026] decode the winning option in content space before the symbol is bound, and Bhatt et al. [2026] trace this across languages. Typical deploys that pre-symbol decision as the interface, read at mid-depth, where upper layers are partly redundant and early exit saves compute [Gromov et al., 2024, Men et al., 2025, Schwartz et al., 2020].

Answer symbols and runtime labels. Answer symbols carry order and symbol biases [Zheng et al., 2023, Pezeshkpour and Hruschka, 2024, Yang et al., 2025, Xue et al., 2024], first-token probabilities disagree with generated text [Wang et al., 2024a], and constrained decoding [Geng et al., 2023, Willard and Louf, 2023] can cost accuracy [Tam et al., 2024]. Δ_q (Definition 5.1) extends the choices-only control of Balepur et al. [2024] to a shuffled real question. Runtime labels have been scored by joint embeddings, entailment, label-conditioned encoders and schemas [Pappas and Henderson, 2019, Yin et al., 2019, Halder et al., 2020, Zaratiana et al., 2024, 2025], by cross-encoders [Nogueira and Cho, 2019], dual encoders [Reimers and Gurevych, 2019], late interaction and decomposition [Humeau et al., 2019, Khattab and Zaharia, 2020, Cao et al., 2020a, MacAvaney et al., 2020, de Jong et al., 2023] and distilled students [Lu et al., 2022], while decoder classification heads fix

labels as indices [Li et al., 2023]. Typical renders all candidates in one causal suffix, so each carries its slot and its set, and Δ_q measures what candidate-independent scoring gives up (Section 6.2).

Computation per decision. Prefix sharing is established systems work [Juravsky et al., 2024, Zheng et al., 2024, Kwon et al., 2023, Gim et al., 2023, Cheng et al., 2023, Pope et al., 2022, Dao et al., 2022], and speculative decoding targets sequential decode overhead [Leviathan et al., 2022, Chen et al., 2023, Cai et al., 2024]; Typical removes the decode loop. Chain-of-thought supplies serial computation a fixed-depth pass lacks [Wei et al., 2022, Li et al., 2024, Merrill and Sabharwal, 2023b,a, Feng et al., 2023], mainly helping math and symbolic tasks [Sprague et al., 2024], as do pause tokens [Goyal et al., 2023, Pfau et al., 2024] and latent reasoning [Deng et al., 2023, 2024, Hao et al., 2024] (Section 8).

Output contracts and training. Score uses a soft ordinal target [McCullagh, 1980, Frank and Hall, 2001, Niu et al., 2016, Cao et al., 2020b, Shi et al., 2023b, Díaz and Marathe, 2019]; the reject option [Chow, 1970, Geifman and El-Yaniv, 2017, 2019], scored by confidence or energy [Hendrycks and Gimpel, 2016, Liu et al., 2020] and benchmarked on out-of-scope intents [Larson et al., 2019, Casanueva et al., 2020], becomes a gate outside the softmax (Section 3.6). Temperature scaling repairs much miscalibration [Guo et al., 2017], binned ECE is biased [Pakdaman Naeini et al., 2015, Kumar et al., 2019], and we evaluate with proper scoring rules [Gneiting and Raftery, 2007, BRIER, 1950, Epstein, 1969, MURPHY, 1970]. Label smoothing [Szegedy et al., 2016, Müller et al., 2019], distillation [Hinton et al., 2015, Kim and Rush, 2016, Yuan et al., 2020, Zhang and Sabuncu, 2020] and context distillation [Askell et al., 2021, Snell et al., 2022] shape probabilities, and fine-tuning distorts calibration [Desai and Durrett, 2020, Wang et al., 2025] and features [Kumar et al., 2022], rivals in-context learning [Mosbach et al., 2023], and under LoRA [Hu et al., 2021] learns and forgets less [Biderman et al., 2024] yet forgets under continual training [Luo et al., 2025]. Compositional benchmarks [Lake and Baroni, 2017, Keysers et al., 2019, Kim and Linzen, 2020, Hupkes et al., 2020] motivate our held-out composition level (Section 4.2); long-context sensitivity to evidence placement [Liu et al., 2024, Levy et al., 2024, Shi et al., 2023a, Hsieh et al., 2024] and silent pipeline defects [Sambasivan et al., 2021] motivate our render-crossed design (Section 7.6). JevBench targets closed-weight "System-One" decision models such as TypeSafe's Jev; among its open entries only GLiNER2 [Zaratiana et al., 2025] and one ModernBERT-based entry [Warner et al., 2025] have method papers.

3 Typical: the decision pass

Typical turns a pretrained causal language model into a decision function whose output is read from hidden states in one pass over a short suffix.

3.1 Typed decisions

Definition 3.1 (Typed decision)**.** A decision is a tuple $D = (x, q, A, t)$: a state x (the document, transcript or trace the decision is about), a question q (the instruction or rubric), a candidate set $A = \{a_1, \ldots, a_K\}$ of label strings supplied at request time, and a type $t \in \{\textsc{Choice}, \textsc{Score}, \textsc{Noul}\}$. The model returns

$$P(a_j \mid x, q, A),\ j = 1, \ldots, K, \qquad P(\varnothing) = P(\varnothing \mid x, q, A), \qquad P(\varnothing) + \textstyle\sum_j P(a_j \mid x, q, A) = 1,$$

and emits no token.

The type is a contract on the returned distribution, enforced by the head and the training target. *Choice* is categorical over A. *Score* is categorical over ordered levels, where near misses cost less than distant ones. *Noul* is a yes/no decision: a Bernoulli probability for one proposition, with $A = \{\text{yes}, \text{no}\}$, invariant to the order in which the caller lists the two labels. The abstention event $\varnothing$ is available under every type and is never one of the candidates. Because A is text supplied per call, the model has no output index for any label and must read the candidates to score them.

3.2 Cached state and causal suffix

Typical encodes the state once as an ordinary causal prefix and retains its key-value cache. Each question against it is a short causal suffix: the question text, the candidates rendered as lettered lines,

an answer cue, and a terminal decision token. The suffix attends to the cached state, and a batch of suffixes against one state runs in chunks without attending to one another. A decision is therefore a forward pass on $\mathrm{concat}(x, q, A)$ with the state half read from cache, which an equivalence test on the real model confirms (Appendix A.18).

3.3 The contextual-candidate readout

At the tap layer the suffix pass yields the terminal decision state h_D, the hidden state at the decision token, and for each candidate a contextual state c_j^{ctx}, the mean hidden state over the tokens of candidate j's rendered span. Candidate j receives the score

$$u = W_h h_D, \qquad v_j = W_c\, c_j^{\mathrm{ctx}}, \qquad s_j = \frac{u^\top v_j}{\sqrt{d_v}} + w^\top (u \odot v_j) + g(u \odot v_j), \tag{1}$$

where W_h and W_c are layer-normalized linear projections to width d_v, w is a vector and g a two-layer MLP: a bilinear form with a small learned correction on the elementwise product. The option letters stay in the rendered text and are never read out. Because the candidates are tokens in the same causal computation as the state and the question, c_j^{ctx} carries the candidate's meaning together with its slot and its set, and h_D has attended to all of them.

Section 6.1 compares three readouts. *N1*, the letter-logit readout, scores h_D against the output embeddings of the option letters (their next-token logits, with a softmax null), the standard multiple-choice interface read without generating. *N2*, the candidate-blind semantic readout, applies Equation (1) to an embedding of the candidate text from a frozen text-embedding model, which carries meaning and no slot identity. *N3*, the contextual-candidate readout Typical deploys, applies Equation (1) to c_j^{ctx}. N2 and N3 share the scorer, the factored null and the feature normalization and differ only in the candidate vector. N1 and N3 differ in the readout and in the null form native to each; option shuffling during training is shared by all three.

3.4 Mid-depth tap

We truncate the backbone at about 71% of its depth: the readout taps layer 20 of 28 at 1.7B, 26 of 36 at 4B and 8B, and 28 of 40 at 14B, and the layers above the tap are never computed. LoRA [Hu et al., 2021] at rank 16 adapts the top eight kept layers, and everything below is frozen. We chose the tap for what the representation holds: our first full-scale model read the final layer and scored 58.6 on SNLI, the hypothesis-only baseline, because it ignored the premise. The mid-depth tap carries evidence and question-conditioned knowledge in one model (Section 6.3), and every released and ladder model uses it.

3.5 Typed heads

The three types share the backbone and the tap and differ in the training target and, for Noul, the terminal head. *Choice* is the K-way readout of Equation (1), trained with cross-entropy against the gold distribution over $A \cup \{\varnothing\}$ through the gate of Section 3.6. *Score* uses the Choice readout over the rendered levels with an ordinal-smoothed target $\tilde{t}_j \propto \exp(-|j - y|/\tau)$ for gold level y, normalized over the valid levels, with $\tau = 0.7$; $\tau \to 0$ recovers the one-hot Choice target, and the change adds no parameters. *Noul* uses a dedicated Bernoulli head, $P(\text{yes}) = \sigma(w_n^\top h_D)$, on a query-only suffix that renders no candidate text, with the complement assigned to "no". Abstention on this path uses a gate of the form of Equation (2) with its own parameters, whose score statistics come from $\log P(\text{yes})$ and $\log P(\text{no})$ and whose state input is h_D, since no candidate span is rendered. Every gate input is symmetric in the two labels, so for rows routed to the Bernoulli head the returned yes, no and $\varnothing$ probabilities are exactly invariant to label order.

Routing is per row, because a batch mixes types. A row takes the Bernoulli path if and only if its candidate set is exactly $\{\text{yes}, \text{no}\}$, Score rows take the ordinal target, and every other row is scored by the Choice path, bit-identical to a model with the typed heads disabled (Appendix A.6).

3.6 Factored abstention gate

Abstention is a separate gate outside the candidate softmax. Given the scores s_j of Equation (1), the gate reads a vector z of permutation-invariant statistics over the valid candidates (maximum

score, top-2 margin, mean score, log-mean-exp of the scores), the projected decision state, and the mean and variance of the projected candidate states v_j, and computes $r = \sigma(g_\varnothing(z))$. The returned distribution is

$$P(a_j \mid x, q, A) = (1 - r)\,\frac{\exp(s_j/T)}{\sum_{k=1}^{K}\exp(s_k/T)}, \qquad P(\varnothing) = r, \tag{2}$$

implemented as logits $\ell_j = \log \mathrm{softmax}(s/T)_j + \log(1-r)$ and $\ell_\varnothing = \log r$, so one cross-entropy over $K+1$ outcomes trains both parts. Two properties follow exactly. (i) *Pairwise non-null odds are invariant to abstention*: for fixed scores $P(a_j)/P(a_k) = \exp((s_j - s_k)/T)$, independent of r, because $(1-r)$ multiplies every non-null candidate; the contextual scores themselves remain set-dependent. (ii) *The gate is temperature-invariant*: T enters only the candidate softmax and r reads untempered statistics, so recalibrating T never moves $P(\varnothing)$. The gate is symmetric in the pairs (s_j, v_j), so it adds no order dependence, while r depends on what was offered by design. We never render the null as a candidate line, because a rendered line competes for softmax mass that dilutes as K grows; Section 6.4 removes that line from a model that already has the gate and measures the difference.

3.7 Cost model

M decisions against one state cost $O(|x|) + M \cdot O(|q| + |A|)$ token positions, where $|A|$ counts the rendered candidate tokens, and every position runs through only the kept 71% of the layers. The state term is paid once and amortized over every question asked of it; the per-decision term grows with the question and the candidate text and is independent of the other questions. In our research harness on one H100 at $K = 2$, a decision takes 45, 56 and 60 ms at 1.7B, 4B and 14B, so an $8\times$ larger backbone is $1.3\times$ slower. At large K the rendered candidates dominate the per-decision term, and in throughput mode a candidate-blind scorer, whose cost does not grow with the rendered set, is cheaper per query by a margin that grows with K (Section 7.5).

4 Training decision models

Every row is a typed decision rendered as in Section 3.2 and trained with cross-entropy against a hard or soft gold distribution over $A \cup \{\varnothing\}$, plus distillation at 14B, using AdamW in bf16 at batch size 64 for 8,000–12,000 steps (Appendix A.8). Table 26 (Appendix A.17) lists the seven corpora, and Section 7 measures the six recipe choices below.

4.1 Family-balanced mixture

We draw batches by decision family, so no large corpus dominates by row count. E is evidence and classification (SNLI [Bowman et al., 2015], MNLI [Williams et al., 2018], ANLI [Nie et al., 2020a], BoolQ [Clark et al., 2019], and wide-vocabulary intent and topic sets including CLINC-150, Banking77 and HWU64 [Larson et al., 2019, Casanueva et al., 2020]), spanning many label vocabularies at few rows per source so that the model must read the rendered candidates. K is closed-book multiple-choice knowledge from nine sources, with TruthfulQA-MC1 [Lin et al., 2022] held out and a leakage audit against MMLU-Pro [Wang et al., 2024b, Hendrycks et al., 2020]. W is rubric-conditioned workflow decisions (routing, eligibility, extraction, urgency, tool selection, long policy documents), and U is soft-target uncertainty. The released recipe samples E:0.40, K:0.15, W:0.35, U:0.10. Unbalanced workflow data costs evidence accuracy, mostly through false abstention; with the U share and null augmentation, E:0.40 keeps evidence within about a point of the evidence-only base, where the sweep of Section 7.2, run without either, needs $E \geq .45$.

4.2 Counterfactual rubric groups

A workflow model that learns "this kind of ticket gets that label" fails the first time the policy changes. DecisionMix v2 (`data_wh`) is a programmatic rule-engine corpus of 60,605 rows over 12 domains and six boolean conditions at compositional depths 1–6, in which every row belongs to a *counterfactual rubric group*: the same state and candidate set under at least two rubrics with distinct gold answers. A predictor that ignores the rubric assigns one distribution to every member of a group, so on a two-member group with golds $y_1 \neq y_2$ it incurs an average cross-entropy of at least

$\log 2$ nats, and the loss falls below that floor only by reading the rubric. Depth-7 compositions, three whole domains, two rubric styles and one grammar per level are evaluation-only (Appendix A.9).

4.3 Soft-target uncertainty corpus

Decisions with a single correct answer cannot teach a model how much probability to spread. The uncertainty corpus `data_u` (31,029 rows, all soft targets) draws on the UNLI validation split [Chen et al., 2020], the ambiguous rows of AmbiEnt, and four synthetic generators with exact closed-form posteriors, each holding out a parameter range (Table 14); ChaosNLI [Nie et al., 2020b] is evaluation-only.

4.4 Null augmentation

A copy of 20% of W rows drops the gold candidate and any rendered catch-all option and takes $\varnothing$ as its target. The fix belongs in the data because the families disagree about abstention: under 1% of W rows carry a null target against over 20% of E rows, so a model trained on both learns two operating points that no single post-hoc temperature and null offset can serve (Sections 7.2 and 7.3).

4.5 Evidence-first long states

We first rendered the long-state corpus `data_wf_long` (23,318 multi-paragraph policy documents) with the case facts after the policy body, and training right-truncates states at the window, so at 1,024 tokens 98.8% of rows lost their facts (Appendix A.19). The fixed recipe renders facts first and drops any row longer than the window. The v2 releases train at a 2,048-token window and the 14B model at 3,072; Section 7.6 isolates the effect with a render-crossed design.

4.6 Selection on held-out likelihood

Selection on in-distribution loss favors the hard-label majority of the mixture over the soft and ordinal minority, so the v2 releases and the 14B model select checkpoints on held-out negative log-likelihood over the uncertainty and curriculum validation sets; the v1 releases do not. The 14B model also distills from its own frozen backbone under a fixed three-shot letter rendering, $\alpha \cdot \mathrm{CE} + \beta \cdot \mathrm{KL}$ at $T = 2$ on 63k labelled rows, and a 14B run without distillation reaches the same held-out Score NLL. Selection enters the 14B recipe together with evidence-first long states, the 3,072-token window, DecisionMix v2, the uncertainty corpus and the typed heads, and is never isolated; Section 7.3 reports what the bundle does.

5 Evaluation protocol

5.1 Evaluation sets

Training corpora are in Table 26. We evaluate on SNLI [Bowman et al., 2015], MNLI [Williams et al., 2018], ANLI [Nie et al., 2020a], BoolQ [Clark et al., 2019], CLINC-150 [Larson et al., 2019], TREC-fine, HWU64, 20 Newsgroups and Banking77 [Casanueva et al., 2020] against all 77 labels; a fixed 1,200-item MMLU-Pro probe [Wang et al., 2024b] scored among the rendered options (two leakage-matched items excluded) and TruthfulQA-MC1 [Lin et al., 2022]; held-out generator splits, including the never-trained DecisionMix level 7 (880 items, guess rate .441), and ChaosNLI [Nie et al., 2020b]; and 605 held-out long states (`policy_permit`, K=2, floor .612; `action_select`, K=4, floor .233). Evaluation-only external sets are PagerDuty (alert triage, yes/no, majority floor .792), jevlogs (log triage, floor .697), Mind2Web (web-agent actions, floor .427), tree-choice (taxonomy routing, $K \leq 320$), typed-decisions (400 cases of all three types, probabilistic gold) and JevBench.

5.2 What "held out" means

Each held-out split excludes a whole family, domain, rubric style, grammar or parameter range from training; none is a random validation split (axes in Table 4).

5.3 Δ_q: question dependence separated from candidate priors

Definition 5.1 (Question-dependence gap)**.** $\Delta_q = \mathrm{Acc}\big(f(x, q, A)\big) - \mathrm{Acc}\big(f(x, \tilde{q}, A)\big)$, where $\tilde{q}$ replaces the real question with the state and candidate set held fixed: an empty placeholder in the *choices-only* variant, a real unrelated question from the same corpus in the *shuffled-question* variant Δ_q^{sh}, primary throughout.

Δ_q extends the choices-only control of Balepur et al. [2024] to a shuffled real question, which is stricter: a placeholder can leak candidate priors through the rendering (on one candidate-blind model choices-only reads $+.017$ where Δ_q^{sh} reads .000). High accuracy with $\Delta_q^{\mathrm{sh}} \approx 0$ means that candidate-set priors explain the score. The reference is the listwise teacher, a fine-tuned decoder reading the options in context, at $\Delta_q^{\mathrm{sh}} = .102$ on the MMLU-Pro probe. $|\Delta_q^{\mathrm{sh}}| < .02$ (about 1.4 standard errors at n=1,200, $p \approx .15$) counts as zero.

5.4 Metrics

We report accuracy, Brier score, expected calibration error (ECE) and negative log-likelihood (NLL), with NLL and Brier primary on soft-gold sets. *Among-K accuracy* takes the argmax over candidates and ignores $P(\varnothing)$; the *false-abstain rate* is the fraction of gold-present items on which $\varnothing$ wins. *Rubric-flip* pairs share a state and candidate set under two rubrics with different gold. On MMLU-Pro, *reorder* Δp is the largest change in any candidate's probability under reordering, and *pairwise* Δ*log-odds* the shift in log-odds between the two top candidates when three unrelated options are added; both are zero for a set-invariant scorer.

5.5 JevBench protocol

JevBench (`fstandhartinger/jevbench` v1.2.1) is a third-party benchmark of yes/no, choice (K=2–6) and ordinal decisions, each with a state, a rubric and an exact label set, in three tiers: *standard* (72 items), *easy* (48) and *hard* (111, eight families including long policies, multi-hop, temporal, probability and trade-off). Its 146 judge-scored items are private and the harness ranks only submissions covering 95% of all items, so every JevBench number here is on the 231 public ids and unranked; no training corpus contains a public id. The 72 standard items are 36 states in two paraphrases sharing a `group`, so intervals are cluster bootstraps over `group` (20,000 resamples): ±6–13 points on standard, ±9 on hard. Comparisons between our models use paired per-item bootstraps; scoring details and leaderboard context are in Appendix A.13.

5.6 Hardware and latency

Models train on single H100 GPUs (Appendix A.8). The *research harness* times one in-process decision on a 256-token state, state encode included, for K from 2 to 256, all sizes on one pod and PyTorch/CUDA build. The *serving path* is the released inference package, timed as warm p50 against an already-cached 256-token state at K=2.

6 Where the decision lives

Unless stated otherwise, models here are 1.7B, trained for 12k steps on the evidence and knowledge mixture, and Δ_q^{sh} is measured on the MMLU-Pro probe against the teacher's .102.

6.1 The contextual-candidate readout

Scoring the decision state against each candidate's contextual state keeps the letter-logit readout's question-conditioned signal within seed noise, with a quarter of its pairwise log-odds shift and 2–7 more points on natural language inference. Table 1 compares the readouts of Section 3.3 on one pass: letter logits (N1), h_D against an independent embedding of each option's text (N2), and h_D against each option's contextual state c_j^{ctx} (N3). All arms render the candidates in the suffix, train on identical data with per-step option shuffling, and read all 28 layers. N1 and N3 differ in readout and in the null form native to each; N2 and N3 differ only in what h_D is scored against.

Table 1: Three readouts of one decision pass, full-depth 1.7B backbone, identical data and steps; N1 and N3 at two seeds (`s0`/`s1`), N2 at one. Teacher: a fine-tuned decoder reading the options in context. Order measures are zero for a set-invariant scorer.

	N1 s0	N1 s1	N2	N3 s0	N3 s1	Teacher
Δ_q^{sh} (primary)	.114	.104	.058	.117	.094	.102
MMLU-Pro among-K	.325	.318	.198	.314	.301	.310
Banking77, K=77	.531	–	.012	.497	–	–
SNLI	.838	.831	.854	.863	.851	–
MNLI	.684	.664	.742	.752	.719	–
ANLI	.377	.393	.410	.431	.437	–
BoolQ	.769	.764	.775	.737	.746	–
Reorder Δp	.165	.165	.059	.101	.109	.097
Pairwise Δlog-odds	.463	.443	.379	.126	.094	.179

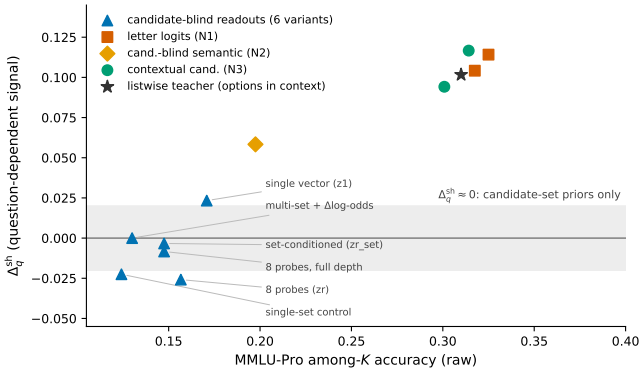

Figure 2: Raw MMLU-Pro among-K accuracy against Δ_q^{sh} (n=1,200; shaded: the $\pm$.02 band of Section 5.3). The six candidate-blind variants cluster at zero at every head capacity, objective and depth; the letter-logit (N1) and contextual (N3) readouts and the listwise teacher carry .09–.12, and the semantic readout N2 sits between. Accuracy alone does not separate the groups.

The candidate's contextual state carries the decision. N2 loses 10–12 points of among-K accuracy against N3 (.198 vs. .314/.301) and 13 against N1, half of the question-conditioned signal (Δ_q^{sh} .058 vs. .106 for N3 across seeds), and collapses on Banking77 at K=77 (.012 against .497 for N3 and .531 for N1). The pattern fits a decision state that encodes which rendered slot holds the answer: the letter embedding and the contextual state both carry slot identity, and an independently embedded label carries only its meaning.

N3 matches N1 on question dependence and is far less sensitive to the candidate set. Across seeds N1 averages $\Delta_q^{\mathrm{sh}} = .109$ and N3 .106 (teacher .102), within the .01–.02 seed spread of both; among-K accuracy is 1–2 points lower for N3. Adding three unrelated options shifts N3's pairwise log-odds by .09–.13 against .44–.46 for N1, a quarter of the shift, and reordering moves N3's probabilities by .10–.11 against .165. On evidence tasks N3 leads in both seeds by 2 points on SNLI, 5.5–7 on MNLI and 4–5 on ANLI, and trails by 2–3 on BoolQ. Option shuffling is shared, so the order-sensitivity difference comes from the readout together with its null form. N3 is the readout Typical deploys; exact order invariance is the Bernoulli head's contract (Section 7.1).

6.2 Candidate-blind decision states

In every configuration we tested, question-conditioned capability appeared only when the candidates were processed inside the forward pass. A candidate-blind variant computes a decision state $Z(x, q)$ before the candidates are seen, scores each candidate against it with a learned head, and is distilled from the listwise teacher. Six such variants, spanning three head factorizations, two distillation objectives and two tap depths, all have Δ_q^{sh} within .03 of zero, while N1, N3 and the teacher sit at .09–.12 and even N2 keeps .058 (Figure 2; per-variant numbers in Table 5, Appendix A.1).

No lever we varied moves Δ_q^{sh}. Cross-attention from the probes over all candidate tokens leaves it at $-.003$ while improving null quality (CLINC out-of-scope .568 $\rightarrow$ .727); supervising the change in the teacher's log-odds across seven candidate sets gives exactly zero; reading all 28 layers gives $-.008$. Two variants fall outside the $\pm.02$ band in opposite directions ($+.023$, $-.026$), the spread 1,200 items allow. One alternative we cannot exclude is optimization: the zero-initialized set-conditioning cross-attention stayed inert through 4k steps of direct supervision.

Accuracy hides the gap because candidate-blind states learn priors and calibration. Their accuracy of .12–.17 sits above the .10 chance rate and comes from which option strings tend to be correct together; ECE on the knowledge probe falls from .39 for an earlier baseline to .04. The question-dependent component lives in candidate-contextual computation, which sets the design principle for Typical: render the candidates inside the backbone's forward pass.

6.3 Mid-depth readout

Reading at 71% of depth gives one model that carries both evidence reasoning and question-conditioned knowledge. The first full-scale model in this project read the last layer and scored 58.6 on SNLI, the hypothesis-only baseline (65.3 frozen and 68.2 with LoRA at layer 20). Under a controlled comparison (head, data and steps fixed), reading N3 at layer 20 instead of 28 raises SNLI/MNLI/ANLI/BoolQ from .863/.752/.431/.737 to .906/.865/.519/.834, CLINC-150 from .740 to .862, and lowers validation NLL from .430 to .341, while Δ_q^{sh} stays at .118 against .117. Evidence accuracy lands within 1–4 points of a candidate-blind energy-scoring baseline on every evidence set (ANLI trails by 3.6) and 8 points above it on CLINC-150, so one model now serves the evidence and knowledge regimes that had needed separate models. For a candidate-blind state the same change moves nothing. That decision features sit below the final, next-token-specialized layers agrees with work on layer pruning and probing [Gromov et al., 2024, Men et al., 2025, Tenney et al., 2019, Belrose et al., 2023]. Every checkpoint we report uses the 71% tap: layer 20 of 28, 26 of 36 and 28 of 40.

6.4 Abstention as a gate

Abstention belongs in the gate alone. A rendered "none of the above" line competes for softmax mass with the real candidates, so its share depends on how many there are. We test this with a matched pair, both arms N3 at layer 20 with the factored gate of Equation (2), in which the treatment removes the rendered $\varnothing$ line. With the line, abstention has degenerate endpoints: on CLINC-150 with the gold label absent the model almost never abstains at $K{=}5$ ($P(\varnothing \mid \text{absent}) = .02$) and always abstains at $K{=}150$ (1.00) and on Banking77 at $K{=}77$, while its null AUROC is .97/.94/.91 at $K = 5/20/50$. Removing the line removes both endpoints: null AUROC is .98/.95/.92/.95 on CLINC-150 at $K = 5/20/50/150$ and .90/.83/.69 on Banking77 at $K = 5/20/77$. MMLU-Pro false abstention falls from .692 to .003 (far outside its seed-to-seed variation of .05–.10) and accuracy rises from .133 to .353, the KL divergence to the teacher on MMLU-Pro falls from 2.43 to .27, Banking77 with all 77 labels rises from .063 to .579, and CLINC out-of-scope accuracy from .639 to .798, with Δ_q^{sh} unchanged (.118 vs. .122) and evidence tasks level (Table 7). The price is 2–7 points on intent and topic label spaces (HWU64 .801 $\rightarrow$ .757, 20NG .589 $\rightarrow$ .515) and a rise in reorder Δp from .077 to .123. This configuration, contextual readout at 71% depth with a gate and no rendered $\varnothing$ line, is the architecture of every checkpoint in Section 7; null augmentation teaches the gate when to fire (Section 7.2).

7 Results

All checkpoints share the architecture fixed in Section 6.

7.1 Typed output contracts

Each output type delivers the property its contract names, with backbone, tap and readout fixed. We compare matched fine-tunes from one parent checkpoint, each trained for 3k steps on the rows of its own type, against a control that treats the same rows as plain Choice (full tables in Appendix A.6).

Score. The ordinal-smoothed target ($\tau = 0.7$) adds no parameters and improves every probability-quality measure on held-out urgency at flat accuracy (.505 → .508): NLL 2.07 → 1.23, Brier .79 → .68, ECE .35 → .19, ordinal MAE .58 → .55. On JevBench's 12 ordinal items the two models make identical per-item predictions, so the gain is in how mass is spread over adjacent levels. On the held-out hard split of a trained typed-decision source (`systemone-lite`) it also lifts accuracy from .853 to .939 and cuts MAE from .15 to .06.

Noul. The Bernoulli head renders no candidates, so label order cannot enter its computation. Reversing the labels of a two-way Choice over {no, yes} moves $P(\text{yes})$ by up to .55 (mean .01–.15 across sets); for rows routed to the Bernoulli head (candidate set exactly {yes, no}) it moves by exactly zero on all nine test sets, and the released v1 checkpoint shows zero on 10 of 11 sets and .009 on one. The head is better or equal on six of seven yes/no sets and clears the untouched PagerDuty floor by a wide margin: accuracy .602 → .886 against a .792 majority floor, NLL 1.04 → .28.

7.2 The training distribution

Decision-type balance, counterfactual rubric groups and null targets each move a separate axis. All comparisons here are matched 1.7B runs that differ only in data or sampler.

Family balance. Adding workflow data at family weights E .35 / K .25 / W .40 cost 11 points on CLINC-150 and 10 on TREC-fine, almost all of it false abstention (CLINC-150 .05 → .17). The release sampler, E .40 with U .10, keeps every evidence set within about 1 point of the evidence-only base (sweep in Appendix A.5).

Counterfactual rubric groups. Training on DecisionMix v2, whose every row sits in a group that shares a state and candidate set across rubrics with different gold (Section 4.2), lifts held-out families, grammars and rubric styles by 35–40 points over a model never trained on the generator (.481/.507/.491 → .827/.881/.888, together with the uncertainty corpus), and accuracy on held-out rubric-flip pairs from .477 to .710. Accuracy under a shuffled rubric falls from .446 to .405: the model now depends on the rubric it is given. Level-7 composition, which the generator never produces, moves from .477 to .498 (Section 9). The original workflow sets give up 3–5 points at a fixed workflow share now split over four corpora.

Null augmentation. Duplicating 20% of workflow rows with the gold candidate removed and target ∅ lowers false abstention on CLINC-150 from .10 to .08 and on TREC-fine from .27 to .12, and raises accuracy by 2.6, 3.6, 4.3 and 6.0 points on CLINC-150, TREC-fine, HWU64 and 20NG. MMLU-Pro among-K returns to the evidence-only base (.353), with the highest Δ_q^{sh} of any mixture (.143). The model now spends mass on ∅ where soft gold has none, so held-out Noul drops 4 points and held-out Score NLL rises (1.52 → 2.04). Fractions in [.10, .20] give the same model within noise. The uncertainty corpus trades top-1 accuracy for likelihood (ChaosNLI NLL 1.27 → 1.00, accuracy .584 → .537; Appendix A.7).

7.3 Probability quality

Probability quality is indexed by decision type, so it is set by the training data, the target and checkpoint selection. A global temperature and null-logit offset cannot supply it: fit on held-out soft-target rows, they want opposite corrections for an evidence-and-knowledge checkpoint and a workflow-trained one (Appendix A.4).

The 14B recipe bundle takes held-out Score NLL from 2.87 to 0.95. Under the earlier workflow recipe, with selection on in-distribution validation loss, held-out Score NLL was 2.03, 2.15, 1.67 and 2.87 at 1.7B, 4B, 8B and 14B. The bundle applied to the same 14B backbone (facts-first long states with truncated rows dropped, a 3,072-token window, DecisionMix v2 and the uncertainty corpus, typed heads with the ordinal target, checkpoint selection on held-out uncertainty and curriculum NLL, and frozen-teacher distillation) reaches held-out Score NLL 0.95 and typed-decisions NLL 1.04 (from 1.96), with JevBench hard-tier Brier .85 → .66 (Figure 7, Appendix A.4). No component is isolated except distillation, and the arm without it reaches the same Score NLL. The v1 1.7B and 4B checkpoints, which have the ordinal target and the uncertainty corpus but no likelihood-

Table 2: Trained checkpoints with frozen controls. Recipe: *v1* = v1 release recipe (1,024-token window, no likelihood-based selection); *wf* = earlier workflow recipe (no DecisionMix, uncertainty corpus or typed heads); *fixed* = facts-first long states, truncated rows dropped, likelihood-based selection (+KD at 14B). JevBench (public subset); 95% cluster-bootstrap intervals (Section 5.5). MMLU-K: MMLU-Pro among-K. NLL: held-out Score. ms: research-harness single decision at $K{=}2$, one pod and build (the 8B was not on that pod). Frozen: no training, letter logits, 3 exemplars. Checkpoints top to bottom: `ts1b`, `ts1c`, `tm1b`, `ladder_8b`, `tl1b`, `tm2`.

		JevBench					
Model	Recipe	std [95% CI]	hard [95% CI]	MMLU-K	Δ_q^{sh}	NLL	ms
Qwen3-1.7B (small v1)	v1	.694 [.569, .819]	.432 [.342, .523]	.343	.127	1.01	45
small v2	fixed	.708 [.569, .833]	.432 [.342, .523]	.338	.127	1.08	–
frozen, 3-shot	–	.528 [.389, .667]	.369 [.279, .459]	–	–	–	–
Qwen3-4B (medium v1)	v1	.806 [.694, .903]	.423 [.333, .514]	.458	.193	1.03	56
frozen, 3-shot	–	.778 [.667, .889]	.441 [.351, .532]	–	–	–	–
Qwen3-8B	wf	.667 [.528, .792]	.396 [.306, .486]	.302	.093	1.67	–
frozen, 3-shot	–	.556 [.417, .694]	.369 [.279, .459]	–	–	–	–
Qwen3-14B	fixed	**.931** [.861, .986]	.450 [.360, .541]	**.498**	**.235**	**0.95**	60
frozen, 3-shot	–	.819 [.708, .917]	.559 [.468, .649]	–	–	–	–
Qwen3.5-4B (medium v2)	fixed	.861 [.778, .944]	.495 [.405, .586]	.429	.194	0.97	–
frozen, 3-shot	–	.764 [.639, .875]	.495 [.405, .586]	–	–	–	–

based selection, already reach 1.01 and 1.03; all 1.7B and 4B checkpoints with typed heads and DecisionMix sit at 0.97–1.08.

7.4 Scaling

Table 2 pairs each trained checkpoint with a frozen control: the same base model, untrained, reading next-token letter logits over the rendered options with three exemplars. The rows are not one recipe; the caption names each row's recipe, and Figure 3 (Appendix A.1) plots the same numbers against backbone size.

Every trained checkpoint beats its frozen backbone on the standard tier, by 16.7, 2.8, 11.1 and 11.1 points at 1.7B (v1), 4B, 8B and 14B. Each gap on its own sits inside its interval, and the paired test at 14B gives frozen-minus-trained $-.111$ $[-.250, +.014]$, $p = .10$. The 14B checkpoint reaches .931 on the standard tier. Question-conditioned knowledge grows with the backbone: Δ_q^{sh} rises from .127 at 1.7B to .235 at 14B, 2.3 times the 1.7B teacher's; the 8B checkpoint is the exception (.093). Under one matched recipe (the earlier workflow recipe run at 4B, 8B and 14B) the 8B is also below the 4B and 14B (.667 vs. .833/.875), as is its frozen control (.556 vs. .778/.819). The second-generation backbone is the strongest point per parameter: Qwen3.5-4B, released as `typical-medium` v2, scores .861 standard and .495 hard, above every Qwen3 checkpoint we trained on the hard tier and 9.7 standard-tier points above its own frozen control. Section 9 discusses the hard tier, where the frozen 14B leads.

7.5 Cost

Decision latency is nearly flat in backbone size. On the research harness a single decision at $K{=}2$ costs 45, 56 and 60 ms at 1.7B, 4B and 14B: an $8\times$ larger backbone is $1.3\times$ slower, because the cached state and a short suffix dominate. The cost stays flat to $K{=}32$ (46, 56 and 62 ms) and reaches 106, 118 and 157 ms at $K{=}256$ (Table 25). With $M{=}32$ questions against one cached state, the batched marginal cost per question is 2.7 and 3.9 ms at $K{=}2$ for 1.7B and 14B, and 28.0 and 109.7 ms at $K{=}256$. In that throughput mode a candidate-blind energy scorer costs about 1 ms per question at every K and is more than $3\times$ cheaper than the contextual readout from $K{=}32$ on, measured on an earlier full-depth prototype (Appendix A.2).

On the serving path a warm decision takes 15.5–17 ms at 1.7B and 19–21 ms for the Qwen3-4B model (warm p50, $K{=}2$, Figure 6); the Qwen3.5-4B model measures 34–46 ms with reference DeltaNet kernels. The serving-path engineering is in Appendix A.3.

Table 3: Training render × evaluation render on 605 held-out long states, no truncation at evaluation. Diagonal (matched) cells score each model in its training render; off-diagonal cells hold the evaluation render fixed across the two models. Majority floors: `policy_permit` .612 (K=2, n=330), `action_select` .233 (K=4, n=275). 14B arms: `tl1b_nokd` (facts-first) and `ladder_14b` (facts-last).

Trained on	Scored on	overall	`policy_permit`	`action_select`
1.7B, single-variable arms (seed 0)				
facts-first	facts-first *(matched)*	**.942**	**.933**	**.953**
facts-first	facts-last	.797	.788	.807
facts-last	facts-first	.640	.615	.669
facts-last	facts-last *(matched)*	.830	.842	.815
14B corroboration				
facts-first	facts-first *(matched)*	**.997**	.997	.996
facts-first	facts-last	.921	.933	.905
facts-last	facts-first	.851	.839	.865
facts-last	facts-last *(matched)*	.919	.930	.905

7.6 Evidence placement in long states

Training with the evidence placed first adds 11.2 points on held-out long states when training right-truncates states at 1,024 tokens, and the gap replicates at a second seed. With the case facts after the policy text, 98.8% of long-policy rows lost their facts at that window (Section 4.5). Two 1.7B arms with byte-identical flags train on the same 23,318 rows, rendered facts-first or facts-last, with truncation active as at release, and are scored on 605 held-out long states under both renders with no truncation at evaluation (Appendix A.14). The comparison is a *matched-configuration gap*: each model is scored in the render it was trained on, so the evaluation render changes along with training; the off-diagonal cells of Table 3 give the fixed-render contrasts.

At 1.7B the matched-configuration gap is $+.112$ $[+.077, +.147]$, $p = 5 \times 10^{-10}$ (`policy_permit` $+.091$ $[+.043, +.139]$, `action_select` $+.138$ $[+.086, +.190]$), and a second seed of the whole pair gives $+.111$ (.937 vs. .826), with every cell within two points of seed 0. At 14B the gap is $+.078$ $[+.055, +.100]$, $p = 7 \times 10^{-12}$, smaller because the facts-first model is at ceiling (.997). The two 14B checkpoints differ in the rest of the recipe bundle as well, so the 14B pair corroborates the direction at scale and the 1.7B pair is the single-variable test.

The fixed checkpoints are the current releases. On facts-first long states the v2 releases score .947 (1.7B) and .950 (4B) against .598 and .612 for v1, and hold their facts-last accuracy (Table 20). JevBench does not register the change: `typical-small` v2 and v1 read .708 and .694 on its standard tier. Its 19-item `long_policy` family gives 6/19 against 2/19 ($p = .232$) at one seed and 5/19 against 5/19 at the other (Appendix A.10).

7.7 Rendering as an experimental factor

Changing only the rendering moves a frozen 9B model 12.5 points. Scoring identical JevBench items with identical frozen Qwen3.5 checkpoints and the same letter-logit readout, changing only the layout of the state and options to a replication of another published system's format, lifts the standard tier by 1.4, 8.3 and 12.5 points at 2B, 4B and 9B (Table 8, Appendix A.1). The effect survives a paired per-item test at 4B ($+.083$ $[+.014, +.167]$, $p = .048$) and 9B ($+.125$ $[+.056, +.208]$, $p < .001$); no hard-tier change separates at any size.

At 1.7B, render mismatch inside one trained model costs more than the training defect. In the off-diagonal cells of Table 3, where the evidence is fully visible and only its position changes, switching the evaluation render against the training render costs 14.5 points for the facts-first 1.7B model and 19.0 for the facts-last one, which lands on its `policy_permit` floor (.615 against .612). At 14B the same switch costs 7.6 and 6.8 points, so the larger backbone tolerates render mismatch better. The earlier-recipe 14B scores .053 on JevBench's 19-item `long_policy` family and .919 on the 605 held-out long states in its own training render, a pair that also changes the item set and stacks truncation damage, render mismatch and a 19-item sample. The frozen controls of Table 2 use

a letter-logit rendering we did not sweep, so frozen-versus-trained comparisons carry a rendering factor of the size shown here.

8 Discussion

Where the decision lives. Question-conditioned capability appeared only when the candidates were processed inside the language model's forward pass (Section 6.2); among readouts that process them, the candidate's contextual state carries the decision and an independently embedded copy of the same text carries much less of it (Section 6.1); and the layer that holds it sits below the top of the stack (Section 6.3). This refines the familiar cross-encoder versus dual-encoder trade-off. In every configuration we tested, a candidate-independent state reproduced the teacher's candidate priors and calibration, and the question-dependent component appeared only with the candidates in context. Candidate-independent scoring keeps a clear role as a shortlisting front end for large candidate sets (Section 7.5), and Δ_q is the measurement that tells the two roles apart. We recommend reporting it for any architecture that scores candidates against a precomputed representation.

Types are enforced by structure. Each typed primitive fixes a property that post-hoc calibration cannot reach. The Bernoulli head has no candidate order to be sensitive to; the ordinal target moves probability mass toward adjacent levels without changing accuracy; keeping abstention out of the rendered candidates removes its dependence on how many candidates were offered. The calibration fit agrees: one global temperature and null offset, fit on held-out soft targets, wants opposite corrections for an evidence-trained and a workflow-trained checkpoint (Section 7.3). Probability quality for a decision model is indexed by decision type, and the head, the target and the training data shape it.

Data and selection as levers. Several of the largest effects in this paper hold the architecture fixed: the DecisionMix v2 curriculum (35–40 points on held-out families, grammars and styles), null augmentation (2.6–6.0 points of label-space accuracy), family-balanced sampling (evidence and intent accuracy back to within 1 and 3–4 points of the evidence-only base after an unbalanced mix cost 10–11), evidence-first long states (11.2 points), and the 14B recipe as a whole (a threefold reduction in held-out ordinal NLL). A decision model is trained on a mixture of decision shapes, and the composition of that mixture deserves the same experimental care as the architecture.

Measuring interfaces. Changing only the rendering moves a frozen 9B model 12.5 points on the same items (Section 7.7), more than the gap between many systems a benchmark is used to rank. Two practices made our own measurements decisive: crossing training render with evaluation render, so that data effects and compatibility effects separate, and building evaluation sets with the power to resolve the effect in question. Our 605 generated, leak-checked long states settled at $p < 10^{-9}$ a question that a 19-item benchmark cell returned at $p = .232$, so a pre-registered decision rule is limited by the power of the metric it names.

What a single pass computes. Typical decides in one forward pass of fixed depth, with no intermediate tokens. Fixed-depth transformers correspond to constant-depth threshold circuits [Merrill and Sabharwal, 2023a], and conditioning on self-generated intermediate tokens extends what they can compute [Merrill and Sabharwal, 2023b, Feng et al., 2023]. The decisions that remain hardest for the models we trained (temporal arithmetic, unit conversion, expected value, multi-attribute trade-offs; Appendix A.10) have the serial shape that this theory singles out, which suggests a concrete next step: keep the typed readout and the cached state, and add a bounded amount of intermediate computation before the decision token. The interface in this paper is a natural terminal layer for such a system, since the result still has to be a typed, thresholdable distribution over the options the caller supplied.

9 Limitations

Scope. Typical makes text-only, bounded decisions over candidate sets that fit in a short suffix: categorical, ordinal and Bernoulli contracts with abstention. It does not plan, search, call tools or read images. Per-query cost grows with the rendered candidate tokens, so candidate sets in the hundreds are served through a shortlisting front end (Section 7.5).

Multi-step composition. Decisions that require carrying an intermediate value through several steps remain the main open problem. On DecisionMix level 7 (temporal, numeric, expected-value and trade-off composition, never trained) the v1 1.7B and 4B models score .493 and .544 against a .441 guess rate, and the 14B model .620. On the JevBench hard tier a frozen 14B backbone read with three in-context exemplars scores above our trained 14B checkpoint (paired difference $+.108$ $[+.027, +.189]$, $p = .015$), while the trained model leads on the standard tier. Appendix A.10 gives the per-family breakdown, and Section 8 the extension this points to.

Statistics. The readout comparison in Section 6.1 and the evidence-first result in Section 7.6 are replicated across seeds; other rows are single training runs. JevBench numbers use the 231 public items and are not a ranked leaderboard entry. Its intervals are ± 6–13 points on the standard tier (36 independent states) and ± 9 points on the hard tier (111 items), so comparisons between our own checkpoints rest on paired tests and on larger purpose-built sets.

Generalization and invariance. "Runtime-defined labels" refers to the axes of Table 4. The rubric-generalization results share one rule engine between training and held-out families. Choice is order-robust rather than order-invariant: per-step option shuffling during training contributes to its robustness, and only Noul is invariant by construction. The pairwise-odds property of Equation (2) holds among non-null candidates; the abstention gate is set-dependent by design.

Ladder and release. The size ladder spans Qwen3 1.7B–14B plus Qwen3.5-4B, and its rows come from the recipe current when each was trained (Table 2); the Qwen3-8B checkpoint sits below its neighbours in both frozen and trained form. Inference code, weights and evaluation outputs are public; the training code is not yet released.

10 Conclusion

Typical reads decisions directly out of a pretrained causal language model. It encodes a state once and reads each question with its runtime candidates in a short suffix, where a head at mid-depth returns a typed distribution with a factored abstention event, at 45 ms per decision at 1.7B and 60 ms at 14B. The design rests on measured answers to where the decision lives: inside the forward pass over the candidates, in their contextual states, below the layers that specialize for generation. The training recipe shows that decision types, counterfactual rubrics and abstention targets shape a decision model as strongly as its architecture, and render-crossed evaluation separates such data effects from interface compatibility. The models and the inference package are public, and the same interface is a natural terminal layer for decision systems that add intermediate computation before the typed readout.

References

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes, 2016. URL `https://arxiv.org/abs/1610.01644`.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Jackson Kernion, Kamal Ndousse, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, and Jared Kaplan. A general language assistant as a laboratory for alignment, 2021. URL `https://arxiv.org/abs/2112.00861`.

Nishant Balepur, Abhilasha Ravichander, and Rachel Rudinger. Artifacts or abduction: How do llms answer multiple-choice questions without the question? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10308–10330. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.555. URL `http://dx.doi.org/10.18653/v1/2024.acl-long.555`.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022. ISSN 1530-9312. doi: 10.1162/coli_a_00422. URL `http://dx.doi.org/10.1162/coli_a_00422`.

Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens, 2023. URL https://arxiv.org/abs/2303.08112.

Divyanshu Bhatt, Abhinav Joshi, Ujjaval Patel, and Ashutosh Modi. From representation to choice: Tracing decision emergence across languages in LLMs. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens, editors, *Findings of the Association for Computational Linguistics: ACL 2026*, pages 42423–42442, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.2105. URL https://aclanthology.org/2026.findings-acl.2105/.

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and John P. Cunningham. Lora learns less and forgets less, 2024. URL https://arxiv.org/abs/2405.09673.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642. Association for Computational Linguistics, 2015. doi: 10.18653/v1/d15-1075. URL http://dx.doi.org/10.18653/v1/d15-1075.

GLENN W. BRIER. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, January 1950. ISSN 1520-0493. doi: 10.1175/1520-0493(1950)078<0001:vofeit>2.0.co;2. URL http://dx.doi.org/10.1175/1520-0493(1950)078<0001:vofeit>2.0.co;2.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D. Lee, Deming Chen, and Tri Dao. Medusa: Simple llm inference acceleration framework with multiple decoding heads, 2024. URL https://arxiv.org/abs/2401.10774.

Qingqing Cao, Harsh Trivedi, Aruna Balasubramanian, and Niranjan Balasubramanian. Deformer: Decomposing pre-trained transformers for faster question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4497. Association for Computational Linguistics, 2020a. doi: 10.18653/v1/2020.acl-main.411. URL http://dx.doi.org/10.18653/v1/2020.acl-main.411.

Wenzhi Cao, Vahid Mirjalili, and Sebastian Raschka. Rank consistent ordinal regression for neural networks with application to age estimation. *Pattern Recognition Letters*, 140:325–331, December 2020b. ISSN 0167-8655. doi: 10.1016/j.patrec.2020.11.008. URL http://dx.doi.org/10.1016/j.patrec.2020.11.008.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.nlp4convai-1.5. URL http://dx.doi.org/10.18653/v1/2020.nlp4convai-1.5.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling, 2023. URL https://arxiv.org/abs/2302.01318.

Tongfei Chen, Zhengping Jiang, Adam Poliak, Keisuke Sakaguchi, and Benjamin Van Durme. Uncertain natural language inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8772–8779. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.774. URL http://dx.doi.org/10.18653/v1/2020.acl-main.774.

Zhoujun Cheng, Jungo Kasai, and Tao Yu. Batch prompting: Efficient inference with large language model apis. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 792–810. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-industry.74. URL http://dx.doi.org/10.18653/v1/2023.emnlp-industry.74.

C. Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1):41–46, January 1970. ISSN 1557-9654. doi: 10.1109/tit.1970.1054406. URL http://dx.doi.org/10.1109/tit.1970.1054406.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1300. URL http://dx.doi.org/10.18653/v1/n19-1300.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *Advances in Neural Information Processing Systems 35*, NeurIPS 2022, pages 16344–16359. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2022. doi: 10.52202/068431-1189. URL http://dx.doi.org/10.52202/068431-1189.

Michiel de Jong, Yury Zemlyanskiy, Nicholas FitzGerald, Joshua Ainslie, Sumit Sanghai, Fei Sha, and William Cohen. Pre-computed memory or on-the-fly encoding? a hybrid approach to retrieval augmentation makes the most of your compute, 2023. URL https://arxiv.org/abs/2301.10448.

Yuntian Deng, Kiran Prasad, Roland Fernandez, Paul Smolensky, Vishrav Chaudhary, and Stuart Shieber. Implicit chain of thought reasoning via knowledge distillation, 2023. URL https://arxiv.org/abs/2311.01460.

Yuntian Deng, Yejin Choi, and Stuart Shieber. From explicit cot to implicit cot: Learning to internalize cot step by step, 2024. URL https://arxiv.org/abs/2405.14838.

Shrey Desai and Greg Durrett. Calibration of pre-trained transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 295–302. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.21. URL http://dx.doi.org/10.18653/v1/2020.emnlp-main.21.

Raúl Díaz and Amit Marathe. Soft labels for ordinal regression. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4733–4742. IEEE, 2019. doi: 10.1109/cvpr.2019.00487. URL http://dx.doi.org/10.1109/cvpr.2019.00487.

Edward S. Epstein. A scoring system for probability forecasts of ranked categories. *Journal of Applied Meteorology*, 8(6):985–987, December 1969. ISSN 0021-8952. doi: 10.1175/1520-0450(1969)008<0985:assfpf>2.0.co;2. URL http://dx.doi.org/10.1175/1520-0450(1969)008<0985:assfpf>2.0.co;2.

Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. Towards revealing the mystery behind chain of thought: A theoretical perspective, 2023. URL https://arxiv.org/abs/2305.15408.

Eibe Frank and Mark Hall. *A Simple Approach to Ordinal Classification*, pages 145–156. Springer Berlin Heidelberg, 2001. ISBN 9783540447955. doi: 10.1007/3-540-44795-4_13. URL http://dx.doi.org/10.1007/3-540-44795-4_13.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks, 2017. URL https://arxiv.org/abs/1705.08500.

Yonatan Geifman and Ran El-Yaniv. Selectivenet: A deep neural network with an integrated reject option, 2019. URL https://arxiv.org/abs/1901.09192.

Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. Grammar-constrained decoding for structured nlp tasks without finetuning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10932–10952. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.674. URL http://dx.doi.org/10.18653/v1/2023.emnlp-main.674.

In Gim, Guojun Chen, Seung-seob Lee, Nikhil Sarda, Anurag Khandelwal, and Lin Zhong. Prompt cache: Modular attention reuse for low-latency inference, 2023. URL https://arxiv.org/abs/2311.04934.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, March 2007. ISSN 1537-274X. doi: 10.1198/016214506000001437. URL http://dx.doi.org/10.1198/016214506000001437.

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. Think before you speak: Training language models with pause tokens, 2023. URL https://arxiv.org/abs/2310.02226.

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A. Roberts. The unreasonable ineffectiveness of the deeper layers, 2024. URL https://arxiv.org/abs/2403.17887.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks, 2017. URL https://arxiv.org/abs/1706.04599.

Kishaloy Halder, Alan Akbik, Josip Krapac, and Roland Vollgraf. Task-aware representation of sentences for generic text classification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3202–3213. International Committee on Computational Linguistics, 2020. doi: 10.18653/v1/2020.coling-main.285. URL http://dx.doi.org/10.18653/v1/2020.coling-main.285.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space, 2024. URL https://arxiv.org/abs/2412.06769.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. 2016. doi: 10.48550/ARXIV.1610.02136. URL https://arxiv.org/abs/1610.02136.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2020. URL https://arxiv.org/abs/2009.03300.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743. Association for Computational Linguistics, 2019. doi: 10.18653/v1/d19-1275. URL http://dx.doi.org/10.18653/v1/d19-1275.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models?, 2024. URL https://arxiv.org/abs/2404.06654.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring, 2019. URL https://arxiv.org/abs/1905.01969.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research*, 67:757–795, April 2020. ISSN 1076-9757. doi: 10.1613/jair.1.11674. URL http://dx.doi.org/10.1613/jair.1.11674.

Jordan Juravsky, Bradley Brown, Ryan Ehrlich, Daniel Y. Fu, Christopher Ré, and Azalia Mirhoseini. Hydragen: High-throughput llm inference with shared prefixes, 2024. URL https://arxiv.org/abs/2402.05099.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. Measuring compositional generalization: A comprehensive method on realistic data, 2019. URL https://arxiv.org/abs/1912.09713.

Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert, 2020. URL https://arxiv.org/abs/2004.12832.

Najoung Kim and Tal Linzen. Cogs: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9087–9105. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.731. URL http://dx.doi.org/10.18653/v1/2020.emnlp-main.731.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327. Association for Computational Linguistics, 2016. doi: 10.18653/v1/d16-1139. URL http://dx.doi.org/10.18653/v1/d16-1139.

Ananya Kumar, Percy Liang, and Tengyu Ma. Verified uncertainty calibration, 2019. URL https://arxiv.org/abs/1909.10155.

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution, 2022. URL https://arxiv.org/abs/2202.10054.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, pages 611–626. ACM, October 2023. doi: 10.1145/3600006.3613165. URL http://dx.doi.org/10.1145/3600006.3613165.

Brenden M. Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. 2017. doi: 10.48550/ARXIV.1711.00350. URL https://arxiv.org/abs/1711.00350.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1311–1316. Association for Computational Linguistics, 2019. doi: 10.18653/v1/d19-1131. URL http://dx.doi.org/10.18653/v1/d19-1131.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding, 2022. URL https://arxiv.org/abs/2211.17192.

Mosh Levy, Alon Jacoby, and Yoav Goldberg. Same task, more tokens: the impact of input length on the reasoning performance of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15339–15353. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.818. URL http://dx.doi.org/10.18653/v1/2024.acl-long.818.

Zhiyuan Li, Hong Liu, Denny Zhou, and Tengyu Ma. Chain of thought empowers transformers to solve inherently serial problems, 2024. URL https://arxiv.org/abs/2402.12875.

Zongxi Li, Xianming Li, Yuzhang Liu, Haoran Xie, Jing Li, Fu-lee Wang, Qing Li, and Xiaoqin Zhong. Label supervised llama finetuning, 2023. URL https://arxiv.org/abs/2310.01208.

Tom Lieberum, Matthew Rahtz, János Kramár, Neel Nanda, Geoffrey Irving, Rohin Shah, and Vladimir Mikulik. Does circuit analysis interpretability scale? evidence from multiple choice capabilities in chinchilla, 2023. URL https://arxiv.org/abs/2307.09458.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.229. URL http://dx.doi.org/10.18653/v1/2022.acl-long.229.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. ISSN 2307-387X. doi: 10.1162/tacl_a_00638. URL http://dx.doi.org/10.1162/tacl_a_00638.

Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. Energy-based out-of-distribution detection, 2020. URL https://arxiv.org/abs/2010.03759.

Yuxiang Lu, Yiding Liu, Jiaxiang Liu, Yunsheng Shi, Zhengjie Huang, Shikun Feng Yu Sun, Hao Tian, Hua Wu, Shuaiqiang Wang, Dawei Yin, and Haifeng Wang. Ernie-search: Bridging cross-encoder with dual-encoder via self on-the-fly distillation for dense passage retrieval, 2022. URL https://arxiv.org/abs/2205.09153.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*, 33:3776–3786, 2025. ISSN 2998-4173. doi: 10.1109/taslpro.2025.3606231. URL http://dx.doi.org/10.1109/taslpro.2025.3606231.

Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, Nazli Goharian, and Ophir Frieder. Efficient document re-ranking for transformers by precomputing term representations. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, pages 49–58. ACM, 2020. doi: 10.1145/3397271.3401093. URL http://dx.doi.org/10.1145/3397271.3401093.

Peter McCullagh. Regression models for ordinal data. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 42(2):109–127, January 1980. ISSN 1467-9868. doi: 10.1111/j.2517-6161.1980.tb01109.x. URL http://dx.doi.org/10.1111/j.2517-6161.1980.tb01109.x.

Xin Men, Mingyu Xu, Qingyu Zhang, Qianhao Yuan, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. Shortgpt: Layers in large language models are more redundant than you expect. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20192–20204. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.1035. URL http://dx.doi.org/10.18653/v1/2025.findings-acl.1035.

William Merrill and Ashish Sabharwal. The parallelism tradeoff: Limitations of log-precision transformers. *Transactions of the Association for Computational Linguistics*, 11:531–545, 2023a. ISSN 2307-387X. doi: 10.1162/tacl_a_00562. URL http://dx.doi.org/10.1162/tacl_a_00562.

William Merrill and Ashish Sabharwal. The expressive power of transformers with chain of thought, 2023b. URL https://arxiv.org/abs/2310.07923.

Marius Mosbach, Tiago Pimentel, Shauli Ravfogel, Dietrich Klakow, and Yanai Elazar. Few-shot fine-tuning vs. in-context learning: A fair comparison and evaluation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12284–12314. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-acl.779. URL http://dx.doi.org/10.18653/v1/2023.findings-acl.779.

ALLAN H. MURPHY. The ranked probability score and the probability score: A comparison^{1,,}². *Monthly Weather Review*, 98(12):917–924, December 1970. ISSN 1520-0493. doi: 10.1175/1520-0493(1970)098<0917:trpsat>2.3.co;2. URL http://dx.doi.org/10.1175/1520-0493(1970)098<0917:trpsat>2.3.co;2.

Rafael Müller, Simon Kornblith, and Geoffrey Hinton. When does label smoothing help?, 2019. URL https://arxiv.org/abs/1906.02629.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial nli: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901. Association for Computational Linguistics, 2020a. doi: 10.18653/v1/2020.acl-main.441. URL http://dx.doi.org/10.18653/v1/2020.acl-main.441.

Yixin Nie, Xiang Zhou, and Mohit Bansal. What can we learn from collective human opinions on natural language inference data? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9131–9143. Association for Computational Linguistics, 2020b. doi: 10.18653/v1/2020.emnlp-main.734. URL http://dx.doi.org/10.18653/v1/2020.emnlp-main.734.

Zhenxing Niu, Mo Zhou, Le Wang, Xinbo Gao, and Gang Hua. Ordinal regression with multiple output cnn for age estimation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4920–4928. IEEE, 2016. doi: 10.1109/cvpr.2016.532. URL http://dx.doi.org/10.1109/cvpr.2016.532.

Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with bert, 2019. URL https://arxiv.org/abs/1901.04085.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1), February 2015. ISSN 2159-5399. doi: 10.1609/aaai.v29i1.9602. URL http://dx.doi.org/10.1609/aaai.v29i1.9602.

Koyena Pal, Jiuding Sun, Andrew Yuan, Byron Wallace, and David Bau. Future lens: Anticipating subsequent tokens from a single hidden state. In *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pages 548–560. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.conll-1.37. URL http://dx.doi.org/10.18653/v1/2023.conll-1.37.

Nikolaos Pappas and James Henderson. Gile: A generalized input-label embedding for text classification. *Transactions of the Association for Computational Linguistics*, 7:139–155, November 2019. ISSN 2307-387X. doi: 10.1162/tacl_a_00259. URL http://dx.doi.org/10.1162/tacl_a_00259.

Pouya Pezeshkpour and Estevam Hruschka. Large language models sensitivity to the order of options in multiple-choice questions. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2006–2017. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-naacl.130. URL http://dx.doi.org/10.18653/v1/2024.findings-naacl.130.

Jacob Pfau, William Merrill, and Samuel R. Bowman. Let's think dot by dot: Hidden computation in transformer language models, 2024. URL https://arxiv.org/abs/2404.15758.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Anselm Levskaya, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference, 2022. URL https://arxiv.org/abs/2211.05102.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3980–3990. Association for Computational Linguistics, 2019. doi: 10.18653/v1/d19-1410. URL http://dx.doi.org/10.18653/v1/d19-1410.

Nithya Sambasivan, Shivani Kapania, Hannah Highfill, Diana Akrong, Praveen Paritosh, and Lora M Aroyo. "everyone wants to do the model work, not the data work": Data cascades in high-stakes ai. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, pages 1–15. ACM, May 2021. doi: 10.1145/3411764.3445518. URL http://dx.doi.org/10.1145/3411764.3445518.

Roy Schwartz, Gabriel Stanovsky, Swabha Swayamdipta, Jesse Dodge, and Noah A. Smith. The right tool for the job: Matching model and instance complexities. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6640–6651. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.593. URL http://dx.doi.org/10.18653/v1/2020.acl-main.593.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context, 2023a. URL https://arxiv.org/abs/2302.00093.

Xintong Shi, Wenzhi Cao, and Sebastian Raschka. Deep neural networks for rank-consistent ordinal regression based on conditional probabilities. *Pattern Analysis and Applications*, 26(3):941–955, 2023b. ISSN 1433-755X. doi: 10.1007/s10044-023-01181-9. URL http://dx.doi.org/10.1007/s10044-023-01181-9.

Charlie Snell, Dan Klein, and Ruiqi Zhong. Learning by distilling context, 2022. URL https://arxiv.org/abs/2209.15189.

Zayne Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning, 2024. URL https://arxiv.org/abs/2409.12183.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826. IEEE, 2016. doi: 10.1109/cvpr.2016.308. URL http://dx.doi.org/10.1109/cvpr.2016.308.

Zhi Rui Tam, Cheng-Kuang Wu, Yi-Lin Tsai, Chieh-Yen Lin, Hung-yi Lee, and Yun-Nung Chen. Let me speak freely? a study on the impact of format restrictions on large language model performance. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1218–1236. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-industry.91. URL http://dx.doi.org/10.18653/v1/2024.emnlp-industry.91.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601. Association for Computational Linguistics, 2019. doi: 10.18653/v1/p19-1452. URL http://dx.doi.org/10.18653/v1/p19-1452.

Xinpeng Wang, Bolei Ma, Chengzhi Hu, Leon Weber-Genzel, Paul Röttger, Frauke Kreuter, Dirk Hovy, and Barbara Plank. "my answer is c": First-token probabilities do not match text answers in instruction-tuned language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 7407–7416. Association for Computational Linguistics, 2024a. doi: 10.18653/v1/2024.findings-acl.441. URL http://dx.doi.org/10.18653/v1/2024.findings-acl.441.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems 37*, NeurIPS 2024, pages 95266–95290. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2024b. doi: 10.52202/079017-3018. URL http://dx.doi.org/10.52202/079017-3018.

Ziming Wang, Zeyu Shi, Haoyi Zhou, Shiqi Gao, Qingyun Sun, and Jianxin Li. Towards objective fine-tuning: How llms' prior knowledge causes potential poor calibration? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14830–14853. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.722. URL http://dx.doi.org/10.18653/v1/2025.acl-long.722.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Griffin Thomas Adams,

Jeremy Howard, and Iacopo Poli. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2526–2547. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.127. URL http://dx.doi.org/10.18653/v1/2025.acl-long.127.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35*, NeurIPS 2022, pages 24824–24837. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2022. doi: 10.52202/068431-1800. URL http://dx.doi.org/10.52202/068431-1800.

Brandon T. Willard and Rémi Louf. Efficient guided generation for large language models, 2023. URL https://arxiv.org/abs/2307.09702.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics, 2018. doi: 10.18653/v1/n18-1101. URL http://dx.doi.org/10.18653/v1/n18-1101.

Hugh Mee Wong, Rick Nouwen, and Albert Gatt. When models decide and when they bind: A two-stage computation for multiple-choice question answering. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 22804–22821. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.1144. URL http://dx.doi.org/10.18653/v1/2026.findings-acl.1144.

Wilson Wu, John X. Morris, and Lionel Levine. Do language models plan ahead for future tokens?, 2024. URL https://arxiv.org/abs/2404.00859.

Mengge Xue, Zhenyu Hu, Liqun Liu, Kuo Liao, Shuang Li, Honglin Han, Meng Zhao, and Chengguo Yin. Strengthened symbol binding makes large language models reliable multiple-choice selectors. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4331–4344. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.237. URL http://dx.doi.org/10.18653/v1/2024.acl-long.237.

Zhen Yang, Ping Jian, and Chengzhi Li. Option symbol matters: Investigating and mitigating multiple-choice option symbol bias of large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1902–1917. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.naacl-long.95. URL http://dx.doi.org/10.18653/v1/2025.naacl-long.95.

Wenpeng Yin, Jamaal Hay, and Dan Roth. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3912–3921. Association for Computational Linguistics, 2019. doi: 10.18653/v1/d19-1404. URL http://dx.doi.org/10.18653/v1/d19-1404.

Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisiting knowledge distillation via label smoothing regularization. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3902–3910. IEEE, 2020. doi: 10.1109/cvpr42600.2020.00396. URL http://dx.doi.org/10.1109/cvpr42600.2020.00396.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. Gliner: Generalist model for named entity recognition using bidirectional transformer. 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 5364–5376. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.naacl-long.300. URL http://dx.doi.org/10.18653/v1/2024.naacl-long.300.

Urchade Zaratiana, Gil Pasternak, Oliver Boyd, George Hurn-Maloney, and Ash Lewis. Gliner2: An efficient multi-task information extraction system with schema-driven interface, 2025. URL https://arxiv.org/abs/2507.18546.

Zhilu Zhang and Mert R. Sabuncu. Self-distillation as instance-specific label smoothing. 2020. doi: 10.48550/ARXIV.2006.05065. URL https://arxiv.org/abs/2006.05065.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. Large language models are not robust multiple choice selectors, 2023. URL https://arxiv.org/abs/2309.03882.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph Gonzalez, Clark Barrett, and Ying Sheng. Sglang: Efficient execution of structured language model programs. In *Advances in Neural Information Processing Systems 37*, NeurIPS 2024, pages 62557–62583. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2024. doi: 10.52202/079017-2000. URL http://dx.doi.org/10.52202/079017-2000.

A Appendix

A.1 Additional tables and figures

Table 4 lists the held-out axes of Section 5.2. Tables 5 and 6 give the per-variant numbers for the candidate-blind states of Section 6.2, and Table 7 the rendered-null-line ablation of Section 6.4. Figure 3 plots Table 2 against backbone size, and Table 8 gives the rendering-only comparison of Section 7.7. Figure 4 gives the long-state truncation histogram behind Section 4.5. The serving-path engineering behind Section 7.5, with Figure 6, is in Appendix A.3.

Table 4: Generalization axes, each with its own held-out split, and the scope a positive result licenses.

Axis	Evaluated on	A positive result licenses
Unseen examples	Held-out rows of every trained corpus	In-distribution competence
Unseen label wording	Label paraphrases; re-worded CLINC-150 labels	Robustness to label surface form
Unseen label space	Banking77-77, HWU64, tree-choice ($K \leq 320$), TruthfulQA-MC1	Scoring labels never trained on, within trained domains
Unseen family	3 of 15 workflow families; 3 DecisionMix domains	Transfer across families of one generator
Unseen rubric style	2 withheld DecisionMix writing styles	Rubric conditioning beyond style memorization
Unseen grammar	1 withheld domain×level grammar	Rule execution inside the generator's grammar
Unseen composition	DecisionMix level 7 (temporal, numeric, expected value, trade-off)	Composition outside the trained grammar
External	PagerDuty, jevlogs, Mind2Web, typed-decisions, JevBench	Transfer outside our generators

Table 5: Candidate-blind variants: six factorizations and objectives, two tap depths. "Shuffled" replaces the real question with an unrelated one, candidate set fixed; Δ_q^{sh} is the difference. n=1,200 MMLU-Pro items. Referenced from Section 6.2.

Variant	Factorization / objective	among-K	shuffled	Δ_q^{sh}
`z1`	single decision vector, bilinear scorer	.171	.147	$+.023$
`zr`	8 probes, token-level max-similarity	.157	.182	$-.026$
`zr_set`	+ $O(K)$ cross-attention over candidate tokens	.147	.151	$-.003$
`ms_ctrl`	single-set distillation, steps-matched control	.124	.147	$-.023$
`ms_zr_set`	same-Z multi-set, Δlog-odds targets	.130	.130	$+.000$
`zr` (full depth)	8 probes, tap at layer 28 of 28	.147	.156	$-.008$
Teacher	listwise decoder, options in context	.310	.208	$+\mathbf{.102}$

We compare the multi-set variant (`ms_zr_set`, Δ_q^{sh} exactly zero) against its steps-matched single-set control (`ms_ctrl`, $-.023$). Reading the full 28 layers lowers among-K accuracy to .147 and regresses SNLI/MNLI/ANLI/BoolQ to .881/.807/.462/.712 from .905/.873/.541/.826 at layer 20 (Table 6). The candidate-blind variants are exactly set-stable (reorder Δp and pairwise Δlog-odds both zero).

Table 6: Candidate-blind Z, closure at full depth vs. the original tap-20 student.

	Tap 20	Full depth (28)	Teacher
MMLU-Pro among-K / choices-only / shuffled-q	.157 / .166 / .182	.147 / .177 / .156	.310 / .222 / .208
Δ_q (choices-only) / Δ_q^{sh} (primary)	$-.009$ / $-.026$	$-.030$ / $-.008$	.088 / .102
SNLI / MNLI / ANLI / BoolQ	.905/.873/.541/.826	.881/.807/.462/.712	–

A.2 Native choice vs. energy scoring vs. batched log-probability, full K-sweep

Throughput mode on an earlier full-depth prototype. These numbers come from a different architecture and harness than Section 7.5: the full-depth (28-layer) contextual-readout prototype of

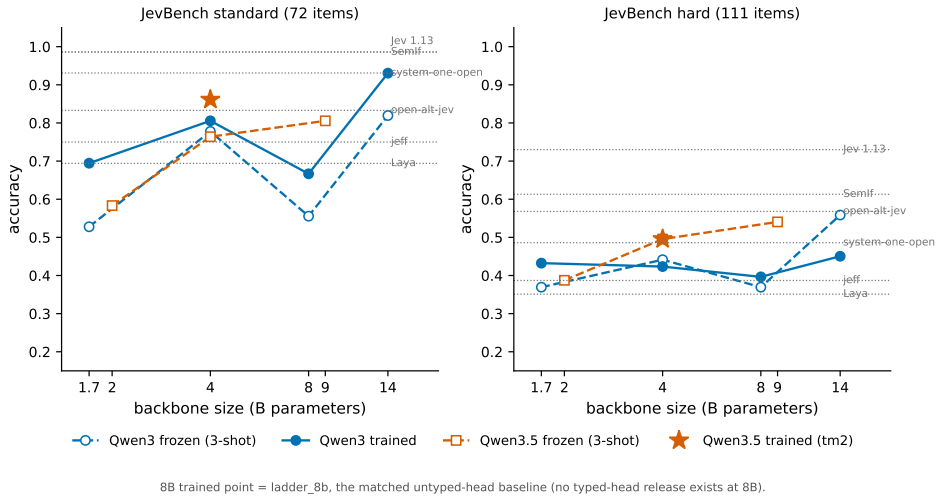

Figure 3: JevBench (public subset) standard and hard accuracy against backbone size. Qwen3 trained (solid; each point uses the recipe named for its row in Table 2) and frozen with three exemplars (dashed); Qwen3.5 frozen controls at 2B/4B/9B and the trained Qwen3.5-4B (star). Dotted lines mark other public entries on the same ids, whose rendering and protocols differ (Appendix A.13). Intervals are in Table 2.

Table 7: Rendered ∅ line vs. the factored null, matched checkpoints (tap 20).

	Rendered ∅ line	Factored null (no rendered line)
MMLU-Pro false-abstain rate	.692	.003
MMLU-Pro accuracy	.133	.353
Banking77 (unseen labels) accuracy / false-abstain	.063 / .924	.579 / .135
CLINC-OOS accuracy / null AUROC	.639 / .840	.798 / .973
Δ_q^{sh} (unchanged)	.118	.122

Section 6.1, timed before the one-pod comparison. At $M{=}256$ queries against one cached 256-token state, its per-query marginal cost is 1.07, 1.55, 3.38 and 43.8 ms at $K{=}2, 10, 32, 256$; a candidate-blind energy-scoring path costs 0.97–1.06 ms across the same range (the contextual readout is $1.1\times$, $1.6\times$, $3.4\times$ and $41\times$ its cost); and a baseline that re-runs the backbone once per candidate costs 2.1, 4.6, 12.2 and 101.7 ms. We transcribed these values from the run's log after an out-of-memory crash lost the run's `bench.json`. We measured timing only and did not evaluate a shortlist-then-decide pipeline for accuracy. On the one-pod harness, batched marginal cost at $M{=}32$ scales more steeply with both size and K than single-decision cost, from 2.7 ms at 1.7B/$K{=}2$ to 109.7 ms at 14B/$K{=}256$.

A.3 Serving-path engineering

The first version of the public inference package deep-copied the entire cached prefix's key-value tensors on every decision. We replaced the copy with a zero-copy strided view plus cached attention masks and position tensors, which cut warm per-decision latency from 21–27 ms to 15.5–17 ms (`typical-small` v1, Qwen3-1.7B) and 19–21 ms (`typical-medium` v1, Qwen3-4B) on one H100 (Figure 6). `tm2` (Qwen3.5-4B, `typical-medium` v2) serves at 34–46 ms warm p50 on the same package with reference DeltaNet kernels. `torch.compile`/CUDA-graph capture reached 8 ms at a single fixed input shape, but probabilities shifted by up to .1 across the shape buckets the serving path sees, and the mutable key-value cache produced stale-buffer crashes under the standard capture-and-replay pattern, so we reverted it. Manual per-bucket graph capture is the remaining path toward $\sim$10 ms decisions.

Table 8: Rendering alone, frozen weights. Identical Qwen3.5 base checkpoints and items, JevBench (public subset), letter-logit readout in both columns; only the state and option rendering differs. Δ: paired cluster bootstrap over the per-item difference, 20,000 resamples. Referenced from Section 7.7.

Tier	Backbone	our rendering	replicated rendering	paired Δ [95% CI], p
standard	Qwen3.5-2B	.583	.597	$+.014$ $[-.194, +.208]$, p=.95
	Qwen3.5-4B	.764	.847	$+.083$ $[+.014, +.167]$, p=.048
	Qwen3.5-9B	.806	**.931**	$+\mathbf{.125}$ $[+.056, +.208]$, p<.001
hard	Qwen3.5-2B	.387	.441	$+.054$ $[-.045, +.153]$, p=.33
	Qwen3.5-4B	.495	.468	$-.027$ $[-.117, +.063]$, p=.64
	Qwen3.5-9B	.541	.595	$+.054$ $[-.027, +.135]$, p=.22

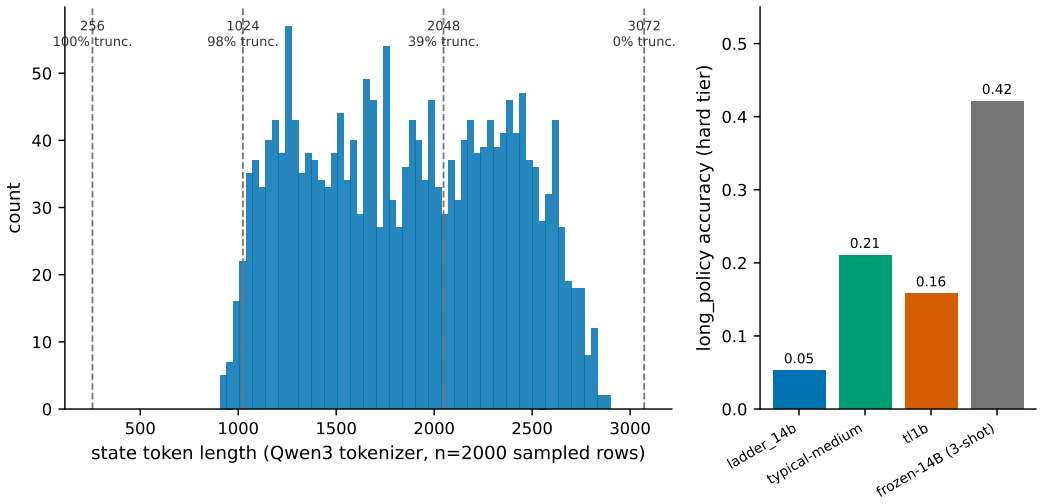

Figure 4: Left: histogram of `data_wf_long` state token lengths with truncation fractions at 256/1024/2048/3072 tokens. Right: long_policy (hard-tier) accuracy across the long-state bug/fix sequence.

A.4 Probability quality: global fit and NLL by checkpoint

Figure 7 gives held-out Score and typed-decisions NLL by checkpoint for Section 7.3.

Fitting one global temperature and null-logit offset jointly on held-out soft-target rows gives opposite corrections for two checkpoints (Section 7.3). For an evidence-and-knowledge checkpoint the fitted offset is -1.75, which cuts CLINC-150 false abstention from .053 to .022. For a workflow-trained checkpoint it is $+3.0$, which repairs typed-decisions NLL (1.95 $\rightarrow$ 1.19) and held-out Score NLL (2.03 $\rightarrow$ 1.09) and collapses intent and topic coverage to 5–9%. We make the correction in data and objective instead, through null augmentation and per-type targets (Sections 7.1 and 7.2).

A.5 Mixture sweep, full table

Table 9: Mixture sweep at 1.7B, matched steps, referenced from Section 7.2. "6A" is the E:0.35/K:0.25/W:0.40 point; "e45+null" adds `null_aug W:0.20` to E:0.45/K:0.20/W:0.35.

	Base (E only)	6A	e45	**e45+null**	long$_{e45}$ (1024 tok)
CLINC-150 acc / false-abstain	.845/.05	.734/.17	.779/.10	**.805/.08**	.741/.13
MMLU-Pro among-K / Δ_q^{sh}	.353/.122	.330/.119	.338/.115	**.353/.143**	.329/.116
Held-out score NLL	–	2.03	1.52	2.04	**1.26**
JevBench std / hard	.694/.378	.750/.387	**.833**/.396	.806/.396	.806/**.450**

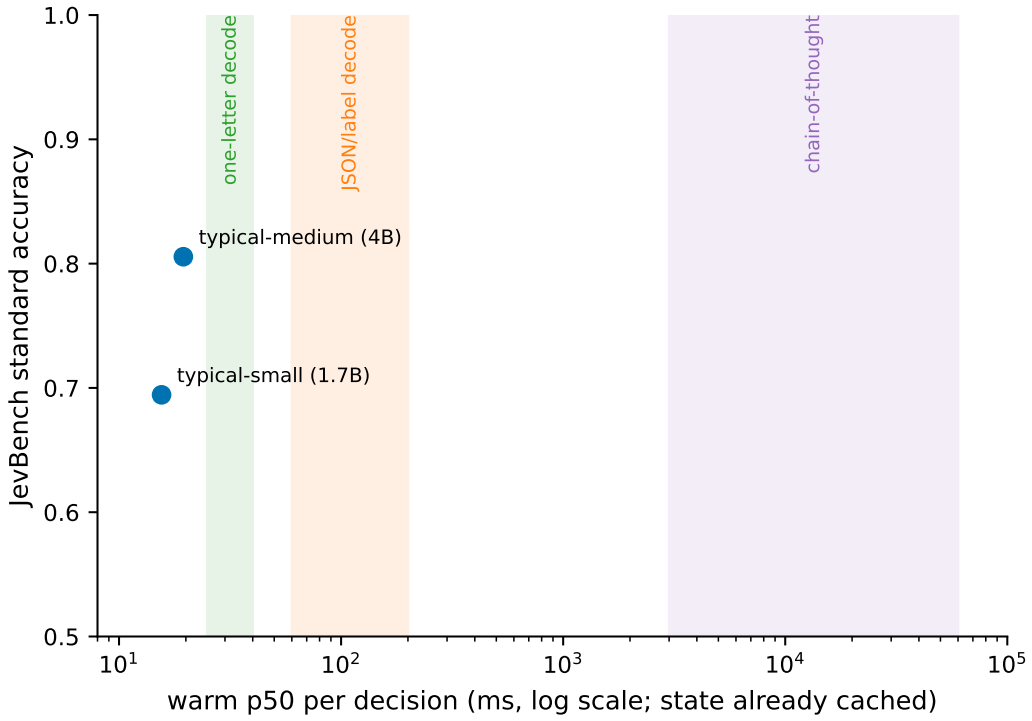

Figure 5: JevBench (public subset) standard accuracy vs. *warm* per-decision p50 latency (ms, log scale) for the v1 release checkpoints (`ts1b`, `tm1b`), with generative-LLM decode-latency comparison bands. Latency is the served path with the state prefix already cached (`runs/serve_bench2`, K=2, 256-token state, one stream, in process, model load excluded), the same quantity as Appendix A.3. The single-decision ladder of Table 25 measures a different quantity: it includes the state encode on the unoptimized path. Referenced from Section 7.5.

In this sweep, which ran without the uncertainty corpus or null augmentation, $E \geq .45$ brings NLI and BoolQ back within 1 point of the evidence-only base and intent and topic sets within 3–4, with workflow metrics within 1–4 (Section 7.2). Before the facts-first fix (Section 4.5), the long-state row (long$_{e45}$, 1,024 tokens, 25% long-policy rows) alone raised `long_policy` from .16 to .37 and `multi_hop` from .17 to .28 relative to 6A, and gave the best held-out Score NLL of the sweep (1.26).

A.6 Typed primitives: parameterization, routing, and full results

Choice is the K-way readout of Section 3.2, unchanged: a hard one-of-K target trained with soft cross-entropy against Equation (2). Score is ordinal (severity, quality, urgency). Plain Choice penalizes a one-level miss as much as an opposite-end miss, so Score uses an ordinal-smoothed target,

$$\tilde{t}_j \propto \exp\left(-\frac{|j-y|}{\tau}\right), \qquad \tau = 0.7, \tag{3}$$

normalized over valid levels, with τ=0 recovering one-hot Choice. Only the target changes; the architecture stays the same. Noul is a yes/no proposition. Rendered as 2-way Choice it has no order-invariance guarantee, since "yes" and "no" occupy two different suffix positions with two different contextual states. We therefore give Noul a dedicated Bernoulli head that reads the query alone, with no candidate text,

$$P(\text{yes}) = \sigma(w_n \cdot h_D), \tag{4}$$

which is exactly order-invariant by construction for rows routed to it; the reversed-label control confirms this with an exact zero. The remaining mass $1-P(\text{yes})$ goes to the other candidate. The log of these two probabilities takes the place of the candidate scores in Equation (2), and a separate gate reading h_D supplies $P(\varnothing)$, so the output keeps the factored abstention form.

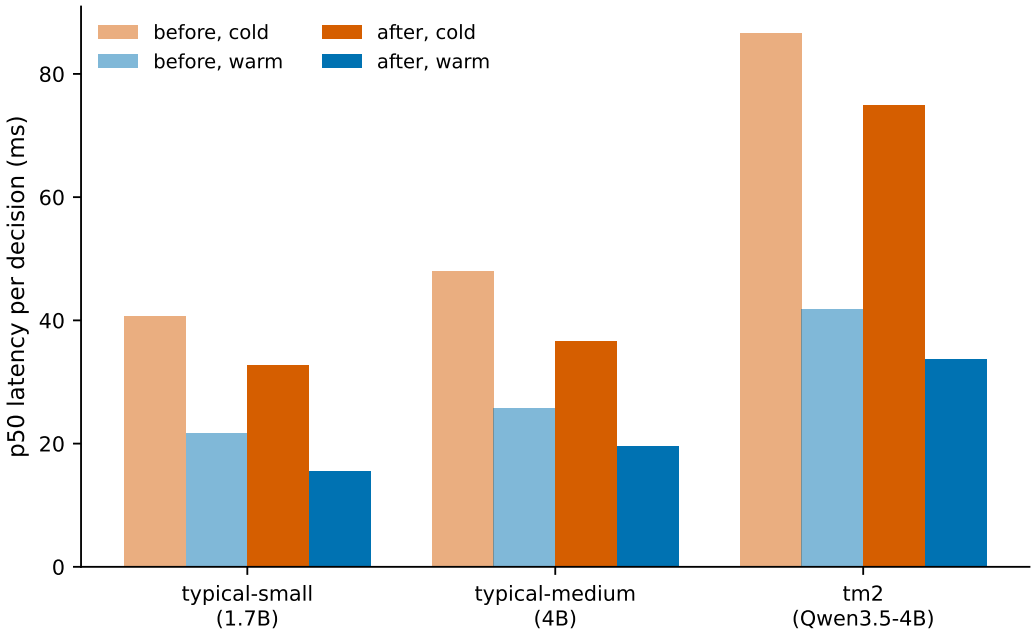

Figure 6: Per-decision p50 latency before/after the zero-copy prefix fix, cold vs. warm, for `typical-small` v1 (1.7B), `typical-medium` v1 (Qwen3-4B), and `tm2` (Qwen3.5-4B, `typical-medium` v2), at $K=2$, 256-token state.

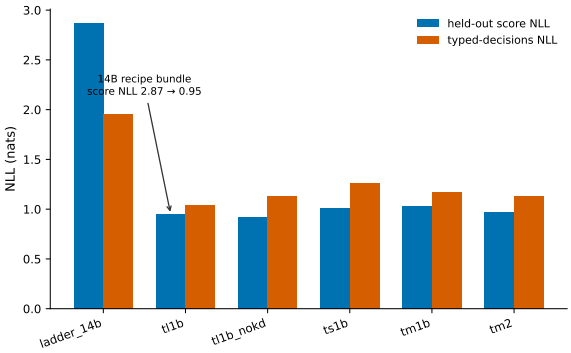

Figure 7: Held-out Score and typed-decisions NLL (held-out sets, not JevBench). `ladder_14b`: the 14B under the earlier workflow recipe; `tl1b`: the same backbone under the recipe bundle of Section 7.3; `tl1b_nokd`: the bundle without distillation. `ts1b`, `tm1b`, `tm2`: 1.7B and 4B checkpoints with typed heads and DecisionMix.

Per-row routing. A row is Noul-eligible iff its candidate set is exactly $\{\text{yes}, \text{no}\}$; every other row scores as ordinary Choice, bit-identical to a run with Noul disabled. The type is a contract on the input. An early release candidate broke that contract by applying the Bernoulli head at the *model* level, so it scored Choice- and Score-typed rows in the same batch with no candidate text, and its accuracy collapsed to chance. Routing per row, which 140 unit tests verify, restored it.

A.7 DecisionMix v2, full numbers

Held-out family/grammar/style accuracy moves from .48/.51/.49 (control, never trained on `data_wh` or `data_u`) to .83/.88/.89 (DecisionMix v2 together with the uncertainty corpus; the treatment also changed the sampler), and rubric-flip accuracy moves from .48 to .71, measured within the generator's own rule engine (Section 4.2). The uncertainty corpus moves typed-decisions NLL from 2.06

Table 10: Typed primitives vs. their K-way Choice control, matched fine-tunes, referenced from Section 7.1.

Score	K-way Choice	Ordinal-smoothed (τ=0.7)
Held-out urgency: acc / NLL / ECE	.505 / 2.07 / .35	.508 / **1.23** / **.19**
JevBench hard: acc / Brier	.360 / .88	**.459** / **.70**
JevBench ordinal items (12), decisions changed	–	0 of 12
Noul	**2-way Choice**	**Bernoulli**
PagerDuty (floor .792): acc / NLL	.602 / 1.04	**.886** / **.28**
Reversed-label max $\lvert\Delta P(\text{yes})\rvert$	.55	**0** (exact, all 9 sets)

to 1.71 and ChaosNLI NLL from 1.27 to 1.00 while ChaosNLI accuracy falls from .584 to .537 (Section 4.3). Level-7 composition is in Appendix A.10.

A.8 Full hyperparameters

Table 11 gives the exact configuration of each checkpoint in the paper. All runs use AdamW, bf16, and LoRA rank 16 on the top eight kept layers of the truncated backbone. Sections 4.1 and 4.4 describe the family-balanced sampler and null augmentation.

Table 11: Training configuration by checkpoint. Tap is given as layer / total layers of the base model. E:K:W:U is the family-sampler weight per batch. Max state is the training-time token budget for the cached prefix.

Checkpoint	Backbone	Tap	Steps	Batch	E:K:W:U	Max state
`typical-small-preview`	Qwen3-1.7B-Base	20/28	12,000	64	.35:.25:.40:0	256
`ts1b` (`typical-small` v1)	Qwen3-1.7B-Base	20/28	12,000	64	.40:.15:.35:.10	1,024
`tm1b` (`typical-medium` v1)	Qwen3-4B-Base	26/36	12,000	64	.40:.15:.35:.10	1,024
`ladder_8b`	Qwen3-8B-Base	26/36	12,000	64	.35:.25:.40:0	256
`ladder_14b`	Qwen3-14B-Base	28/40	12,000	64	.35:.25:.40:0	256
`tl1b` (14B)	Qwen3-14B-Base	28/40	8,000	64	DecisionMix v2 + U, facts-first	3,072
`tm2` (`typical-medium` v2)	Qwen3.5-4B-Base	≈71%	12,000	64	.40:.15:.35:.10	2,048

`ts1b`/`tm1b` also set `null_aug W:0.20`, `ordinal_smooth 0.7` (Score), `noul_head bern` (routed per row, Section 3.5), `null factored`, and `nc_render letters_nonull`. `tl1b` adds `-drop_truncated`, frozen-teacher KD ($\alpha \cdot \text{CE} + \beta \cdot \text{KL}$, T=2, against the same backbone's own frozen 3-shot letter distribution on 63k labelled rows), and checkpoint selection on held-out uncertainty-plus-curriculum validation NLL (`-best_on`) in place of in-distribution loss. For `ts1b` we fit the final temperature and null offset post hoc, T=1.124 and offset 0, and apply them at evaluation. `ts1c` (`typical-small` v2) is the 1.7B model trained on the facts-first long-state corpus with a 2,048-token window, `-drop_truncated` and the same `-best_on` selection for 8,000 steps; its long-state and JevBench numbers are in Table 20. `tm2` also uses `-best_on`; `ts1b` and `tm1b` have no likelihood-based selection.

A.9 Corpus construction detail

`data_v5` (633,213 train / 4,001 val rows) covers evidence tasks and classification with a wide span of label vocabularies at low rows-per-source, so the model cannot memorize a fixed small set of label sets and must read the rendered options.

`data_kb` (147,447 train / 3,001 val) is distilled from ARC-Easy/Challenge, OpenBookQA, CommonsenseQA, SciQ, QASC, LogiQA, AQuA-RAT, MedMCQA, and MMLU-auxiliary-train. We hold out TruthfulQA-MC1, and after a leakage audit against MMLU-Pro we excluded two items from all MMLU-Pro numbers.

`data_wf` (72,006 train / 3,000 val) is our own rubric-conditioned workflow generator, 15 families of Noul/Choice/Score decisions (policy checks, adequacy, support, fact verification, eligibility, routing, action selection, enumeration extraction, categorical judgment, tool selection, and four ordinal quality/severity/relevance/completeness scales, plus urgency). We hold three whole families (eligibility,

tool_select, urgency, one per type) out of training. Of the 72,006 rows, 26,513 belong to *rubric groups*: the same state and candidate set rendered under two or three different rubrics with different gold answers. We built these groups to measure and train rubric dependence over state/candidate pattern-matching.

`data_wf_hf` (86,600 train rows) adds three externally sourced sets in the same typed shape: `cua-s1-forms` (60,000 rows, procedural GUI-form option-scoring, MIT license, with a form-signature-disjoint 24,370-row test split and a 196-row real-usage demo split held out), `systemone-lite-general` (20,000 rows, rule-labeled typed decisions, MIT, with a 5,400-row hard split held out), and `jev-4b-distill-data` (6,600 rows, programmatic gold, Apache-2.0, with a 600-row adversarial split held out).

`data_wf_long` (23,318 train rows) is a longer-state variant of the workflow generator for multi-paragraph policy documents, rendered facts-first (Section 4.5).

`data_wh` (60,605 train / 3,000 val) is a second, independent rubric-conditioned generator (DecisionMix v2): a programmatic rule engine over 12 domains and six boolean conditions at seven levels of compositional depth. We train on levels 1–6 and keep level 7 (compositions requiring temporal, unit-conversion, expected-value, or trade-off reasoning that the generator does not otherwise produce) for evaluation only. Every trained row belongs to a counterfactual rubric group. We hold out one whole domain/level pair, three whole domains, and two rubric styles.

`data_u` (31,029 train / 2,000 val) is a soft-target uncertainty corpus: the validation split of UNLI, the ambiguous rows of AmbiEnt (uniform target over the listed labels), and four synthetic generators with exact closed-form posteriors (partial-evidence combination, a noisy-sensor Bayesian update, ordinal confusion via an inverted confusion matrix, and conflicting-sources log-odds pooling), each with a held-out parameter range; ChaosNLI-ambiguity (MNLI portion) is evaluation-only.

Table 12: `data_wf`: 15 families, 72,006 train rows, 3 held out (marked *). "In rubric groups" counts rows that are part of a same-state-same-candidates, different-rubric-different-gold group.

Family	Type	Rows	In rubric groups
policy_permit	Noul	9,000	5,205
adequacy	Noul	5,000	0
support	Noul	6,000	0
fact	Noul	5,000	0
eligibility*	Noul	1,500	851
routing	Choice	9,003	5,208
action_select	Choice	7,000	4,054
enum_extract	Choice	7,000	3,618
categorical	Choice	7,004	3,882
tool_select*	Choice	1,500	905
quality	Score	6,000	0
severity	Score	6,000	3,497
relevance	Score	3,998	0
completeness	Score	4,001	2,196
urgency*	Score	1,502	900

Table 13: `data_wh` (DecisionMix v2): 60,605 train rows across 12 domains $\times$ 6 boolean conditions, levels 1–6; level 7 (temporal/numeric/probability/trade-off composition) is eval-only.

Level	Train rows	Held-out grammar rows
1	10,601	142
2	10,600	141
3	10,600	141
4	10,601	140
5	10,602	142
6	10,601	141

Eval file	Rows
`wh_heldout_family`	422
`wh_heldout_style`	403
`wh_heldout_grammar`	816
`wh_level7`	880
`wh_rubric_flip`	2,206
`wh_rubric_shuffled`	2,437

Table 14: `data_u`: 31,029 train rows, 100% soft-target.

Source	Train rows
Synthetic (4 closed-form generators)	28,560
UNLI (validation split only)	2,068
AmbiEnt (ambiguous rows)	401
By decision type: Noul / Score / Choice	23,403 / 7,225 / 401

A.10 Hard compositional tier and the frozen 14B comparison

Single-pass readout reaches its frontier on compositions deeper than anything in training. Rubric-conditioned training moves held-out families, grammars and styles inside the generator's distribution to .84–.90 (Appendix A.7). Level-7 composition, which requires temporal arithmetic, unit conversion, expected-value reasoning or multi-attribute trade-offs, moves far less. On the 880-item level-7 set the uniform-guess rate is .441 (598 items at K=2, 224 at 3, 58 at 4; the majority-class rate is .250), and the checkpoints score .493 (`ts1b`), .544 (`tm1b`) and .620 (`tl1b`). Only the 14B checkpoint sits well above the guess rate, and all three sit 25–40 points below their own in-distribution held-out scores: the curriculum that moves every in-distribution family by more than 30 points moves level 7 by 2 (.477 to .498). The JevBench hard tier's temporal, probability and trade-off families show the same pattern.

Table 15: 14B comparisons. JevBench (public subset); 95% cluster-bootstrap intervals over the paraphrase `group` field (n_{eff}=36 standard states, n=111 hard items). `tl1b_nokd` is identical to `tl1b` in every flag except `-distill_beta 0`. Paired per-item tests are in the text; the marginal intervals overlap, and no comparison rests on them.

	`ladder_14b`	`tl1b`	`tl1b_nokd`	Frozen, 3-shot
JevBench std	.875 [.792, .944]	.931 [.861, .986]	.917 [.833, .986]	.819 [.708, .917]
JevBench hard	.468 [.378, .559]	.450 [.360, .541]	.477 [.387, .568]	**.559** [.468, .649]
JevBench Brier std (hard)	.173 (.85)	.175 (.66)	.127 (–)	.30 (.60)
`long_policy` (hard subfamily)	.053	.158	.211	**.421**
Training val NLL	–	.438	.410	–
Held-out score NLL	2.87	**0.95**	–	–
Level-7 composition (guess rate .441)	–	.620	–	–

The frozen 14B backbone with three exemplars leads the trained 14B on the hard tier. Table 15 compares the previous-recipe 14B checkpoint (`ladder_14b`); the bundle that fixed the long-state rendering, dropped over-length rows, and added DecisionMix v2, the typed heads and likelihood-based checkpoint selection (`tl1b`); the same bundle without frozen-teacher distillation (`tl1b_nokd`); and the frozen backbone with three exemplars. A paired cluster bootstrap on the per-item difference over the same 111 hard items gives frozen-minus-trained $+.108$ $[+.027, +.189]$, p=.015 against `tl1b`, and $+.081$ $[-.009, +.171]$, p=.082 against `tl1b_nokd`. On the standard tier the ordering reverses (frozen-minus-trained $-.111$ $[-.250, +.014]$, p=.10). We scored the frozen control with a 3-shot letter-logit rendering that we did not sweep; rendering alone moves a frozen 9B checkpoint by 12.5 standard-tier points (Table 8).

The backbone carries hard-tier behavior that the current recipe does not preserve. Per family (Figure 8, Table 17), the frozen 14B leads `tl1b` on `tradeoff` (.500 vs. .167), `long_policy` (.421 vs. .158), `ambiguous` (.714 vs. .429), `probability` (.500 vs. .300) and `temporal_numeric` (.333 vs. .200), and trails on `multi_hop` (.444 vs. .611). Families hold 3–19 items and no single family separates; the direction is the same across every serial family. Our generators compose boolean rules up to six levels deep and never produce the temporal, probability and trade-off operations these families use, which explains why training did not add the behavior. The measurement that separates missing coverage from a limit of fixed-depth computation is the same frozen backbone given a scratchpad on the same 111 items; Section 8 takes this up.

Same-backbone frozen-teacher distillation has no detectable effect. It is the one component of the 14B bundle with its own matched control. `tl1b_nokd` has the better point estimate on hard accuracy (.477 vs. .450), `long_policy` (.211 vs. .158), training validation NLL (.410 vs. .438) and

standard-tier Brier (.127 vs. .175); `tl1b` has the better one on standard accuracy (.931 vs. .917). The paired bootstraps are $-.027$ $[-.090, +.027]$ (p=.45) on hard and $+.014$ $[-.042, +.069]$ (p=.81) on standard.

The bundle's clear effect is on probability quality. The `ladder_14b` → `tl1b` standard-tier change is .875 → .931 (paired $+.056$ $[+.000, +.125]$, p=.13). Held-out score NLL falls from 2.87 to 0.95 and hard-tier Brier from .85 to .66, both large relative to their noise.

The same question on 19 JevBench items. `ladder_14b` scores .053 on the JevBench `long_policy` family (n=19) and .919 on the 605 held-out long states in its own training render (Table 3). Before the 2×2 of Section 7.6, we asked the facts-first vs. facts-last question on JevBench (Table 16). The two 1.7B arms are identical in every flag except whether `data_wf_long` renders its facts at 0% or 97.6% of the state text (same rows, same labels, matched state length, $p50 = 8{,}702$ characters in both arms), trained at `-max_state 1024` with `-drop_truncated` off so that truncation bites exactly as it did before the fix. On 19 `long_policy` items in one fixed benchmark render the seed-0 arms read .316 (6/19) vs. .105 (2/19), Fisher exact p=.232, with the hard-tier aggregate moving the other way. A second seed of the whole pair, byte-identical flags apart from the seed, reads 5/19 vs. 5/19: the 19-item family gives a four-item gap at one seed and none at the other. On the 605 held-out long states the same pairs give matched-configuration gaps of $+.112$ $[+.077, +.147]$, $p = 5 \times 10^{-10}$ (seed 0) and $+.111$ (seed 1, .937 vs. .826), and the 14B pair gives $+.078$ $[+.055, +.100]$, $p = 7 \times 10^{-12}$ (Table 3).

Table 16: Matched facts-first vs. facts-last training arms, 1.7B, single variable, two seeds, scored on JevBench: the question of Table 3 asked on 19 `long_policy` items in a single fixed render. The last row gives the same pairs on the 605 held-out long states. JevBench (public subset), 95% cluster-bootstrap intervals.

	Arm A (facts-first)	Arm B (facts-last)	Test
seed 0			
`long_policy`	.316 (6/19)	.105 (2/19)	Fisher p=.232; bootstrap Δ $[-.053, +.474]$
JevBench hard	.378 [.288, .468]	.396 [.306, .486]	opposite direction
JevBench std	.750 [.625, .861]	.736 [.597, .861]	indistinguishable
seed 1			
`long_policy`	.263 (5/19)	.263 (5/19)	tie
JevBench hard	.405	.387	–
JevBench std	.792	.708	–
605 states, matched	.942 / .937	.830 / .826	gap $+.112$ (s0), $+.111$ (s1)

A.10.1 Per-family JevBench-hard table

Table 17 gives the hard-tier sub-families at 14B. We read the frozen-control column and the `ambiguous` row from each run's JSON evaluation artifact. For `ladder_14b`'s temporal and probability cells, the JSON artifact and the project's internal report disagree (.333/.50 in the JSON vs. .20/.40 in the report); the table gives the report values, marked †.

A.11 Closed-loop decisions: a case study

We drove three games with `typical-small` through the public decision API, one decision per tick, on a laptop (Apple M4 Pro, MPS; the probes ran against the v1 weights, before the v2 upload). Each tick the game engine renders the current situation as the state, offers the legal actions as the candidate set, and executes the argmax. The state holds only the situations that apply at that tick, each written as a sentence that already contains the comparison a rule needs. Drive and Doom put a precedence-ordered list of one-condition rules in the question; Snake encodes its rules in the state sentences and asks a plain question. Accuracy is agreement with the literal rule list applied over the offered actions (Table 18).

Drive. The rule list is `stop`, `brake`, `change lane right`, `accelerate`, `change lane left`, `follow`, and a `hold speed` default, with lane changes removed from the candidate set when the lane is occupied or leads away from the exit. The model matched the rules on 70 scripted situations,

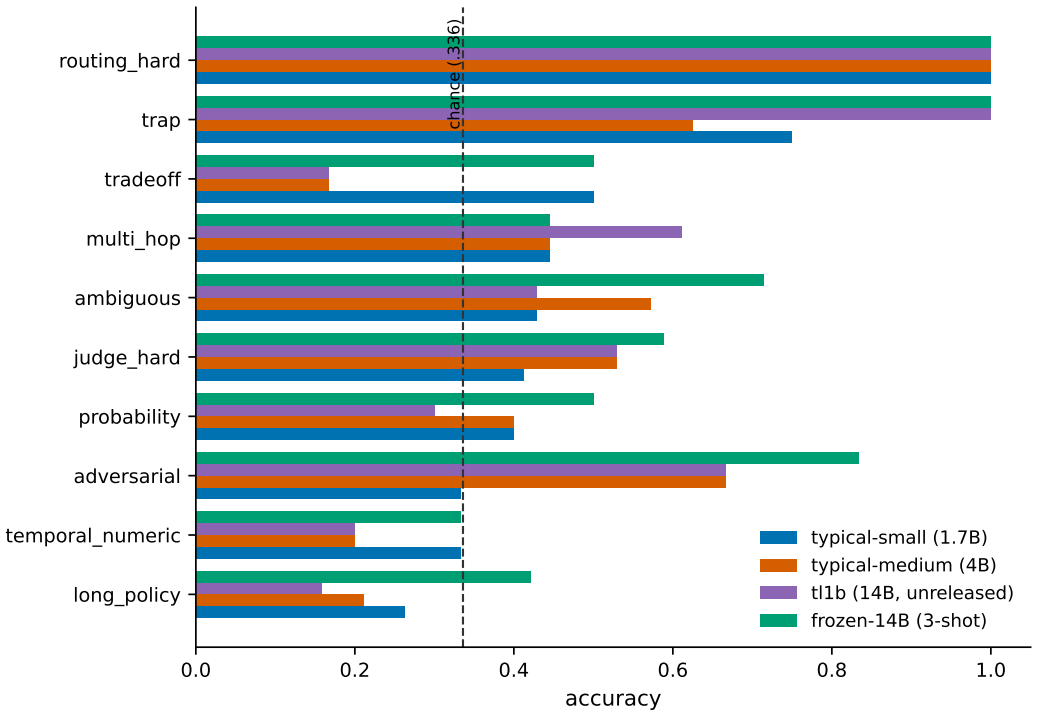

Figure 8: Per-family JevBench (public subset) hard-tier accuracy for the v1 release checkpoints (`ts1b`, `tm1b`), the 14B checkpoint `tl1b`, and the frozen-14B 3-shot control, chance line at .336. Per-family n ranges from 4 to 19 items; no single family separates, and the direction across the serial families is uniform. Values come from each run's JSON evaluation artifact.

Table 17: JevBench (public subset) hard-tier sub-family accuracy, 14B; per-family cells are 3–19 items each and carry no useful interval.

Sub-family	`ladder_14b`	`tl1b`	Frozen 14B, 3-shot
long_policy	.05	.158	.421
multi_hop	.44	.611	.444
trap	1.00	1.00	1.00
adversarial	.67	.667	.833
ambiguous	.429	.429	.714
temporal	.20†	.20	.333
probability	.40†	.30	.50
tradeoff	.67	.17	.50

† The JSON-derived values for these two `ladder_14b` cells are .333 (temporal) and .50 (probability).

including 14/14 exit lane changes and 15/15 overtakes. Removing the guardrail and offering all seven actions drops agreement to .771.

Doom. The model picks one of five actions and the engine presses the key, with a closed-loop turn by the enemy's bearing. On 48 scripted situations it matched the rules 48/48, including 29/29 turns, and it matched 20/20 sentences in the real engine's vocabulary. In a recorded run on DOOM's E1M1, with the engine walking a scripted route when the model chooses `explore`, all 200 decisions match the rule list: 133 explore, 64 shoot and 3 turns, for 4 kills.

Snake. The state is sentences such as "The snake must move right to close the larger gap to the food." and "Moving left hits the snake's body.", the question is "Which move brings the snake closer to the food without dying?", and the candidates are the safe moves. It matched the literal greedy policy on 61 of 62 boards (.984). Moving the same rules into the question as four direction rules without a default scored .82–.95 with $P(\varnothing)$ .84–.94.

Table 18: Closed-loop decisions with `typical-small`. "Probe" is agreement with the literal rule list on scripted situations; "chance" is uniform over the offered legal actions; "no rules" keeps the state and replaces the rule list with a goal question.

Game	Candidates (legal actions)	Probe	Chance	No rules	Recorded run
Drive	7 manoeuvres, lane changes only when legal	70/70	.16	.157	231/231 ticks match the rules; 18/18 red lights, 3 lane changes, exit taken
Doom (aiming)	retreat, shoot, turn left, turn right, explore; no shoot at 0 ammo	48/48	.20	.729	arena 240/240, 8 kills; E1M1 200/200 (133 explore, 64 shoot, 3 turn), 4 kills
Snake	safe moves	.984	.48	–	216/217 ticks, 24 food

What the rendering controls. The same weights and the same rules succeed or fail depending on how we write the state, which makes these games a closed-loop instance of Section 7.7. Four regularities held across games.

- *Agent-subject sentences that pre-compute the comparison fire their rule.* "The player must turn left to face the enemy." fires `turn left` 29/29; the object-location form "The nearest enemy is 5 cells to the left." fired 0/12 over five phrasings. In Drive, "The car must take the exit on the right in 200 m." fires 11/11, and "The exit on the right is 200 m ahead." 0/11. In Snake, the pre-computed sentences score .984 against .774 for the coordinate form "The food is 3 cells to the right and 2 cells up." on the same 62 boards.
- *An explicit default rule beats "none of the above".* "explore: applies when the player sees no enemy" fires 6/6; "explore: applies when none of the above rules fire" fires 0/8, with the mass going to `shoot` and `retreat`.
- *Raw numbers the model would have to compare leak into the decision.* With "The player has 41 health and 34 ammo." in the state, an imp in the crosshair reads `retreat` .56 / `shoot` .42; without the sentence `shoot` is .89. Mid-health crosshair situations score .500 with the numbers sentence and 1.000 without it, and 66 of 200 decisions in an earlier recorded run failed this way. Health and ammo now live on the game display, outside the state.
- *One condition per rule, within the trained list length.* A rule containing "or" fires where it should not (Drive .30 and .61 with a disjunctive rule vs. 1.00 without). A rule at position 6 fired 0/5 under four phrasings and the same rule at position 4 fired 4–5/5, matching the training shape of three rules plus a default.

A.12 Development chronology

Sections 4, 6 and 7 are organized by claim. The experiments ran in this order: (1) a candidate-blind decision-state branch (dual-encoder-style $Z(x, q)$ with an energy scorer over cached candidate embeddings), which produced the ladder of Table 5 and was closed by the shuffled-question control; (2) a candidate-in-suffix decision pass, comparing letter-logit, candidate-blind semantic, and contextual-candidate readouts, which produced Table 1; (3) the last-layer tap and rendered abstain-line diagnoses, which merged a planned two-regime (energy + native) architecture into a single model; (4) the workflow and uncertainty corpora, the mixture sweep, and null augmentation; (5) the Qwen3 size ladder at 1.7B/4B/8B/14B with frozen controls; (6) the 14B bundle (facts-first long-state rendering, `-drop_truncated`, DecisionMix v2, typed heads, likelihood-based checkpoint selection, frozen-teacher KD) and its matched no-KD control; (7) the Qwen3.5 port; (8) cluster bootstraps, the matched truncation ablation, and the purpose-built held-out long-state set, which produced the paired tests of Appendix A.10 and the 2×2 of Section 7.6; and (9) the v2 releases, `ts1c` (`typical-small`) and `tm2` (`typical-medium`), both trained on the facts-first corpus.

We closed the candidate-blind branch on Δ_q. Each student closed 20–55% of the chance-to-teacher accuracy gap, and the shuffled-question control showed that none of the closed gap was question-dependent; selection on accuracy would have kept the branch.

A.13 JevBench leaderboard context

On JevBench we drop $P(\varnothing)$ and renormalize the remaining probabilities; option order follows the harness (reversing it moves one checkpoint's standard accuracy by ± 4 points); majority baselines are .311 standard, .284 easy and .336 hard. Table 19 places our numbers among other entries scored on the same 231 public ids. Entries differ in protocol and rendering, several have no method paper, and the 146 private judge items are not included. The public standard tier has n_{eff}=36 independent states, and rendering alone moves a frozen model by 12.5 points (Section 7.7), which exceeds most of the spread in this table (Section 5.5).

Table 19: Other entries scored on the same public 231 ids, from the harness's own per-item files. JevBench (public subset) for all rows including ours. Majority/chance baselines: .311 standard, .284 easy, .336 hard. 95% cluster-bootstrap intervals for our own rows are in Tables 2 and 15; differences smaller than $\pm$6–13 points (standard) or $\pm$9 points (hard) are not rankings.

Entry (backbone)	standard (72)	easy (48)	hard (111)
open-jev-deberta-v3-large (classifier)	.431	1.00	.378
GLiNER2	.639	1.00	.369
jeff (GLiFormer 400M)	.750	1.00	.387
Laya (ModernBERT)	.694	1.00	.351
open-alternative-jev (Qwen3.5-4B)	.833	1.00	.568
system-one-open (Gemma E2B LoRA)	.931	1.00	.486
SemIf (Qwen3.5-4B)	.986	1.00	.613
OpenJev (26B-A4B)	.972	1.00	.640
Jev 1.13.0 (closed)	.986	1.00	.730
Typical, Qwen3-14B (`tl1b`)	.931	1.00	.450
Typical, Qwen3.5-4B (`tm2`, `typical-medium` v2)	.861	1.00	.495

A.14 Long-state truncation: distributions and the held-out evaluation set

Figure 4 gives the `data_wf_long` state-length histogram with truncation fractions at 256/1024/2048/3072 tokens, which is the measurement behind the 98.8% figure of Section 4.5, together with `long_policy` accuracy across the bug/fix sequence. The tokenizer's `truncation_side` is `'right'`; we verified this empirically on the released tokenizer.

We built the held-out long-state evaluation set (`scripts/make_long_eval.py`, n=605) for the 2×2 of Table 3. It has zero exact-state overlap with either training corpus, and we dropped `Case` stem overlaps. We score it at a 4,096-token window against states with $p50 = 1{,}965$ and $\max = 2{,}843$ tokens, so no item is truncated at evaluation in any of the eight cells. The two renderings differ only in whether the evidence block precedes or follows the request text; the items, labels, and candidate sets are identical. The set shares a generator family with the training data, and it measures the training defect of Section 7.6. Significance for the matched-configuration gap is a two-proportion test over the 605 paired items ($z = 6.2$, $p = 5 \times 10^{-10}$ at 1.7B; $p = 7 \times 10^{-12}$ at 14B).

Released checkpoints. We trained the v1 releases before the facts-first fix; the v2 releases replace them with post-fix checkpoints (Table 20). On facts-first long states `ts1c` gains 34.9 points over `ts1b` and `tm2` gains 33.8 over `tm1b`, while facts-last accuracy moves by -0.8 and $+1.3$. JevBench does not register the change: `ts1c` reads .708/.432 against `ts1b`'s .694/.432.

Table 20: Release checkpoints on the 605 held-out long states (4,096-token window, no truncation), identical items in both render columns. Majority-class floors: .334 overall, .612 on `policy_permit`. JevBench (public subset).

Release	Checkpoint	facts-first	facts-last	`policy_permit` (first)	JevBench std / hard
`typical-small` v1	`ts1b`	.598	.798	.536	.694 / .432
`typical-small` v2	`ts1c`	**.947**	.790	.948	.708 / .432
`typical-medium` v1	`tm1b`	.612	.866	.555	.806 / .423
`typical-medium` v2	`tm2`	**.950**	.879	.967	.861 / .495

A.15 Operations and reproducibility notes

Training and evaluation ran on rented RunPod H100 instances (SXM and NVL variants). For reproducibility, we record the infrastructure issues that affected timing. A pod created with an SSH-only startup flag never reached a ready state on the account used and billed regardless; we hit this several times before diagnosing it, and a fixed exposed port at boot avoided it. Stopping a pod without an attached persistent volume wipes it, so we wrote checkpoints and manifests to a network volume or pushed them to the Hugging Face Hub before stopping any pod. `tmux kill-session` does not terminate a `timeout`-wrapped child process, which once produced a stray duplicate training run sharing a GPU with the intended one; process matching over SSH must target the exact command, since a loose `pkill` pattern can match and kill the SSH shell itself before the target process. Two pods sharing one Python environment on the same network volume corrupted it under concurrent dependency resolution; per-pod environments with dependency syncing disabled avoided this. Every run's manifest records the harness commit, checkpoint hash, and dependency lock file used. We compute every JevBench per-item comparison in this paper against the same fixed set of 231 public ids from one harness commit.

A.16 Full result tables

Tables 21 to 25 give the complete cross-checkpoint result tables for every checkpoint with a completed evaluation. `ts1b` and `tm1b` are the v1 releases; `tm2` is `typical-medium` v2, and `ts1c` (`typical-small` v2) is in Table 20.

Table 21: Evidence and knowledge-classification accuracy by checkpoint.

Model	Checkpoint	CLINC-150	TREC-fine	HWU64	20NG	SNLI	MNLI	BoolQ	ANLI
Qwen3-1.7B	`ts1b`	.801	.508	.761	.540	.894	.859	.824	.497
Qwen3-4B	`tm1b`	.847	.414	.769	.588	.909	.861	.843	.544
Qwen3-8B	`ladder_8b`	.769	.486	.740	.452	.853	.831	.822	.468
Qwen3-14B	`tl1b`	.827	.508	.760	.690	.911	.868	.882	.583
Qwen3.5-4B	`tm2`	.795	.480	.717	.579	.900	.855	.840	.554

Table 22: Knowledge probe (MMLU-Pro among-K, Δ_q^{sh}) and TruthfulQA MC1.

Model	Checkpoint	MMLU-Pro among-K	Δ_q^{sh}	TruthfulQA MC1
Qwen3-1.7B	`ts1b`	.343	.127	.257
Qwen3-4B	`tm1b`	.458	.193	.408
Qwen3-8B	`ladder_8b`	.302	.093	.277
Qwen3-14B	`tl1b`	.498	.235	.481
Qwen3.5-4B	`tm2`	.429	.194	.414

Table 23: Held-out workflow and external floors. Rows marked *full* are the post-hoc `eval_wf` pass over every row at full state length (4,096-token window, `-limit 0`); rows marked *train-time* are the 1,500-row, 256-token pass emitted during training, which truncates the long external sets and is not comparable to the full pass. The distinction matters on the external sets: on `ts1b`, PagerDuty reads .560 at train time and .817 at full length, because the 256-token window removes the incident text. Numbers in parentheses after held-out score are NLL.

Model	Pass	Held-out noul	Held-out score (NLL)	Held-out style	PagerDuty (.792)	jevlogs (.697)	Mind2Web (.427)
`ts1b`	full	.715	.520 (1.01)	.859	**.817**	**.710**	.357
`tm1b`	full	.811	.528 (1.03)	.859	**.838**	.673	.544
`tl1b`	full	.831	.623 (0.95)	.880	**.878**	.546	.563
`ladder_8b`	train-time	.681	.543 (1.72)	.882	.321	.501	.404
`tm2`	train-time	.858	.586 (0.97)	.842	.757	.518	.497

Table 24: JevBench (public subset) accuracy (Brier): 72 standard over 36 paraphrase-clustered states / 48 easy / 111 hard; see Tables 2 and 15 for intervals.

Model (checkpoint)	Standard (Brier)	Easy (Brier)	Hard (Brier)
Qwen3-1.7B (`ts1b`)	.694 (.40)	1.00 (.005)	.432 (.79)
Qwen3-4B (`tm1b`)	.806 (.30)	1.00 (.012)	.423 (.77)
Qwen3-8B (`ladder_8b`)	.667 (.55)	1.00 (.015)	.396 (.83)
Qwen3-14B (`tl1b`)	**.931** (.18)	1.00 (.005)	.450 (.66)
Qwen3.5-4B (`tm2`)	.861 (.22)	1.00	**.495** (.72)
Qwen3.5-4B, frozen 3-shot	.764 (.36)	1.00	.495 (.60)
Qwen3.5-9B, frozen semif-render	.931 (.11)	1.00	.595 (.51)
Qwen3-14B, frozen 3-shot	.819 (.28)	1.00	.559 (.56)

Table 25: Single-decision research-harness latency (ms), L_s=256. The 1.7B, 4B and 14B rows are one pod and one PyTorch build; we timed the 8B row (marked $*$) on a separate pod.

Model (checkpoint)	K=2	K=32	K=256	Peak memory K=2→256 (GB)
Qwen3-1.7B	45	46	106	6.4 → 9.3
Qwen3-4B	56	56	118	14.6 → 18.6
Qwen3-8B*	70	71	126	29.3 → 34.1
Qwen3-14B	60	62	157	51.6 → 57.5

A.17 Training corpora

Table 26 lists the seven training corpora of Section 4; Appendix A.9 gives their construction.

Table 26: Training corpora. "Held out" counts whole families, domains, styles or parameter ranges excluded from training, distinct from a random validation split; Section 5.2 maps these onto generalization axes. No corpus contains any of the 231 public JevBench ids.

Corpus	Family	Train rows	Held out	Role
`data_v5`	E	633,213	label wordings	Evidence (NLI, BoolQ) and wide-vocabulary classification
`data_kb`	K	147,447	TruthfulQA-MC1, GPQA	Closed-book multiple-choice knowledge
`data_wf`	W	72,006	3 of 15 families	Rubric-conditioned workflow (Noul, Choice, Score)
`data_wf_hf`	W	86,600	3 external test/demo splits	External workflow sets (forms, rules, distillation)
`data_wf_long`	W	23,318	–	Long-state (multi-paragraph policy) workflow rows
`data_wh`	W	60,605	3 domains, 2 styles, 1 grammar per level, level 7	DecisionMix v2 counterfactual rubric groups
`data_u`	U	31,029	4 synthetic parameter ranges	Soft-target uncertainty

DecisionMix v2 spreads its 60,605 training rows evenly over levels 1–6. A leak check against the public JevBench ids finds 0 of 231 hits.

A.18 Cached-state equivalence

An equivalence test on the real model checks that the cached-state pass of Section 3.2 computes the same decision as a forward pass on $\mathrm{concat}(x, q, A)$. Cached-state decisions match the full-concatenation path to within 10^{-3} across suffix chunk sizes 1, 2, 3, 5 and 64, are consistent under permutation of the queries in a batch, and repeat exactly.

A.19 Long-state corpus statistics

States in `data_wf_long` have p10/p50/p90 lengths of 1,190, 1,845 and 2,490 tokens. In the original render, with the case facts after the policy body, right-truncation at a 1,024-token window removed the facts from 98.8% of rows, and at 256 tokens from all of them. The facts-first render places the case inside the first 8% of the text, and `-drop_truncated` removes any row longer than the training window.