

StartLux-Decision: AutoThink Decision Models with Reinforcement Learning for Calibrated Deliberative Decisions

Xialie Zhuang^{1,2} Zunhai Su^{1,2} Jia Pan² Ngai Wong² Guanhua Chen^{3,*} Chuan-Wei Kuo^{1,*}

¹StartLux ²The University of Hong Kong ³Southern University of Science and Technology

Abstract: We introduce StartLux-Decision, state-of-the-art decision models spanning 0.8B–27B parameters, combining efficient single-pass prediction with adaptive, confidence-guided reasoning. The *Instant* models leverage bidirectional attention in full-attention layers to improve decision accuracy while preserving efficient single-pass inference. The *AutoThink* models selectively invoke explicit reasoning for uncertain decisions while answering confident cases directly. We develop a three-stage training pipeline that establishes shared decision capabilities, specializes the two variants, and equips AutoThink with explicit reasoning through reasoning supervision and reinforcement learning. To improve reasoning efficiency while preserving decision quality, we introduce Reinforcement Learning for Calibrated Deliberative Decisions (RLCD²). Extending decision-oriented reinforcement learning to explicit reasoning, RLCD² uses a frozen model and proper scoring rules to reward improvements in decision probability quality, while imposing a soft penalty on overlong reasoning traces. RLCD² increases the proportion of reasoning traces that terminate within the generation limit by up to 48 percentage points and reduces average reasoning length by up to 65%, while largely preserving decision accuracy. On the public subset of Decision Index 0.3, the Instant models outperform the strongest evaluated comparable-scale open-weight decision models by 7.4–16.9 points, with the 27B AutoThink model further improving its score to 73.4. We also introduce DecisionPivot, a programmatically verifiable benchmark of computational decisions featuring contrast pairs that differ in a single answer-changing detail. On DecisionPivot, AutoThink consistently outperforms its Instant counterparts by 11.5–18.4 percentage points in pair accuracy and achieves leading performance among evaluated systems, highlighting the benefits of adaptive reasoning for computation-intensive decisions. To facilitate practical deployment, we support inference across vLLM, llama.cpp, and MLX, together with quantized model variants for reduced memory requirements. We envision StartLux-Decision advancing a new generation of decision models that adaptively allocate reasoning to challenging decisions, balancing decision quality, computational efficiency, and reliability.

🤖 Models: [Hugging Face](#)

🔗 Code: [GitHub](#)

📝 Blog: [decision.startlux.com](#)

STARTLUX-DECISION

Score vs. size

DI 0.3 (public) · as of 11 October 2026 · StartLux-Decision against other decision models

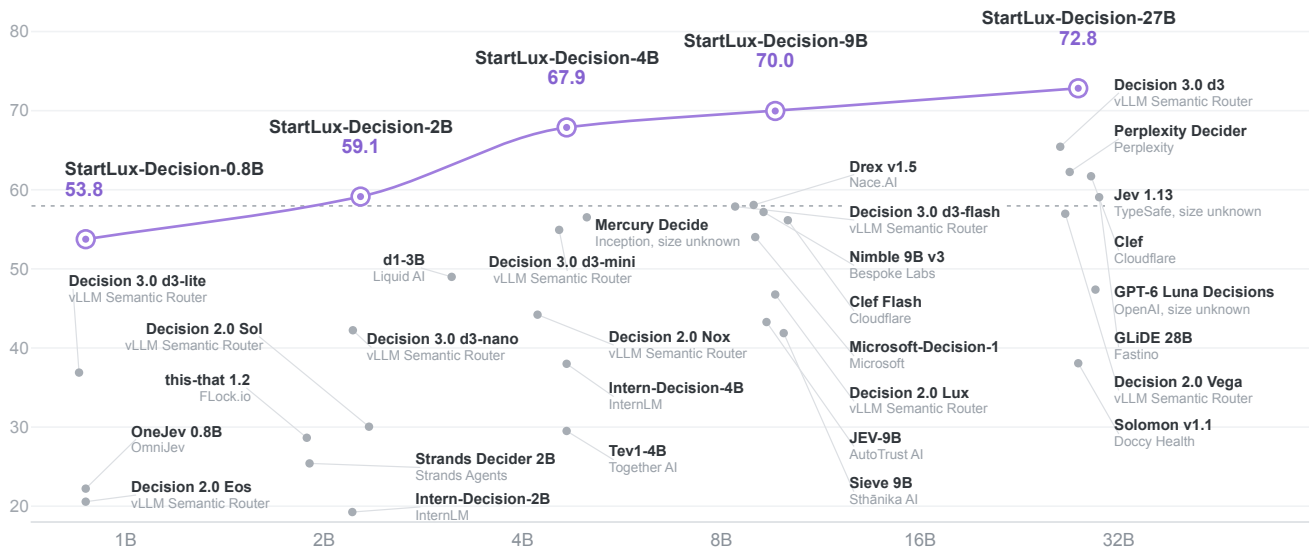


Figure 1. StartLux-Decision leads at every evaluated scale on the public subset of Decision Index 0.3.

*Corresponding authors.

Contents

1	Introduction	4
2	Model Design and Training	5
2.1	Decision Formulation and Probability Readout	5
2.2	Model Variants and Adaptive Inference	5
2.3	Three-Stage Training Pipeline	7
3	RLCD²: Reinforcement Learning for Calibrated Deliberative Decisions	10
3.1	Scoring Foundations and Reward Formulation	10
3.2	Policy Optimization and Training	11
3.3	Effects on Reasoning and Decision Quality	11
3.4	Reward Design and Reasoning Budget	12
4	Experimental Evaluation	15
4.1	Experimental Setup	15
4.2	Decision Index	15
4.3	DecisionPivot	16
4.4	Probability Calibration	17
5	Capability and Reliability Analysis	18
5.1	Effects of Training Data and Attention Design	18
5.2	Adaptive Reasoning and Prediction Fusion	19
5.3	Evidence Sensitivity and Probability Consistency	21
5.4	Policy-Induced Miscalibration	21
5.5	Policy-Aware Calibration	22
6	Efficient Inference and Deployment	23
6.1	Runtime Support	23
6.2	GPU Serving Performance	24
6.3	Local Inference on Apple Silicon	24
6.4	Quantized Inference	24
7	Conclusion	28
	References	28

A	Data Sources	30
A.1	Record Format	30
A.2	Data Composition Across Training Stages	30
A.3	Benchmark-Specific Training Coverage	30
A.4	Licensing and Data Usage	30
A.5	Generated and Model-Written Data	31
A.6	Data Integrity Checks	31
B	Training Implementation and Additional Analyses	31
B.1	Bidirectional Prompt Attention	31
B.2	Additional Training Ablations	32
B.3	Calibration and Resolution under Proper Scoring Rules	32
B.4	Direct Probability Optimization	32
C	Policy-Aware Calibration: Proofs and Details	33
C.1	Setting and Notation	33
C.2	Theoretical Properties	33
C.3	Evaluation Protocols	34
C.4	Results Across Decision Pipelines	35
C.5	Policy-Specific Threshold Certification	35
D	Evaluation Details	36
D.1	Decision Index Evaluation	36
D.2	GSM8K Benchmark Reconstruction	37
D.3	Evaluation Reliability and Serving Configuration	37
D.4	Handling Multiple Questions and Long Option Lists	37

1. Introduction

Decision-making is central to modern large language model (LLM) systems, underpinning tasks such as request routing, retrieval assessment, tool-call validation, and agent outcome verification (Gu et al., 2024). A *decision model* supports these operations by answering typed questions about an input state and producing probability distributions over candidate options (Tang and Zheng, 2026). Beyond identifying the most likely outcome, these probabilities enable downstream systems to decide whether to act autonomously, invoke a stronger model, or defer to human review. Existing open-weight decision models largely rely on single-pass prediction without explicit reasoning (Cloudflare, 2026; Perplexity AI, 2026). While efficient, this approach can struggle with computation-intensive decisions involving arithmetic, scheduling, and procedural verification, where multi-step reasoning may be essential (Wei et al., 2022; Snell et al., 2024). Moreover, we observe that decision-oriented training can substantially impair a model’s ability to generate and reliably terminate reasoning traces. Enabling token generation alone is insufficient to recover effective reasoning. These challenges raise a central question: *How can decision models incorporate reliable reasoning when needed while retaining efficient single-pass prediction and probabilistic decision outputs?*

To address this challenge, we introduce StartLux-Decision, a family of decision models spanning 0.8B–27B parameters, with two complementary variants sharing a probability interface. The *Instant* models leverage bidirectional attention in full-attention layers to improve decision accuracy while preserving efficient single-pass inference. This allows earlier prompt representations to incorporate information from subsequent questions and candidate options. The *AutoThink* models selectively invoke explicit reasoning for uncertain decisions while answering confident cases directly, combining initial and post-reasoning probabilities when needed. A three-stage training pipeline establishes shared decision capabilities, specializes the two variants, and equips AutoThink with explicit reasoning through reasoning supervision and reinforcement learning (Figure 2; Section 2).

We introduce Reinforcement Learning for Calibrated Deliberative Decisions (RLCD²), building on Reinforcement Learning for Calibrated Decisions (RLCD) (Almeida, 2026) to extend decision-oriented reinforcement learning to explicit reasoning. Rather than rewarding answer correctness alone, RLCD² optimizes the reasoning policy for the quality of the probabilistic decisions it supports. A frozen copy of the starting model provides decision probability readouts before and after reasoning, with their proper-score difference defining the decision-quality reward. A soft length penalty further discourages unnecessarily long reasoning traces, balancing decision quality with computational efficiency. Empirically, RLCD² increases the proportion of reasoning traces terminating within the generation limit by up to 48 percentage points and reduces average reasoning length by up to 65%, while largely preserving decision accuracy (Section 3).

StartLux-Decision demonstrates strong performance across established decision benchmarks. On the public subset of Decision Index 0.3, the Instant models outperform the strongest evaluated comparable-scale open-weight decision models by 7.4–16.9 points, with the 4B Instant model surpassing all evaluated reference decision models (Figure 1). Aggregate benchmark scores, however, can obscure sensitivity to small but decision-critical input changes. We therefore introduce DecisionPivot (StartLux, 2026), a programmatically verifiable benchmark of computational decisions featuring contrast pairs that differ in a single answer-changing detail. By requiring both decisions in each contrast pair to be correct, DecisionPivot evaluates whether models can reliably distinguish between closely related cases with different correct answers. On DecisionPivot, AutoThink consistently outperforms its Instant counterparts by 11.5–18.4 percentage points in pair accuracy, with the 27B AutoThink model reaching 86.9% and leading all evaluated reference systems (Section 4).

Beyond predictive accuracy, the reliability of decision probabilities also depends on how downstream systems select and route predictions. Calibration of individual predictions does not necessarily carry over to these downstream policies. For example, predictions assigned 80–90% confidence by our 9B verifier achieve 92% accuracy when evaluated individually, but only 78% after selecting the highest-scoring response from 32 candidates. This discrepancy reveals how downstream selection can distort confidence–accuracy relationships even when the underlying model remains unchanged. We introduce policy-aware calibration, which incorporates selection and routing information to calibrate confidence under the deployed policy, reducing confidence–accuracy gaps to within 2.7 percentage points across 184 settings spanning four decision pipelines (Section 5). For practical deployment, we support both variants across vLLM, llama.cpp, and MLX. We also provide quantized model variants to reduce memory requirements and facilitate deployment across diverse hardware environments (Section 6).

Together, these contributions establish StartLux-Decision as a state-of-the-art decision model family that combines strong predictive performance with efficient single-pass inference and adaptive reasoning through a shared probabilistic decision interface. More broadly, we envision a new generation of decision models that not only make accurate predictions but also determine when additional reasoning is warranted and how confidently their outputs should guide downstream actions, balancing decision quality, computational efficiency, and reliability. We publicly release the model weights and inference code to enable independent evaluation, facilitate practical deployment, and advance further research on adaptive and reliable decision-making. Our implementation and deployment resources are available at <https://github.com/StartLuxLabs/StartLux-Decision>.

2. Model Design and Training

StartLux-Decision comprises two complementary variants, Instant and AutoThink, designed for efficient single-pass prediction and adaptive reasoning, respectively. We develop a novel three-stage training pipeline that establishes foundational decision capabilities, specializes the two variants through task and attention adaptation, and equips AutoThink with explicit reasoning through reasoning supervision and reinforcement learning.

2.1 Decision Formulation and Probability Readout

Existing open-weight decision models typically answer structured questions about an input state by assigning probabilities to candidate options, often through a single forward pass (Tang and Zheng, 2026; Cloudflare, 2026; Perplexity AI, 2026).

Typed decisions. A decision request comprises an input state and a set of named, typed questions concerning that state. The input state may be represented as a string or any JSON value, optionally accompanied by images. Each question specifies a choice among named options, a binary decision, or a level on an ordered scale with 2 to 10 levels. Candidate options may include natural-language criteria that further define their intended semantics. For each question, the model produces a probability distribution over candidate options, providing both a predicted outcome and a confidence estimate to guide downstream decisions.

One-pass readout. Existing open-weight decision models typically produce probability distributions over candidate options through a single forward pass, without autoregressive token generation. These probabilities can be obtained through dedicated prediction heads or directly from the output logits of a pretrained language model. In the latter approach, each question is formatted as a prompt containing the input state, the question, and candidate options mapped to output tokens, whose logits are read from the language-model head at the final prompt position. For a question with K candidate options, let z_k denote the logit associated with option k . The decision probabilities can then be obtained by applying temperature scaling and a softmax over the candidate options:

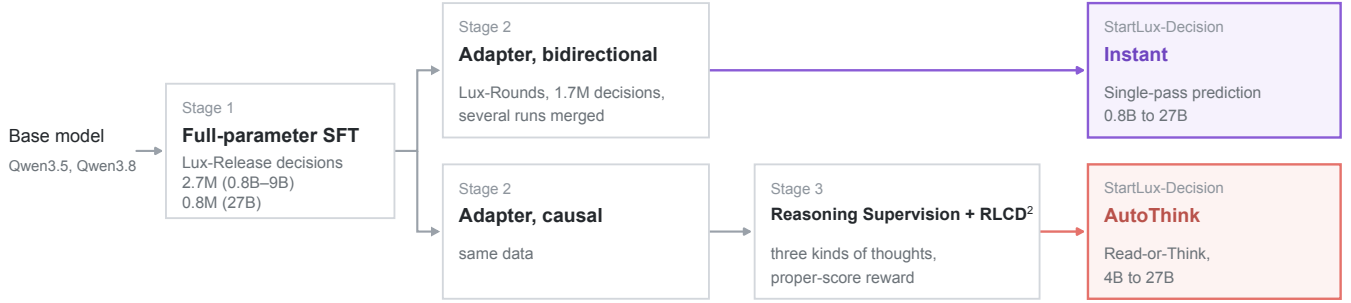
$$p_k = \frac{\exp(z_k/T)}{\sum_{j=1}^K \exp(z_j/T)}, \quad k = 1, \dots, K, \quad (1)$$

where $T > 0$ is a temperature parameter that can be fitted on held-out data for probability calibration.

2.2 Model Variants and Adaptive Inference

Shared probability readout. StartLux-Decision adopts an option-letter probability readout that maps each decision question to a separate prompt containing the input state, question, and candidate options labeled with single letters (A–Z). The corresponding logits are extracted at the final prompt position through the existing language-model output projection, without introducing an additional prediction head. The same readout is used for Instant’s single-pass predictions and AutoThink’s initial and post-reasoning probability estimates. Each readout supports up to 26 candidate options. Longer lists are processed in parallel groups of at most 25 options, with the top three candidates from each group advancing to a final round (Appendix D.4). Questions within a request are processed as separate batch rows. Both variants support contexts of up to 262,144 tokens and process images through the pretrained vision tower, which remains frozen during training.

a Training



b Inference: Read-or-Think

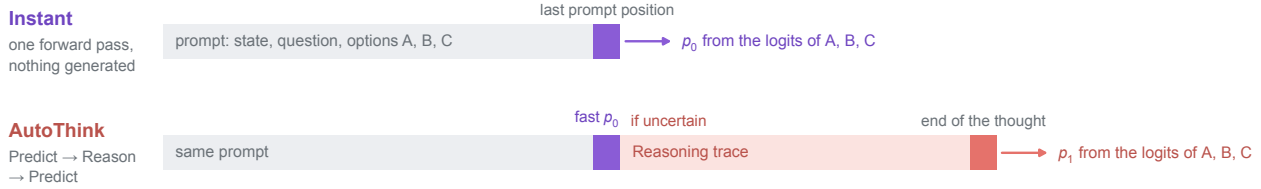


Figure 2. Training and inference workflows of StartLux-Decision. (a) Three-stage training establishes shared decision capabilities, specializes the Instant and AutoThink variants, and develops explicit reasoning in AutoThink through reasoning supervision and reinforcement learning. Instant adopts bidirectional attention in full-attention layers, while AutoThink retains causal attention. (b) Instant produces decision probabilities in a single forward pass, whereas AutoThink first obtains an initial prediction p_0 and selectively invokes reasoning based on its confidence. When reasoning is invoked, the post-reasoning prediction p_1 is combined with p_0 to form the final decision.

Instant models. The *Instant* models follow the single-pass decision paradigm, producing decision probabilities without autoregressive token generation. Since the complete prompt is available before prediction, we enable bidirectional attention in the full-attention layers, allowing earlier prompt representations to incorporate information from subsequent questions and candidate options. The linear-attention layers remain causal, preserving the underlying hybrid architecture while adapting full-attention computation to the single-pass decision setting.

AutoThink models and the default answer. The *AutoThink* models extend single-pass prediction through *Read-or-Think*, a confidence-guided inference mechanism that selectively invokes explicit reasoning for uncertain decisions while answering confident cases directly. For each question, AutoThink first performs a causal one-pass readout to obtain an initial probability distribution p_0 over candidate options. If the maximum predicted probability is at least the confidence threshold $g = 0.9$, the model returns p_0 without additional reasoning. Otherwise, it reprocesses the prompt with reasoning enabled, generates a reasoning trace of up to 4,096 tokens, and obtains a post-reasoning distribution p_1 from the option-letter logits following the trace. The reasoning process and both probability readouts share the same model weights. When reasoning is invoked, AutoThink combines the initial and post-reasoning probability distributions to form its final prediction, rather than relying exclusively on the post-reasoning estimate. The final *default answer* is defined as

$$p_{\text{default}} = \begin{cases} p_0, & \max_k p_{0,k} \geq g, \\ (1 - \alpha)p_0 + \alpha p_1, & \max_k p_{0,k} < g, \end{cases} \quad g = 0.9, \quad \alpha = 0.75. \quad (2)$$

The confidence gate is applied independently to each question, and the runtime closes any reasoning trace that reaches the generation limit without terminating. Section 5.2 examines the effects of the gate threshold and fusion weight. Figure 3 illustrates Read-or-Think using two cases from the same data-export policy in DecisionPivot. The 27B AutoThink model answers the confident case directly but invokes reasoning for the uncertain case, where additional computation changes the predicted decision. Across the 7,800 core DecisionPivot items, the model answers 39% of questions directly and improves accuracy from 57% to 93% on the remaining reasoning-invoked cases. Table 1 provides the corresponding probability readouts and generated reasoning trace.

One data-export policy

Rule 3: escalate to a human reviewer when the data is under legal hold, but not for exports under 100 records. Rule 4: approve when the export is under 500 records and there were fewer than 2 exports in the last 30 days. Rules 1 and 2 do not apply here. When rules conflict, the lower number wins.

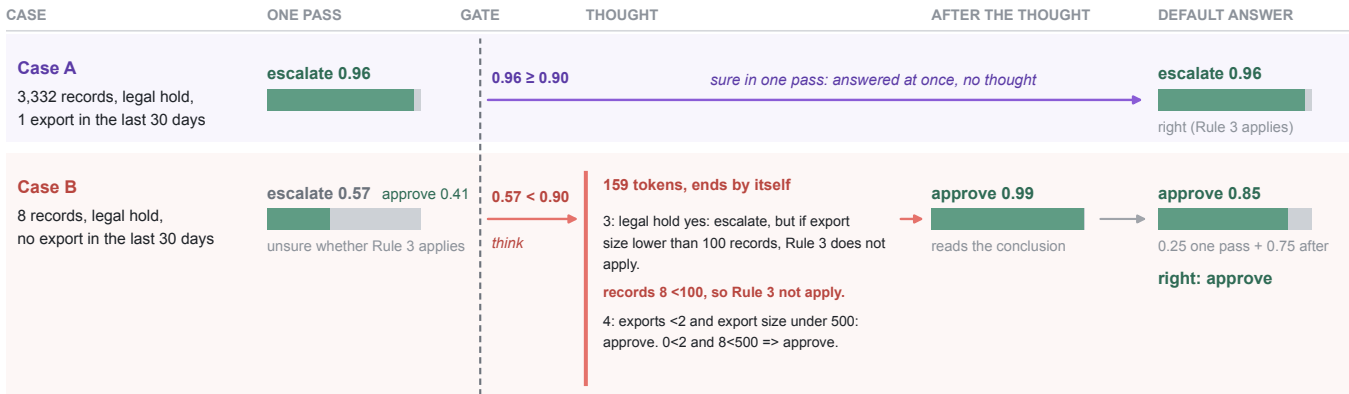
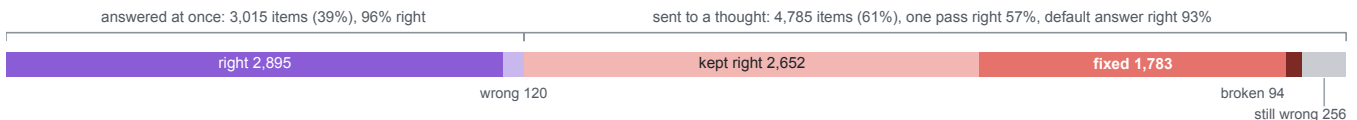
**The same model on all 7,800 DecisionPivot core items**

Figure 3. AutoThink selectively invokes reasoning for uncertain decisions. Top: Two cases from the same DecisionPivot policy illustrate confidence-guided inference, with direct prediction for a confident case and reasoning-assisted correction for an uncertain case. Bars show decision probabilities, with correct options highlighted in green. Bottom: Across 7,800 core DecisionPivot items, the 27B AutoThink model makes direct predictions on 39% of cases with 96% accuracy and invokes reasoning for the remaining 61%. On reasoning-invoked cases, accuracy improves from 57% before reasoning to 93% after default-answer fusion, with 1,783 initial errors corrected and 94 new errors introduced.

2.3 Three-Stage Training Pipeline

Our three-stage training pipeline develops the complementary capabilities of Instant and AutoThink. Stages 1 and 2 establish and specialize probabilistic decision prediction, while Stage 3 equips AutoThink with explicit reasoning through supervision and reinforcement learning, preserving its single-pass prediction capability.

2.3.1 Stage 1: Full-Parameter SFT

Stage 1 establishes the foundational decision capabilities shared by all StartLux-Decision models through full-parameter supervised fine-tuning (SFT) on Lux-Release, our initial training mixture.

Training Data. Lux-Release integrates diverse public datasets reformulated as typed decisions, spanning classification, natural language inference, question answering, safety, preference modeling, and tool-use tasks. Additional sources include community collections of typed decisions, benchmark training splits, multiple-choice questions covering knowledge and commonsense reasoning, and a smaller portion of programmatically generated examples (Appendix A). The training mixture contains approximately 2.7 million examples for models ranging from 0.8B to 9B parameters, whereas the 27B model uses approximately 0.8 million examples, with a greater proportion of benchmark training splits and multiple-choice questions. To mitigate answer-position bias, we deterministically permute candidate options for each example using a hash-based procedure. To reduce training-evaluation overlap, we exclude examples matching the Decision Index test sets or any of the 231 public JevBench v1.4.2 items (Benchmark Heaven, 2026), together with all other examples belonging to the same parent groups (Appendix A).

Training Procedure. We fine-tune all language-model parameters, including the output projection, using soft cross-entropy between the target and predicted probability distributions over candidate options at the final prompt position. This objective directly optimizes the option-letter probability readout without introducing an additional

Table 1. Selective reasoning corrects an initial decision error in policy interpretation. A 27B AutoThink example from the programmatically generated DecisionPivot benchmark illustrates how reasoning identifies a policy exception and corrects an initially incorrect prediction. The reasoning trace is reproduced verbatim after the runtime’s fixed initial line, without added line breaks. The final row provides a contrasting confident case under the same policy, answered without reasoning.

Step	What the model reads and writes
Policy	(1) When the verified requester flag is set to no and the number of exports in the last 30 days is 3 or more: Deny the request. (2) When the legal hold flag is set to yes and the data region is anything other than Branzim and the number of exports in the last 30 days is greater than 3: Deny the request. (3) When the case has the legal hold flag set to yes: Escalate the request to a human reviewer. (4) When the number of exports in the last 30 days is less than 2 and the export size is under 500 records: Approve the request. If the export size is lower than 100 records, Rule 3 does not apply. When rules conflict, the rule with the lower number wins. If no rule applies, the request is escalated to a human reviewer.
Uncertain case	8 records, 0 exports in the last 30 days, data region Branzim, legal hold yes, requester verified no, and seven other fields that no rule mentions
Question	What does the policy require for this case? Options approve, deny and escalate. The correct answer is approve.
Single-pass readout	escalate 0.567 , approve 0.408, deny 0.026. The maximum probability is below 0.9, so the model invokes reasoning.
Reasoning trace, 159 tokens	We need answer user’s question. Need analyze policy. Need choose exactly one letter. Case: records 8, exports_last_30_days 0, data_region Branzim, legal_hold yes, requester_verified no. Rules: 1: verified no and exports >=3? no (0) 2: legal hold yes and region other than Branzim and exports >3? no 3: legal hold yes: escalate, but if export size lower than 100 records, Rule 3 does not apply. records 8 <100, so Rule 3 not apply. 4: exports <2 and export size under 500: approve. 0<2 and 8<500 => approve.
Post-reasoning readout	approve 0.995 , escalate 0.005, deny 0.001
Default answer	approve 0.848 , escalate 0.145, deny 0.007; the predicted option is correct
Confident case	Same policy; 3,332 records, 1 export in the last 30 days: escalate 0.959 in one pass. The prediction is correct and exceeds the confidence threshold, so reasoning is not invoked

prediction head. Following training, we fit a separate temperature for each question type on held-out data to calibrate the predicted probabilities without changing the ranking of candidate options. All fitted temperatures exceed one, indicating a consistent tendency toward overconfidence before calibration.

2.3.2 Stage 2: Adapter-Based Decision Training

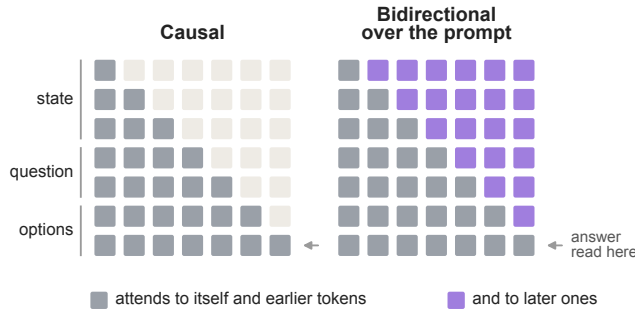
Stage 2 extends the shared decision foundation along two complementary dimensions: broader task coverage and attention adaptation. We train adapters on Lux-Rounds, our second training mixture, and subsequently merge them into the Stage 1 model weights. The Instant models adopt bidirectional attention in full-attention layers, while the AutoThink models retain causal attention to support subsequent reasoning training.

Training Data. Lux-Rounds comprises approximately 1.7 million decision examples designed to broaden task coverage beyond Stage 1. Most examples are drawn from benchmark training splits and related public datasets, supplemented by *breadth* data covering decision types not represented in the Decision Index. Additional sources include programmatically generated procedural decisions and select-all questions, API-generated examples, and teacher-labeled decisions. We retain API-generated examples only when two independent verifier models agree with their labels. A subset of Lux-Release is also retained to preserve the capabilities established in Stage 1. GSM8K contributes only training problems formatted according to the revised DI 0.3 benchmark.

Training Procedure. We optimize the adapters using the same decision objective as Stage 1 before merging them into the model weights. Each released Instant model is obtained by uniformly averaging the weights of three independently trained runs at 0.8B or four runs at 2B–27B, differing only in their random seeds. Following weight averaging, we refit the temperature parameters for probability calibration. The AutoThink models use the same Stage 2 data but retain causal attention, preparing them for autoregressive reasoning in Stage 3.

Bidirectional Attention over the Prompt. We train the Instant models with bidirectional attention in their full-attention layers, allowing earlier prompt representations to incorporate information from the question and candidate

a Attention over one prompt



b Same data, every size

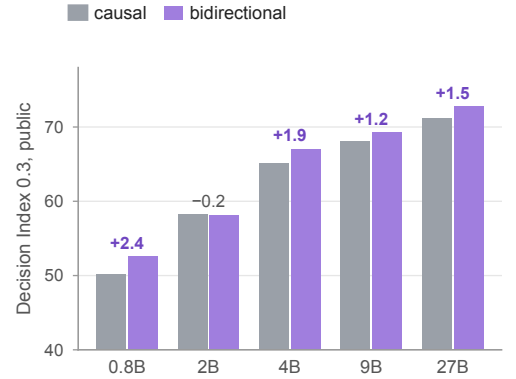


Figure 4. Bidirectional attention improves single-pass decision performance across most evaluated model scales. (a) Causal and bidirectional attention patterns in full-attention layers, with linear-attention layers remaining causal and decision probabilities read from the final prompt position. (b) DI 0.3 performance of Stage 2 models trained on identical data from the same Stage 1 checkpoint, differing only in their attention configurations. Runs with identical training configurations exhibit variations of 0.2–0.7 index points across random seeds.

options that follow them. At 4B and 9B, this modification applies to 8 of the 32 layers, while the linear-attention layers and option-letter readout remain unchanged (Figure 4a). We train with the modified attention pattern rather than applying it only at inference, since changing the attention mask of a causally trained model can disrupt its predictions (BehnamGhader et al., 2024). Perplexity Decider v1.1 adopts a similar bidirectional attention design.

Controlled comparisons using identical training data and the same Stage 1 checkpoint show that bidirectional attention improves DI 0.3 performance at four of the five evaluated model scales, with gains of up to 2.4 points (Figure 4b). Additional evaluations show improvements on DI 0.2.1 and other benchmark-related tasks, providing further evidence of its effectiveness beyond DI 0.3. These results support bidirectional attention as an effective adaptation for single-pass decision prediction across most evaluated scales and settings.

2.3.3 Stage 3: Reasoning Supervision and RL

Stage 3 equips the causal Stage 2 models with explicit reasoning capabilities for decisions requiring multi-step computation, while preserving efficient single-pass prediction. We first apply reasoning supervision to improve reasoning generation and termination, followed by reinforcement learning to optimize reasoning for decision quality and computational efficiency. The reasoning process and both probability readouts share model weights.

Reasoning Supervision. Decision-oriented training can substantially impair a model’s ability to generate and reliably terminate reasoning traces. In our experiments, the decision-trained 4B model produces little or no reasoning for approximately 40% of questions on which it is prompted to reason, while the 9B model continues generating until the length limit on approximately 70% of such questions. These contrasting failure modes highlight the need to train both reasoning generation and termination, with the latter also posing challenges for general reasoning models (Chen et al., 2025). We first train an adapter on Lux-Think, using approximately 4,000–21,000 supervised reasoning examples per model. Related work has explored supervised training, distillation, and reinforcement learning to develop reasoning capabilities in smaller models (Zhuang et al., 2025). Following reasoning supervision, all evaluated AutoThink models achieve higher post-reasoning accuracy than with single-pass prediction.

Despite these improvements, reasoning supervision does not fully resolve termination failures or excessive reasoning length. We therefore introduce RLCD² to optimize the reasoning policy for probabilistic decision quality, rewarding reasoning-induced improvements in decision probabilities while imposing a soft penalty on overlong reasoning traces. This final training stage further improves reasoning efficiency while largely preserving decision accuracy, as detailed in Section 3.

3. RLCD²: Reinforcement Learning for Calibrated Deliberative Decisions

Reinforcement Learning for Calibrated Decisions (RLCD) was introduced alongside Jev, a hosted decision model whose training procedure has not been publicly disclosed (Almeida, 2026). OpenJev-RLCD subsequently explores a two-stage approach that calibrates post-rationale probability readouts before optimizing reasoning with proper scoring rules (Gao and Wang, 2026). Building on this line of research, we propose RLCD², extending decision-oriented reinforcement learning to explicitly optimize the contribution of reasoning to probabilistic decision quality. Our formulation combines a frozen probability scorer, a decision-weighted proper scoring rule, and a soft length penalty to balance decision quality with reasoning efficiency.

3.1 Scoring Foundations and Reward Formulation

Proper Scoring Rules. We employ proper scoring rules to evaluate the full probability distribution over candidate decisions rather than only the correctness of the selected option. For a predicted distribution $p = (p_1, \dots, p_K)$ and a target distribution t , a scoring rule $S(p, t)$ is *proper* if its expected value is maximized by reporting the true outcome distribution, and *strictly proper* if this maximizer is unique.

We consider three scoring rules. The Brier score $S_B(p, t) = -\|p - t\|_2^2$ and logarithmic score $S_L(p, t) = \sum_{k=1}^K t_k \log p_k$ evaluate probability quality through squared error and log likelihood, respectively, with the latter corresponding to the negative cross-entropy objective used in Stages 1 and 2. To account for confidence-sensitive decisions, we further adopt a decision-weighted proper scoring rule that emphasizes probability errors near operational confidence thresholds. For an option with reported probability q and binary outcome $z \in \{0, 1\}$, define

$$s(q, z) = zW(q) - C(q), \quad W(q) = \int_0^q w(c) dc, \quad C(q) = \int_0^q c w(c) dc. \quad (3)$$

Summing the option-wise scores yields the decision-weighted score $S_D(p, t) = \sum_{k=1}^K s(p_k, t_k)$. For an option with true correctness probability π , the expected score satisfies

$$\frac{\partial}{\partial q} \mathbb{E}[s(q, z)] = w(q)(\pi - q). \quad (4)$$

The expected score is maximized at $q = \pi$ for nonnegative w , with a unique maximizer when w is strictly positive over the relevant probability range. We define $w(c)$ as an equal mixture of a uniform density and a Beta(8, 2) density, placing greater emphasis on high-confidence decisions. For ordered response levels, the score is applied to cumulative probabilities.

Reasoning Reward. Many reasoning-oriented reinforcement learning methods primarily reward answer correctness (Guo et al., 2025; Ma et al., 2025; Yu et al., 2025). In contrast, RLCD² evaluates changes in probabilistic decision quality following reasoning while accounting for generation cost. Let $p_0(x)$ denote the initial single-pass probability distribution for input x , and $p_1(x, \tau)$ the distribution obtained after reasoning trace τ , both produced by a frozen copy of the starting model. We define the reasoning reward as

$$r(x, \tau) = S(p_1(x, \tau), t) - S(p_0(x), t) - \lambda \max(0, |\tau| - B), \quad (5)$$

where S is the decision-weighted score used for the released models, B is the soft reasoning-length budget, and $\lambda = 10^{-4}$ controls the penalty per token beyond this budget. The proper-score difference quantifies the change in probabilistic decision quality following reasoning, while the length penalty discourages excessive computation without imposing a hard truncation at B . Related approaches also penalize overlong generations, including DAPO (Yu et al., 2025), while Kimi k1.5 favors shorter responses relative to other samples for the same prompt (Kimi Team et al., 2025).

Table 2. RLCD² improves reasoning termination and efficiency while largely preserving held-out accuracy. Results compare reasoning-supervised and post-RLCD² checkpoints of the 4B, 9B, and 27B AutoThink models on 2,986 held-out examples, after excluding 35 examples overlapping with training or RL data. Reasoning is invoked for every example, with one trace and a 4,096-token generation limit. Accuracy is reported in percent, and default-answer probabilities combine post-reasoning and single-pass predictions with weights of 75% and 25%, respectively (Section 5.2).

Checkpoint	Single-pass accuracy	Post-reasoning accuracy	Default-answer accuracy	Termination rate	Trace tokens: mean / 90th percentile
4B					
After reasoning supervision	72.04	73.91	74.35	89.5%	1,140 / 4,096
After RLCD²	72.10	74.61	74.82	99.3%	722 / 1,358
9B					
After reasoning supervision	79.37	79.97	80.38	45.9%	2,494 / 4,096
After RLCD²	79.44	80.27	80.48	93.9%	868 / 2,658
27B					
After reasoning supervision	77.13	78.77	79.07	97.2%	313 / 446
After RLCD²	76.99	79.17	79.54	97.4%	303 / 435

3.2 Policy Optimization and Training

A frozen copy of the reasoning-supervised starting model provides both probability readouts for reward computation, while all language-model parameters of the reasoning policy are optimized. Traces terminating within eight tokens are treated as empty reasoning and assigned zero reward. Training is restricted to questions whose maximum initial option probability is below 0.9, matching the confidence gate used at inference.

We optimize the expected reward using group-relative policy gradients (Shao et al., 2024; Guo et al., 2025). For each prompt, we sample G reasoning traces, center their rewards within the group, and clip the resulting advantages as $A_i = \text{clip}(r_i - \bar{r}, -5, 5)$. The clipped surrogate objective, regularized toward the starting policy, is

$$\mathcal{L} = -\frac{1}{N} \sum_{i \in \mathcal{T}} \sum_{s=1}^{|\tau_i|} \min \left(\rho_{i,s} A_i, \text{clip}(\rho_{i,s}, 1 - \epsilon, 1 + \epsilon) A_i \right) + \beta \widehat{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}). \quad (6)$$

Here $\mathcal{T}$ indexes the sampled traces in the training batch, N is the total number of generated tokens, and $\rho_{i,s}$ is the probability ratio between the current and sampling policies for token s in trace i . The KL estimator averages $\pi_{\text{ref}} / \pi_\theta - \log(\pi_{\text{ref}} / \pi_\theta) - 1$ over generated tokens.

For groups containing only nonempty traces, the initial decision score is constant across traces and cancels under reward centering. Consequently, their relative advantages depend on post-reasoning scores and length penalties, while the initial score provides a reference for interpreting the incremental value of reasoning. Empty traces retain zero reward. Following Liu et al. (2025), we do not normalize advantages by the within-group standard deviation, allowing groups with smaller reward differences to contribute proportionally smaller updates.

3.3 Effects on Reasoning and Decision Quality

We evaluate the incremental effects of RLCD² by comparing reasoning-supervised checkpoints with their post-RLCD² counterparts on 2,986 held-out examples from the training distribution. Reasoning is invoked for every example, with one trace per question and a 4,096-token generation limit. We additionally evaluate 7,800 core DecisionPivot examples under the default confidence gate, using the pair-accuracy metric described in Section 4.3. Table 2 summarizes the held-out results, with default-answer probabilities obtained by combining single-pass and post-reasoning predictions (Section 2.2).

With a thought: the final models before and after RLCD², budget 3,072

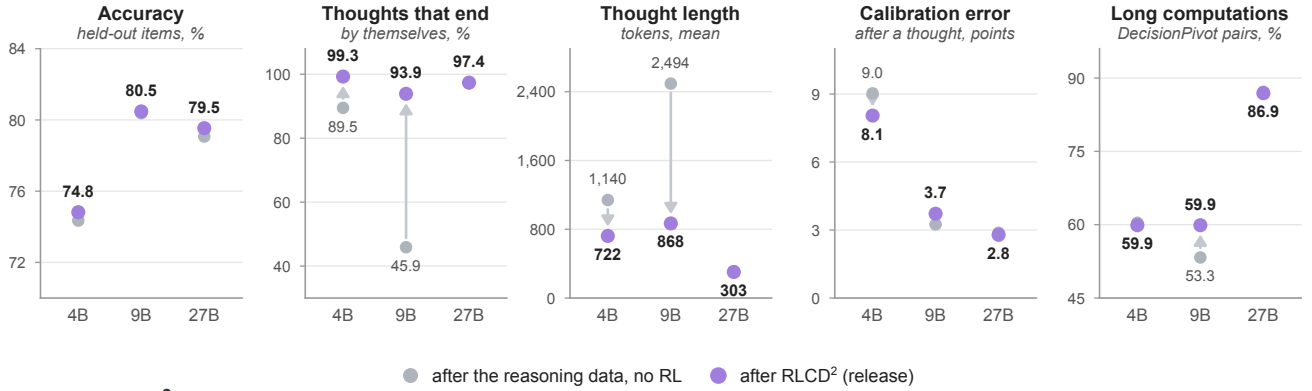


Figure 5. RLCD² improves termination and efficiency while largely preserving held-out accuracy. The 4B, 9B, and 27B AutoThink models before RL (gray) and after RLCD² (violet), using the decision-weighted score and a soft penalty budget of 3,072 tokens (distinct from the 4,096-token generation limit). Held-out metrics are measured on 2,986 examples with reasoning invoked for every example: default-answer accuracy, termination rate, mean trace length, and post-reasoning calibration error before temperature scaling. DecisionPivot pair accuracy is measured on 7,800 core items at the default confidence gate of 0.9.

Reasoning Termination and Efficiency. The primary benefits of RLCD² are improved reasoning termination and reduced generation cost (Table 2; Figure 5). At 9B, termination increases from 45.9% to 93.9%, while mean trace length decreases from 2,494 to 868 tokens, a 65% reduction. At 4B, termination rises from 89.5% to 99.3%, with mean trace length decreasing by 37% and the 90th percentile falling from 4,096 to 1,358 tokens. The 27B model already terminates 97.2% of traces with a mean length of 313 tokens, leaving limited room for further improvement.

Decision Quality and Calibration. Held-out default-answer accuracy remains largely unchanged across all three scales, with uncertainty intervals for the changes including zero. At 4B, raw post-reasoning calibration error decreases from 9.0 to 8.1 percentage points, with an interval of $[-1.7, -0.1]$ for the change. After temperature scaling fitted on development examples, however, the corresponding errors are comparable at 3.1 and 3.2 points. At 9B and 27B, calibration changes remain within statistical uncertainty, consistent with their lower initial errors.

On the held-out training distribution, the 4B model’s gains are primarily reflected in raw calibration rather than answer accuracy or confidence-based ranking, with most of its predictive improvement already established through reasoning supervision. On DecisionPivot, the 9B model exhibits a different pattern: pair accuracy increases from 53.3% to 59.9% under the 3,072-token budget, with an interval of $[+5.0, +8.3]$ for the change (Table 3). Together, these results show that RLCD² substantially improves reasoning efficiency while largely preserving held-out decision accuracy, with additional gains in calibration or benchmark accuracy depending on the model scale and evaluation setting. Appendix B.3 further discusses calibration and resolution under proper scoring rules.

3.4 Reward Design and Reasoning Budget

Reasoning Budget and Accuracy Trade-offs. The soft reasoning budget determines when the length penalty takes effect, governing the trade-off between reasoning efficiency and accuracy on computation-intensive decisions (Table 3). Before RL, the 4B AutoThink model generates approximately 2,100 tokens per trace on DecisionPivot and achieves 60.4% pair accuracy at the default confidence gate. A 512-token budget approximately halves reasoning length but reduces pair accuracy to 52.9%. Increasing the budget to 2,048 and 3,072 tokens restores pair accuracy to 59.2% and 59.9%, respectively, with the latter differing from the pre-RL checkpoint by -0.4 points (interval $[-1.8, +0.8]$). Removing the length penalty increases pair accuracy to 61.7% but produces 40% longer traces, with approximately one third failing to terminate at the 4,096-token generation limit, compared with about one tenth under the 3,072-token budget. Across these configurations, held-out accuracy changes by at most 0.6 points, while budgets of 512–3,072 tokens achieve approximately 99% termination. At the shorter budget, both Brier and decision-weighted rewards improve calibration and termination but reduce DecisionPivot accuracy, highlighting the cost of overly restrictive reasoning.

Table 3. Effects of reward design and reasoning budget on decision quality and reasoning efficiency. Comparison of reasoning-supervised checkpoints and selected reinforcement learning configurations for the 4B, 9B, and 27B AutoThink models. Default-answer accuracy, termination rate, mean trace length, and post-reasoning calibration error are evaluated on 2,986 held-out examples with reasoning invoked for every example; DecisionPivot pair accuracy is measured on 7,800 core items at the default confidence gate of 0.9. Accuracy is reported in percent and calibration error in percentage points, with temperature-scaled values in parentheses. Default-answer probabilities assign 75% weight to the post-reasoning readout (Section 5.2).

Checkpoint / RL reward	Default accuracy	Termination rate	Mean trace tokens	Post-reasoning calibration error	DecisionPivot pair accuracy
4B AutoThink model					
After reasoning supervision (no RL)	74.35	89.5%	1,140	9.03 (3.08)	60.4
Brier score, budget 512	74.68	99.3%	504	6.96 (3.06)	53.8
Decision-weighted score, budget 512	74.58	99.7%	511	6.29 (2.44)	52.9
Decision-weighted score, budget 2,048	74.28	99.3%	677	8.11 (2.55)	59.2
Decision-weighted score, budget 3,072 (release)	74.82	99.3%	722	8.05 (3.24)	59.9
Decision-weighted score, no length penalty	74.48	91.6%	1,186	9.38 (2.73)	61.7
9B AutoThink model					
After reasoning supervision (no RL)	80.38	45.9%	2,494	3.25 (2.15)	53.3
Decision-weighted score, budget 2,048	80.64	98.1%	601	3.29 (2.32)	61.1
Decision-weighted score, budget 3,072 (release)	80.48	93.9%	868	3.72 (1.42)	59.9
27B AutoThink model					
After reasoning supervision (no RL)	79.07	97.2%	313	2.88 (1.97)	87.2
Decision-weighted score, budget 3,072 (release)	79.54	97.4%	303	2.79 (2.00)	86.9

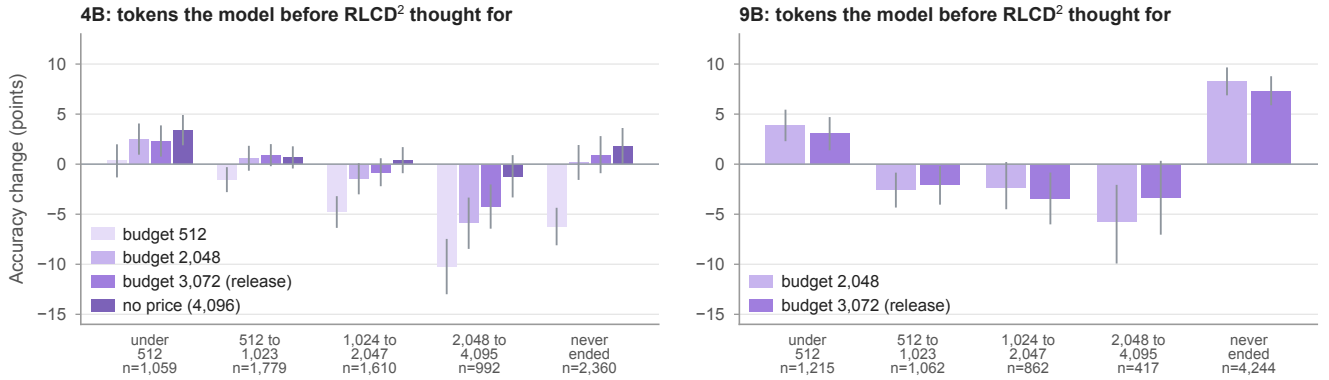


Figure 6. RLCD² improves decisions with previously unterminated traces but can reduce accuracy on long computations. Change in DecisionPivot accuracy at the default gate, after RL minus before RL, grouped by pre-RL trace length. Error bars show 95% intervals, and n denotes the number of items in each group.

Effects on Longer Computations. To identify where these trade-offs arise, we group DecisionPivot examples by their pre-RL reasoning lengths (Figure 6). At 4B, a 2,048-token budget improves accuracy by 2.5 points on examples with pre-RL traces shorter than 512 tokens but reduces accuracy by 5.9 points on those with traces of 2,048–4,095 tokens. Increasing the budget to 3,072 tokens reduces the loss in the latter group to 4.2 points.

At 9B, RLCD² yields gains on previously unterminated traces, which account for 54% of DecisionPivot examples before RL. Accuracy on these examples improves by 7.3 points, although examples with pre-RL traces of 2,048–4,095 tokens still experience losses of 3.4 and 5.8 points under budgets of 3,072 and 2,048 tokens, respectively. The 27B model exhibits reliable termination and short reasoning traces, with no statistically clear changes under RLCD². We adopt a training budget of 3,072 tokens across the released AutoThink models, which maintains 99.3% termination for the 4B model on held-out examples while limiting accuracy losses on longer computations.

Table 4. StartLux-Decision achieves leading performance across parameter scales on the public subset of DI 0.3. Scores on DI 0.3 and DI 0.2.1 are reported in points. Reference scores are drawn from the public leaderboard, except for Decision 3.0, which we evaluate using the same pipeline as StartLux-Decision ([‡]). The leaderboard excludes models that generate reasoning traces; a dash indicates an unavailable result.

System	DI 0.3	DI 0.2.1
27B		
StartLux-Decision-27B-AutoThink	73.37	72.60
StartLux-Decision-27B-Instant	72.83	72.12
Decision 3.0 d3 (vLLM Semantic Router) [‡]	65.44	64.77
Perplexity Decider v1.1	62.25	–
Jev (hosted)	57.96	57.91
9B		
StartLux-Decision-9B-AutoThink	70.45	69.65
StartLux-Decision-9B-Instant	69.96	69.04
Decision 3.0 d3-flash (vLLM Semantic Router) [‡]	57.88	57.08
Bespoke Nimble 9B v3	57.19	–
4B		
StartLux-Decision-4B-AutoThink	68.21	67.42
StartLux-Decision-4B-Instant	67.88	66.73
Decision 3.0 d3-mini (vLLM Semantic Router) [‡]	54.93	54.11
d1-3B (Liquid AI)	48.99	–
2B		
StartLux-Decision-2B-Instant	59.14	57.80
Decision 3.0 d3-nano (vLLM Semantic Router) [‡]	42.24	41.23
Decision 2.0 Sol (vLLM Semantic Router)	30.05	–
0.8B		
StartLux-Decision-0.8B-Instant	53.75	52.37
Decision 3.0 d3-lite (vLLM Semantic Router) [‡]	36.89	35.97
Decision 3.0 d3-edge, 0.6B (vLLM Semantic Router) [‡]	28.77	27.95
OneJev 0.8B (OmniJev)	22.22	–

Termination Dynamics and Reward Components. The substantial reduction in reasoning length at 9B is associated with repetitive generation loops that persist until the token limit. Early in RL training, 29% of 9B traces reach this limit, compared with 9% at 4B. Penalizing overlong traces lowers their rewards and the group mean, increasing the relative advantages of terminating trajectories and encouraging earlier termination. This pressure diminishes as repetitive loops disappear, particularly at 4B. In 4B training, rollouts shorter than 16 tokens generally receive negative relative advantages, although their share of the rollouts increases from 1% to 3% as termination improves.

Controlled reward comparisons on an earlier checkpoint further distinguish the contributions of the reward components. Including the length penalty produces nearly universal termination even when the predictive reward is replaced with random noise, whereas removing it results in longer traces and more termination failures. Calibration improvements, however, depend on the predictive reward: proper-score rewards outperform accuracy-only rewards, while replacing the length penalty with a termination bonus provides no calibration improvement. These findings indicate that the length penalty primarily drives termination improvements, whereas proper-score rewards contribute to probability calibration.

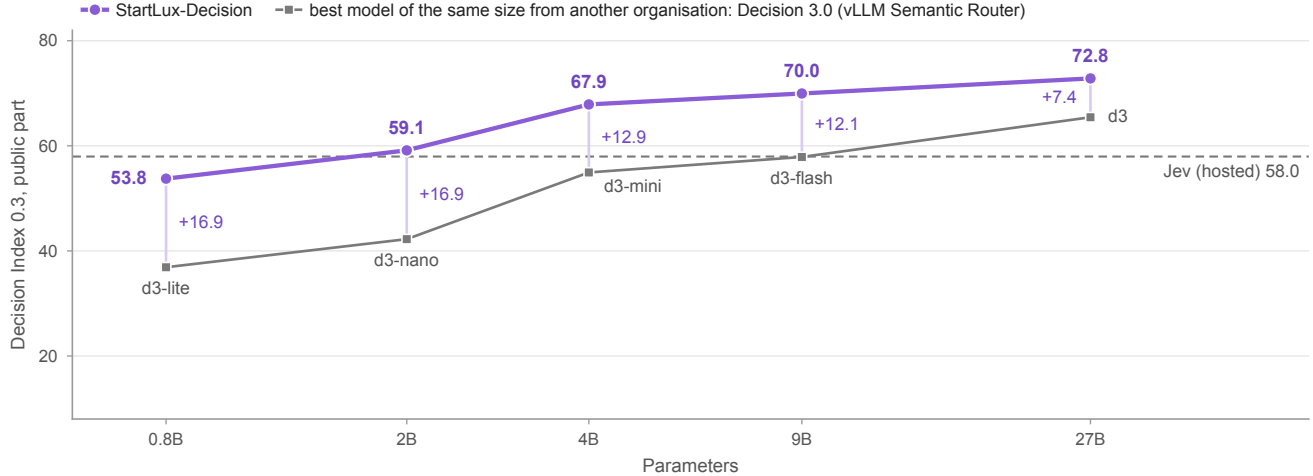


Figure 7. The Instant models outperform comparable-scale reference decision models by 7.4–16.9 points on the public subset of DI 0.3. Scores are plotted against parameter count. The gray line connects the strongest evaluated third-party model at each scale, all from the Decision 3.0 family of vLLM Semantic Router. Jev is hosted, with an undisclosed parameter count.

4. Experimental Evaluation

We evaluate decision accuracy and probability reliability across tasks with varying computational demands. Decision Index 0.2.1 and 0.3 assess performance on established benchmarks, while DecisionPivot examines computation-intensive decisions and sensitivity to answer-changing details through programmatically verifiable contrast pairs. We further evaluate probability calibration.

4.1 Experimental Setup

All StartLux-Decision models are evaluated under fixed inference configurations, including the prompt format, probability readout, and checkpoint-specific calibration temperatures described in Section 2. Neither model parameters nor calibration temperatures are fitted on the evaluation sets (Appendix D). Decision Index (DI) 0.2.1 aggregates chance-corrected scores across 38 benchmarks spanning five domains. DI 0.3 removes ForecastBench and replaces the original GSM8K questions with 2,638 reconstructed requests designed to eliminate answer-revealing cues in the candidate options. The public evaluation accounts for 20% of the DI 0.3 leaderboard score, while private evaluations covering existing capabilities and additional domains determine the remaining 80%. We evaluate only the public subset, with most evaluated tasks belonging to families represented during training. The public subset of DI also served as the development benchmark. We compared Stage 2 data recipes, prompt attention variants and the seed averages that form the Instant checkpoints by their DI public scores, so these scores reflect model selection and are not independent test results. Calibration temperatures and the AutoThink gate, mixture weight and reasoning budget were set on held-out items of the training distribution, not on DI. DecisionPivot was not used for any data, checkpoint or configuration choice, and it provides the independent evaluation in this report.

4.2 Decision Index

Table 4 compares StartLux-Decision with representative reference models on both editions of Decision Index. Across all five parameter scales, the Instant models outperform the strongest evaluated comparable-scale third-party decision models, all from the Decision 3.0 family, by 7.4–16.9 points on the public subset of DI 0.3 (Figure 7). Notably, the 4B Instant model outperforms all evaluated reference decision models, including substantially larger systems. The relative ordering of StartLux-Decision models remains consistent across DI 0.3 and DI 0.2.1.

The 27B Instant model outperforms all evaluated reference decision models across the five domains, while the 4B and 9B Instant models lead in four. Knowledge & Reasoning is particularly challenging for single-pass prediction,

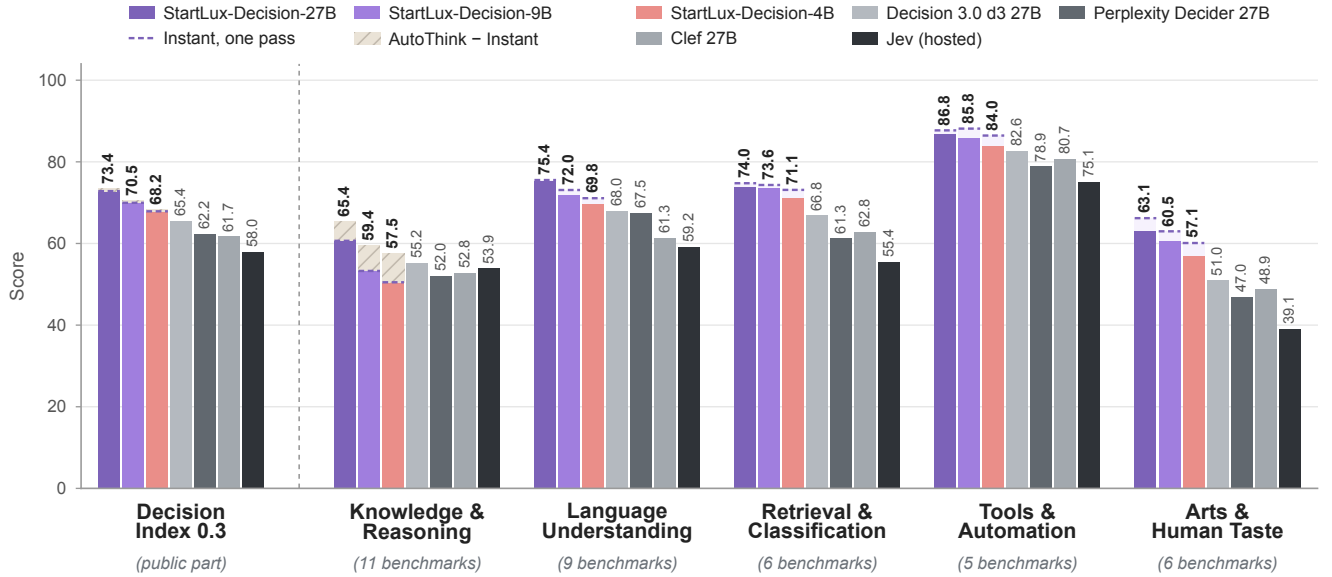


Figure 8. The 27B Instant model outperforms all evaluated reference decision models across the five DI 0.3 domains. Overall and domain scores are reported in points on the public subset, with domain weights matching the leaderboard. Solid bars show Instant performance, while hatched extensions represent the score differences between AutoThink and Instant at each scale. When AutoThink scores lower, the dashed line marks the higher Instant score. Reference scores come from the public leaderboard, except for Decision 3.0 d3, which we evaluate using the same pipeline as StartLux-Decision.

as many questions require explicit computation. In this domain, the 4B and 9B AutoThink models outperform their Instant counterparts by 7.0 and 6.2 points, respectively (Figure 8). Detailed results for individual Decision Index benchmarks are provided in Table D.1 of Appendix D.

4.3 DecisionPivot

DecisionPivot (StartLux, 2026) complements Decision Index by evaluating demanding decisions through programmatically verifiable contrastive examples. Its core track comprises 13 generated task families with 600 examples each, organized predominantly into contrast pairs that differ in a single answer-changing detail. Pair accuracy requires both examples to be answered correctly and is averaged across the 13 families. A separate *reused tasks* reference track comprises 35 public test sets, restricted to 34,104 examples absent from our training data.

The 27B Instant model achieves 75.4% pair accuracy, outperforming Decision 3.0 d3 (71.3%), Perplexity Decider v1.1 (70.1%), and the hosted Jev (31.0%), while GLM-5.3 reaches 82.2% with explicit reasoning (Figure 9). AutoThink substantially improves pair accuracy relative to Instant across all three evaluated scales, with gains of 18.4, 12.4, and 11.5 points at 4B, 9B, and 27B, respectively. At the default confidence threshold of 0.9, both the 4B and 9B AutoThink models achieve 59.9% pair accuracy, while the 27B model reaches 86.9%, outperforming all evaluated reference systems, including GLM-5.3. These gains are particularly pronounced on computation-intensive tasks.

The family-level results reveal distinct strengths across computational and structural decision tasks (Figure 10). Relative to their Instant counterparts, the 4B and 9B AutoThink models improve pair accuracy by approximately 51 and 56 points on filtering tasks, 30 and 44 on goal-state verification, and 46 and 25 on schedule-conflict detection, respectively. In contrast, single-pass prediction remains highly effective for tasks involving policy interpretation, clause changes, and dependency resolution.

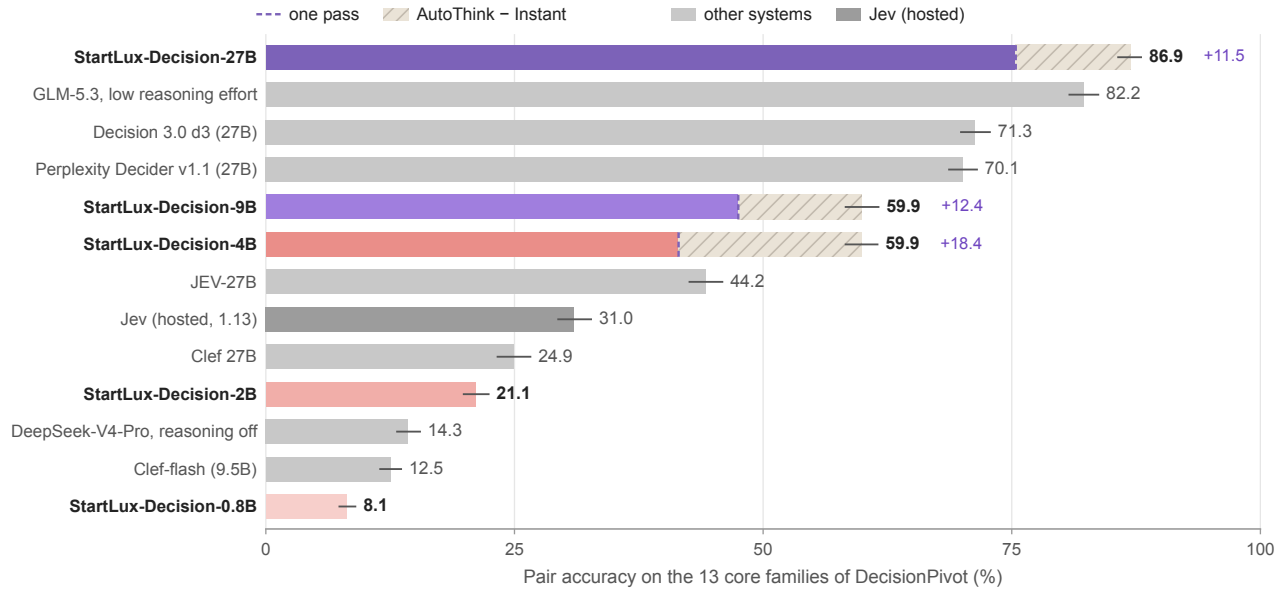


Figure 9. The 27B AutoThink model achieves the highest pair accuracy among evaluated systems on DecisionPivot. Results cover 7,800 examples across 13 programmatically generated task families, with 95% bootstrap confidence intervals. Solid bars show single-pass performance, while hatched extensions indicate the gains of AutoThink over Instant at the default confidence threshold of 0.9.

	answerable facts	clause change	dependency resolution	filter match	ledger reconcile	log count check	permission inheritance	policy case	record rule	schedule conflict	state goal	tool call valid	triage order
StartLux-Decision-27B AutoThink	93	81	85	84	88	84	80	86	94	85	94	94	84
StartLux-Decision-9B AutoThink	79	41	69	91	48	51	49	81	25	52	76	83	33
StartLux-Decision-4B AutoThink	78	59	53	91	45	71	28	80	23	69	72	73	37
StartLux-Decision-27B Instant	95	98	100	46	43	95	85	100	55	58	81	89	36
StartLux-Decision-9B Instant	74	39	73	35	34	65	43	94	7	27	32	69	24
StartLux-Decision-4B Instant	81	37	59	40	33	39	38	86	8	23	42	31	22
GLM-5.3, low reasoning effort	93	93	92	96	70	74	69	82	79	72	79	97	75
Perplexity Decider 27B	92	96	95	37	41	72	79	98	50	51	77	90	34
JEV-27B	72	59	42	26	36	32	34	36	35	36	56	87	24
Jev (hosted)	50	62	20	18	33	15	24	32	22	20	31	55	21
Clef 27B	39	26	12	14	31	11	29	28	23	22	28	45	15

Figure 10. AutoThink excels on computation-intensive decisions, while strong single-pass models remain competitive on structural tasks. Pair accuracy is reported in percent across the 13 DecisionPivot core families, with darker cells indicating higher scores. StartLux-Decision models are shown in violet and reference systems in gray.

4.4 Probability Calibration

Reliable decision-making requires well-calibrated probabilities, particularly when downstream systems use confidence to act or defer. On 7,800 DecisionPivot core examples, we compare predicted confidence, defined as the maximum option probability, with empirical accuracy across 15 confidence bins (Figure 11).

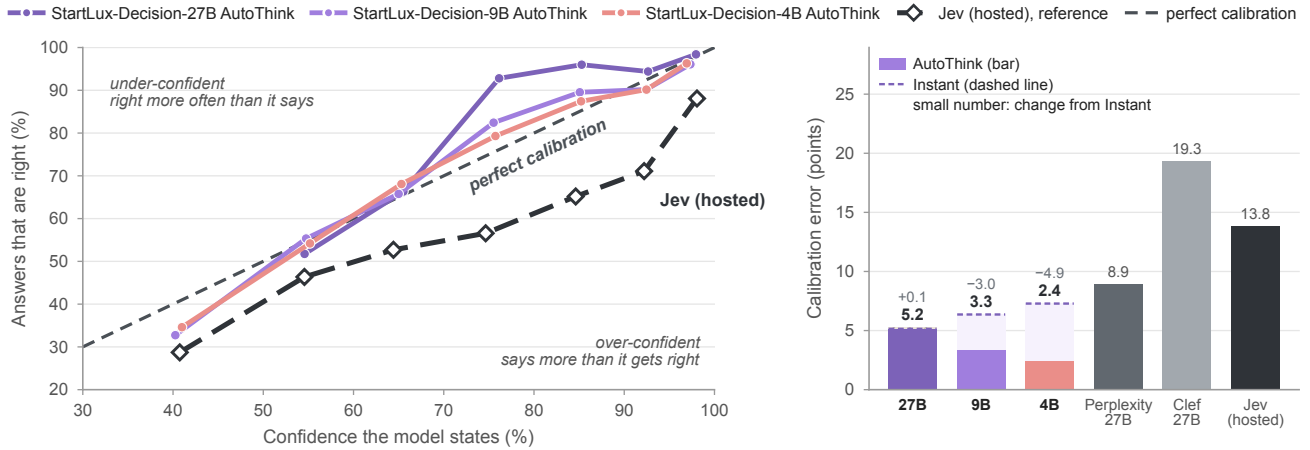


Figure 11. StartLux-Decision achieves lower calibration error than evaluated reference decision models on DecisionPivot. Left: empirical accuracy versus predicted confidence (maximum option probability) for the 4B, 9B, and 27B AutoThink models and hosted Jev across 7,800 core examples. Points below the diagonal indicate overconfidence. Right: calibration error based on confidence–accuracy gaps across 15 bins, reported in percentage points. Bars show AutoThink at the default confidence threshold of 0.9, with dashed lines marking the corresponding Instant models.

The evaluated StartLux-Decision models exhibit lower calibration error than the reference systems shown in Figure 11. The AutoThink models achieve calibration errors of 2.4, 3.3, and 5.2 percentage points at 4B, 9B, and 27B, respectively, compared with 7.3, 6.4, and 5.1 points for their Instant counterparts. The 27B AutoThink model is the only StartLux-Decision model that is underconfident on average.

Perplexity Decider v1.1, hosted Jev, and Clef exhibit calibration errors of 8.9, 13.8, and 19.3 points, respectively, with systematic overconfidence. Previous studies have examined Jev’s calibration on datasets (Deußner et al., 2026) and its probabilities for individual decisions (Porcedda, 2026). DecisionPivot extends this evaluation to programmatically verifiable decisions involving explicit computation. These results highlight probability reliability alongside predictive accuracy, while Section 5 further examines its behavior under selection and routing.

5. Capability and Reliability Analysis

Beyond overall benchmark performance, we investigate how training choices and adaptive reasoning shape decision performance, and assess the reliability of the resulting probability estimates. We further examine how downstream selection and routing affect confidence reliability and how policy-aware calibration addresses these effects.

5.1 Effects of Training Data and Attention Design

To distinguish gains from task-family exposure and attention adaptation, we evaluate additional tasks beyond Decision Index, whose public part also guided development choices (Section 4.1). Additional controlled training runs use the 4B model; analyses of the released AutoThink models cover the 4B, 9B, and 27B scales.

Effects of Training Data. Additional decision-training data primarily improve performance on represented task families. On the reused tasks of DecisionPivot, adding Stage 2 data from rating, safety, preference, and logic families improves the 18 corresponding tasks by 8.6 points, compared with only 0.3 points on tasks outside those families. Across the evaluated data groups, we find no clear improvement outside their respective task-family coverage. A similar pattern emerges for procedural decisions: adding related training families raises the 4B model’s accuracy on generated answerable-fact decisions from 61.5% to 75.8%. These results indicate task-family specialization, with limited evidence of transfer to unrepresented families.

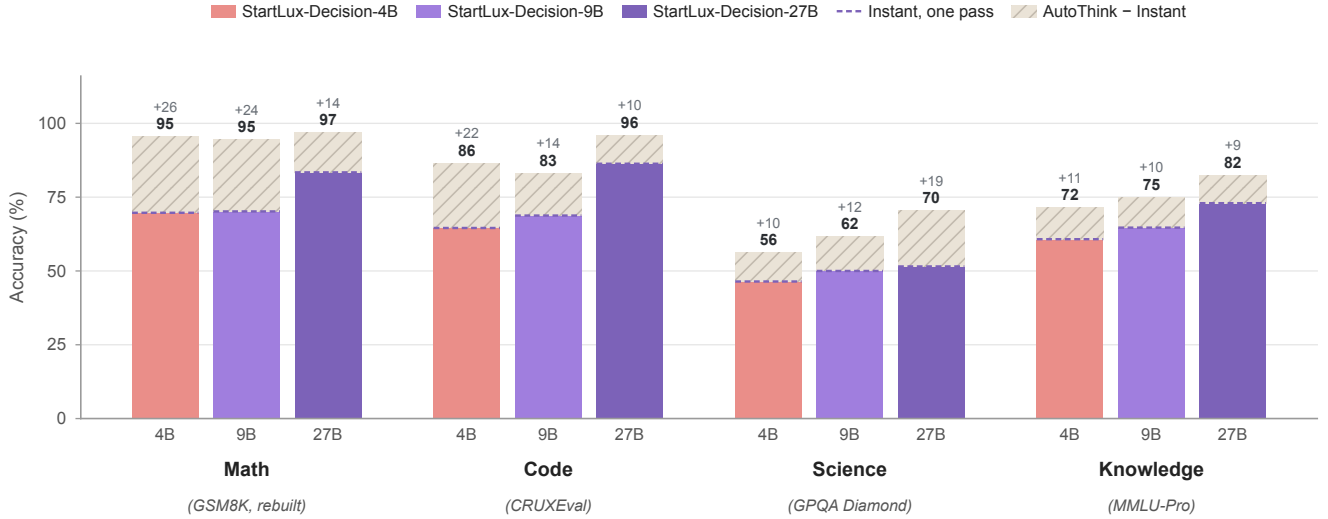


Figure 12. AutoThink outperforms Instant on computation-intensive tasks across all three evaluated scales. Solid bars show Instant accuracy, while hatched extensions indicate accuracy differences between the complete model variants. AutoThink uses the default confidence threshold of 0.9. Results cover the 4B, 9B, and 27B models on 2,638 reconstructed GSM8K problems from DI 0.3, along with CRUXEval, GPQA Diamond, and MMLU-Pro from DI 0.2.1 (Table D.1).

Effects of Bidirectional Attention. Bidirectional prompt attention improves performance beyond task families represented in Stage 2 training. With a narrower Stage 2 mixture, it improves performance by 3.0 points on 16 tasks whose families are absent from that mixture. This suggests that attention adaptation can improve single-pass prediction beyond gains attributable to task-family coverage.

5.2 Adaptive Reasoning and Prediction Fusion

Benefits of Adaptive Reasoning. AutoThink exhibits its largest advantages over Instant on tasks requiring multi-step computation (Figure 12). At 4B, the AutoThink model exceeds its Instant counterpart by 26 accuracy points on reconstructed GSM8K problems, 22 on CRUXEval, 10 on GPQA Diamond, and 11 on MMLU-Pro. The pattern persists at larger scales. At 9B, AutoThink exceeds Instant by 24.4 points on GSM8K and 14.4 on CRUXEval, with a 0.5-point overall DI 0.3 advantage. At 27B, Instant already achieves 83% accuracy on reconstructed GSM8K problems, with AutoThink scoring another 13.6 points higher. AutoThink also leads by 19 points on GPQA Diamond, 10 on CRUXEval, and 9 on MMLU-Pro, while increasing DI 0.3 by 0.5 points. These comparisons involve complete model variants; the following analysis isolates the effect of reasoning within fixed AutoThink checkpoints.

Confidence-Guided Reasoning. Within each AutoThink checkpoint, varying the confidence threshold from 0 (never reason) to 1 (always reason) reveals the accuracy–computation trade-off (Figure 13). Routing low-confidence predictions to reasoning captures most of the attainable accuracy gains. On held-out examples from the training distribution, the 9B model gains 1.0 accuracy point over its single-pass readout at the default threshold of 0.9, averaging 596 reasoning tokens per example; always invoking reasoning yields no further gain. At 27B, the same threshold yields a 2.5-point gain with 273 reasoning tokens per example.

The gains are more pronounced on DecisionPivot, where decisions often require explicit computation. At the default threshold, the 9B model achieves 59.9% pair accuracy with 1,365 reasoning tokens per example, compared with 22.3% when reasoning is disabled in the same checkpoint. The corresponding 27B results are 86.9% pair accuracy with 304 tokens per example, versus 48.0% without reasoning. RLCD² further improves the accuracy–efficiency trade-off at smaller scales. At 9B, it raises DecisionPivot pair accuracy from 53.3% to 59.9% while reducing reasoning-token consumption by 40%; at 4B, token consumption decreases by 25% with pair accuracy changing by less than 0.5 points. The 27B model shows little change, consistent with its short reasoning traces (Section 3.4).

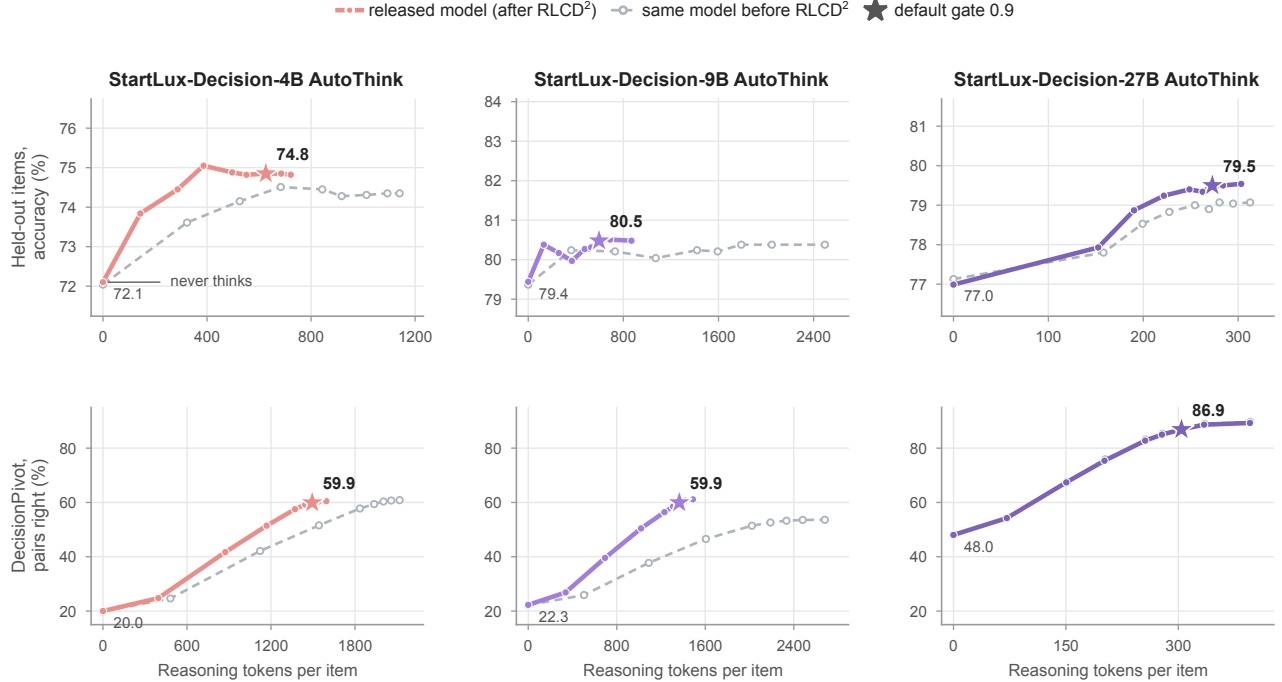


Figure 13. Confidence-guided reasoning captures most of the attainable accuracy gains, while RLCD² reduces reasoning cost at 4B and 9B. Accuracy is plotted against average reasoning tokens per example, including those answered without reasoning, as the confidence threshold varies from 0 (never reason) to 1 (always reason). Stars indicate the default threshold of 0.9. The top row evaluates 2,986 held-out examples from the training distribution; the bottom row reports pair accuracy on 7,800 DecisionPivot core examples. Each model scale uses its own token axis. Solid curves represent the released AutoThink models after RLCD², while dashed gray curves represent their pre-RLCD² checkpoints.

Combining Single-Pass and Post-Reasoning Predictions. For questions routed to reasoning, AutoThink combines initial single-pass and post-reasoning probability distributions through a weighted average. The fusion weight is particularly important for computation-intensive decisions, where initial predictions can be confidently incorrect. On DecisionPivot, the 9B model achieves 56.6% pair accuracy with equal weights, 59.9% with a post-reasoning weight of 75%, and 61.5% using only post-reasoning probabilities.

However, maximizing benchmark accuracy does not necessarily optimize probability quality. On held-out examples from the training distribution, initial predictions are already accurate, whereas post-reasoning predictions exhibit some overconfidence, consistent with prior observations (Lacombe et al., 2025). We therefore select the fusion weight by minimizing the default prediction’s Brier score on held-out data. A post-reasoning weight of 75% minimizes this score at 9B and 27B and is close to optimal at 4B, motivating the default 75%–25% combination.

Relative to equal weighting, the selected fusion improves DecisionPivot pair accuracy by 2.7 points at 4B but reduces held-out accuracy by 0.5 points at 9B and 0.3 points at 4B. DI 0.3 scores change by less than 0.15 points. At 27B, increasing the post-reasoning weight to 75% reduces held-out calibration error from 3.7 to 1.6 points, with little change in accuracy. In contrast, on Decision Index, the same weight increases calibration error by 0.8–2.0 points relative to equal weighting, according to the leaderboard’s calibration metric. These results highlight the importance of selecting fusion weights based on both probability quality and accuracy under the intended evaluation distribution.

Computational Cost and Remaining Limitations. Adaptive reasoning incurs additional generation cost for routed decisions. At the default threshold, the 4B, 9B, and 27B AutoThink models invoke reasoning on approximately 33%, 29%, and 28% of Decision Index questions, averaging 372, 271, and 110 reasoning tokens per question, respectively. These averages include questions answered directly without reasoning. Instant remains preferable for latency-sensitive applications, as it avoids autoregressive generation (Section 2.2).

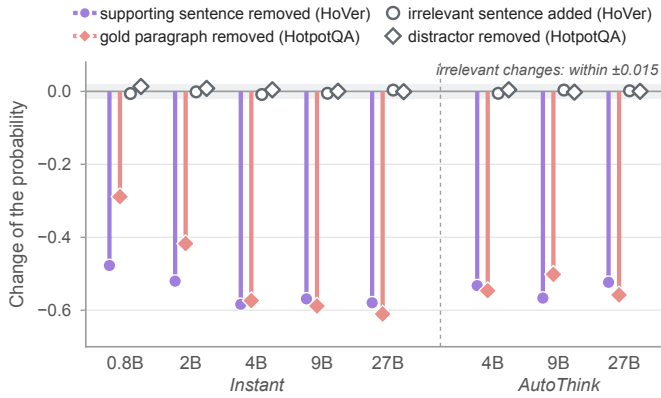
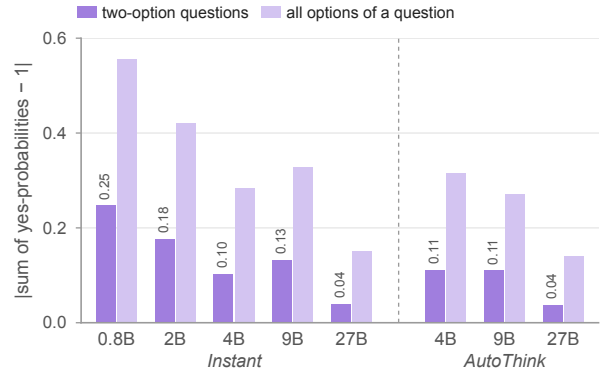
a Evidence removed against an irrelevant change**b** Each answer checked on its own

Figure 14. Decision probabilities respond to supporting evidence and become more consistent across related question formulations as model scale increases. Results use single-pass readouts of the released models. (a) Mean change in the supported answer’s probability after removing a supporting sentence from HoVer or a gold paragraph from HotpotQA. Control perturbations add an irrelevant sentence or remove a distractor paragraph, producing mean changes within ± 0.015 of zero. (b) Mean absolute deviation of the summed affirmative probabilities from one when each candidate option is independently reformulated as a yes-or-no question, reported for binary questions and for the full set of questions.

5.3 Evidence Sensitivity and Probability Consistency

We examine whether decision probabilities respond to relevant evidence and remain consistent across related question formulations. Both analyses use single-pass readouts, with reasoning disabled for AutoThink.

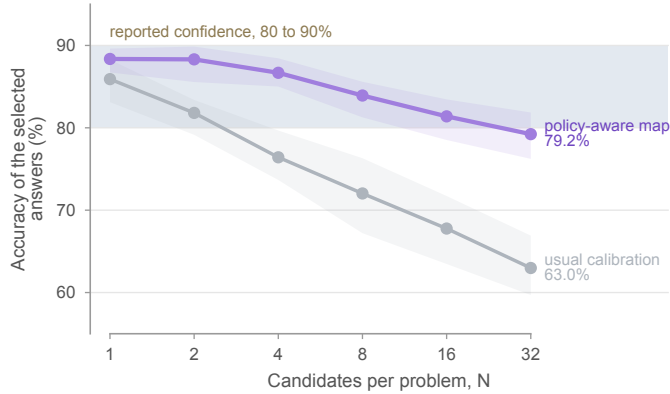
Evidence Sensitivity. We assess how predicted probabilities respond to removing supporting evidence. For HoVer claims, we remove one supporting sentence; for HotpotQA questions reformulated as binary decisions, we remove one gold supporting paragraph. Both probes use evaluation items only, and no evaluation data is used in training. Across model scales, removing supporting evidence reduces the probability assigned to the supported answer in 90–97% of cases, with mean decreases ranging from 0.29 to 0.61. By contrast, adding an irrelevant sentence or removing a distractor paragraph changes the corresponding probability by less than 0.015 on average (Figure 14a). These results indicate sensitivity to supporting evidence with limited responses to irrelevant perturbations.

Probability Consistency. We assess whether predicted probabilities remain coherent across related question formulations. For questions with mutually exclusive and collectively exhaustive options, the affirmative probabilities obtained by asking about each option separately as a yes-or-no decision should ideally sum to one. Following the consistency test applied to Jev by Li et al. (2026), we measure the mean absolute deviation of this sum from one for binary questions and extend the analysis to questions with more than two options. Consistency generally improves with model scale: the mean deviation decreases from 0.25 for the 0.8B Instant model to 0.04 for the 27B Instant model. AutoThink exhibits comparable deviations to Instant at 4B (0.11 versus 0.10) and 27B (0.04 for both), with a lower deviation at 9B (0.11 versus 0.13).

5.4 Policy-Induced Miscalibration

Well-calibrated individual predictions need not remain calibrated after downstream selection or routing, as these policies condition returned predictions on confidence and other decision signals. We examine best-of- N , think-gate, cascade, and select-then-think policies. Using StartLux-Decision-9B as a verifier reveals a substantial selection-induced calibration gap. Among individual responses assigned 80–90% confidence, 92% are correct. After selecting the highest-scoring response from 32 candidates, accuracy within the same confidence range falls to 78%. Calibrating

a Best of N with a 9B verifier



b 184 settings in four pipelines

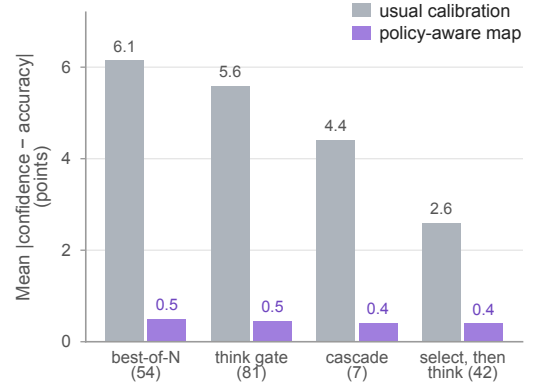


Figure 15. Selection and routing induce miscalibration that policy-aware calibration mitigates without retraining the underlying models. (a) Accuracy of responses selected by a 9B verifier from N candidates, restricted to reported confidence of 80–90%. Shaded bands indicate variation across 20 splits. (b) Mean absolute confidence–accuracy gaps across 184 settings spanning best-of- N , think-gate, cascade, and select-then-think pipelines. Policy-aware calibration reduces these gaps across all four pipelines, with signed gaps ranging from -2.7 to $+2.1$ percentage points.

individual responses does not eliminate this effect: selected-response accuracy at fixed confidence declines as the candidate pool grows (Figure 15a).

Selection changes the conditional distribution of returned predictions. Choosing the highest-scoring candidate can favor upward fluctuations in confidence estimates, an effect related to the *optimizer’s curse*. Competing candidates’ scores also convey information about problem difficulty that the winning score alone may not capture. Consequently, identical reported confidence can correspond to different empirical accuracies under different selection policies. The same concern extends to confidence-dependent routing, including AutoThink’s think-gate and cascades that escalate uncertain decisions to larger models (Gupta et al., 2024). Calibration and acceptance thresholds must therefore be evaluated on the outputs of the deployed policy. Appendix C.2 formalizes these effects.

Selection and Routing Effects. We evaluate selection-induced miscalibration across generator–verifier combinations. After conventional calibration on individual responses, selected responses are overconfident in 111 of 113 generator–verifier pairs at $N = 16$. The calibration gap increases with candidate count in all 126 evaluated runs. Confidence-based routing exhibits a related effect: for 11 of 12 models, think-gate routing is associated with lower accuracy at matched reported confidence. The direction of this shift depends on the task, as low initial confidence can identify decisions that benefit substantially from additional computation on reasoning-intensive tasks.

Selection also affects confidence-based acceptance thresholds. We certify thresholds using a fixed-sequence Clopper–Pearson test at 95% on the calibration half of each of 20 source–problem splits (Appendix C.3). Thresholds certified on individual predictions need not preserve their error bounds after selection. For best-of-16 with a target error rate of 5%, individually certified thresholds yield observed error rates above the target in 44 of 67 generator–verifier pairs. By contrast, no such violations are observed when thresholds are certified on policy-selected responses. These findings support policy-specific threshold certification.

5.5 Policy-Aware Calibration

To mitigate policy-induced miscalibration, we calibrate confidence conditional on the information used for selection and routing. We apply the deployed policy to labeled calibration examples and fit calibration maps using the returned predictions, their reported confidence, and the associated selection or routing scores. For confidence-based gates and cascades, predictions are stratified by routing-confidence quantiles, with isotonic calibration fitted within each stratum. For candidate selection, we fit a logistic model using reported confidence, mean candidate score,

Table 5. Single-pass decision readouts on an H200 take 31 ms at 0.8B and under 100 ms at 27B for 1,024-token prompts. Median latency in milliseconds for a single forward pass reading option-letter logits, measured with vLLM 0.31 on one H200. Timings include tokenization and are based on ten measurements after three warm-up passes, with one request in flight. Approximately 90% of DI requests contain fewer than 1,000 tokens. AutoThink columns report causal readouts without reasoning generation under the same conditions as Instant. BF16 baseline rows are shaded.

Precision	Prompt length (tokens)	Instant					AutoThink		
		0.8B	2B	4B	9B	27B	4B	9B	27B
BF16	256	24	24	31	33	52	29	30	49
	1,024	31	32	41	42	98	39	39	95
	4,096	35	38	72	114	358	70	109	346
	8,192	47	69	142	230	737	131	214	698
FP8	256	26	25	30	29	51	30	30	49
	1,024	33	33	39	40	72	38	38	70
	4,096	37	36	61	85	251	56	79	239
	8,192	43	59	119	172	534	105	154	489

candidate count N , and $\log N$, followed by isotonic calibration of its predictions. This procedure calibrates reported confidence without modifying the underlying models, while acceptance thresholds are certified separately for the relevant policy routes (Appendices C.4 and C.5).

Across 184 settings spanning best-of- N , think-gate, cascade, and select-then-think pipelines, policy-aware calibration reduces signed confidence–accuracy gaps from $[-11.7, +20.5]$ to $[-2.7, +2.1]$ percentage points (Figure 15b). Mean absolute gaps decrease across all four pipelines. Nevertheless, calibration remains sensitive to distribution shift: maps fitted on the held-out distribution exhibit residual overconfidence of up to 6.1 percentage points when transferred to unseen datasets.

We further examine calibration and selective prediction under the default AutoThink inference policy. On 2,986 held-out examples, default-answer calibration errors are 3.7, 2.2, and 1.6 points at 4B, 9B, and 27B, respectively, compared with post-reasoning errors of 8.1, 3.7, and 2.8 points. At 4B, post-reasoning predictions exhibit 4.0 points of overconfidence despite well-calibrated single-pass predictions, motivating route-specific threshold certification. At 9B and 27B, both inference routes exhibit signed confidence–accuracy gaps within 1.5 points. Under a target error rate of 5%, approximately half of the default predictions from the 9B and 27B models can be accepted, compared with 2% at 4B. For the 4B model, restricting acceptance to post-reasoning predictions increases coverage to 23%, reflecting errors among highly confident single-pass predictions. These findings distinguish probability calibration from selective coverage, which additionally depends on discrimination.

6. Efficient Inference and Deployment

We evaluate runtime compatibility, serving latency, local inference, and reduced-precision deployment for StartLux-Decision. Tables 5 and 8 report single-readout latency without reasoning generation. Tables 6 and 7 compare per-request latency on matched single-pass requests, using one GPU and one request in flight. For AutoThink, end-to-end latency includes reasoning generation and the post-reasoning readout when invoked.

6.1 Runtime Support

Both Instant and AutoThink support inference on vLLM, llama.cpp with GGUF, and MLX for Apple silicon. Available precisions include BF16 across all three backends, load-time FP8 weight quantization on vLLM, GGUF Q8_0 and Q4_K_M on llama.cpp, and 8-bit and 4-bit MLX builds. We implement bidirectional attention separately for each runtime. On vLLM, a plugin recomputes prompt representations in full-attention layers without the causal mask, achieving 99.5–99.8% top-choice agreement with the reference implementation. For llama.cpp, GGUF

conversion marks Instant models as non-causal, while a small server patch processes each prompt as a single unit without modifying the model computation code. On MLX, our backend implements the bidirectional prompt mask, with BF16 builds achieving 99.4–99.9% top-choice agreement with vLLM. Top-choice agreement does not imply numerical agreement between predicted probabilities.

6.2 GPU Serving Performance

Single-Readout Latency on vLLM. On a single NVIDIA H200, BF16 Instant readouts take 24–52 ms for 256-token prompts and 31–98 ms for 1,024-token prompts (Table 5). The latter setting covers most DI request lengths. FP8 achieves comparable latency for short prompts while reducing readout latency by approximately 25–30% at 8,192 tokens for the 9B and 27B Instant models. Across the 4B–27B models in BF16 and FP8, Instant readouts have approximately 2–8% higher latency than causal AutoThink readouts at 1,024 tokens and 6–14% at 8,192 tokens.

Matched Model Comparison on vLLM. Table 6 compares per-request latency with third-party decision models on 186 of 300 requests that all systems process in one forward pass. After prompt-length weighting, this subset represents approximately 91% of the DI request distribution. At 4B and 9B, Instant achieves median latency within 3 ms of the fastest comparable-scale reference, and within approximately 6 ms at 27B, while attaining substantially higher DI 0.3 scores. Mean latencies remain comparable across the evaluated 4B–27B models.

GGUF Inference on llama.cpp. We evaluate GGUF Q8_0 builds on an H200 using the same request subset and prompt-length weighting as the vLLM comparison (Table 7); reference models are converted using the llama.cpp converter. The 4B Instant model achieves the lowest median latency among comparable-scale references, while the 9B model is within 3 ms of the fastest. At 2B and 27B, median latency is approximately 5 ms and 14 ms higher than d1-3B and Kev 27B, respectively. At 0.8B, Instant matches Bosun 3.1 1.7B in median latency, with broadly comparable mean latencies across similar scales. Instant GGUF inference uses standard llama.cpp model computation with the server-side prompt handling described in Section 6.1.

6.3 Local Inference on Apple Silicon

We evaluate local inference using MLX on an Apple M5 Pro with 64 GB of unified memory (Table 8). For 1,024-token prompts, BF16 Instant readout latency ranges from 59 ms at 0.8B to approximately 1.1 s at 9B. While the 27B models exceed the available memory in BF16, 8-bit and 4-bit quantization enable their local deployment.

6.4 Quantized Inference

StartLux-Decision supports reduced-precision inference with FP8 on vLLM, GGUF Q8_0 and Q4_K_M on llama.cpp, and 8-bit and 4-bit builds on MLX. FP8 and Q8_0 largely preserve BF16 decision performance across the evaluated settings, whereas Q4_K_M can incur larger losses, reaching 4.3 points for the 27B AutoThink model on DecisionPivot (Table 9). MLX 8-bit builds achieve 99.5–99.6% top-choice agreement with the vLLM BF16 reference, compared with 96.3–97.5% for 4-bit builds.

Table 6. Instant achieves higher DI 0.3 scores with competitive latency among comparable-scale decision models. Per-request latency in milliseconds, pooled over three runs on 186 of 300 requests that all systems handle in a single forward pass. All systems receive token IDs from the same client, whereas Table 5 includes tokenization. Refused denotes the fraction of rejected requests. DI 0.3 scores come from Table 4 for StartLux-Decision and the public leaderboard for reference models. †Indicates our evaluation using the model’s own prompt and readout, as for Decision 3.0. FlyMyJev-4B has no DI score because its 26-option limit prevents complete evaluation.

System	Median	Mean	80th percentile	Refused	DI 0.3
27B					
StartLux-Decision-27B-Instant	51.9	138.9	88.9	0	72.83
Kev 27B	46.1	126.8	72.7	0	56.69
JEV-27B (AutoTrust AI Lab)	46.8	123.1	76.6	0	53.64
Decision 2.0 Vega (vLLM Semantic Router)	51.5	130.3	75.7	0	56.97
Perplexity Decider v1.1	52.4	143.4	84.2	0.08%	62.25
9B					
StartLux-Decision-9B-Instant	27.8	55.4	41.5	0	69.96
JEV-9B (AutoTrust AI Lab)	25.2	53.5	39.6	0	43.28
Decision 2.0 Lux (vLLM Semantic Router)	25.5	55.6	38.8	0	46.76
4B					
StartLux-Decision-4B-Instant	27.4	45.9	41.6	0	67.88
FlyMyJev-4B (FlyMy.AI)	24.8	43.6	38.6	0.08%	–
Decision 2.0 Nox (vLLM Semantic Router)	25.2	46.7	38.7	0	44.21
H2O-Lightning-4B (H2O.ai)	27.4	45.3	42.3	0	51.43†
2B and 3B					
StartLux-Decision-2B-Instant	24.6	35.1	36.8	0	59.14
d1-3B (Liquid AI)	16.7	26.6	24.1	0	48.99
Decision 2.0 Sol (vLLM Semantic Router)	22.8	33.3	33.1	0	30.05
0.6B to 1.7B					
StartLux-Decision-0.8B-Instant	24.8	33.7	36.6	0	53.75
Decision 2.0 Kai, 0.6B (vLLM Semantic Router)	10.2	17.7	19.6	0.08%	16.68
Bosun 3.1 1.7B (Hanno Labs)	11.8	20.8	22.6	0	20.55
Decision 2.0 Eos, 0.8B (vLLM Semantic Router)	24.5	32.6	34.1	0	20.57

Table 7. GGUF Q8_0 Instant models achieve competitive single-pass request latency on llama.cpp. Per-request latency in milliseconds on one H200 using llama.cpp b11549, pooled over three runs with the requests, weighting, and client of Table 6. The server processes one prompt in a micro-batch of up to 16,384 tokens to support bidirectional attention; this configuration increases causal-model median latency by 1–3%. For models with a separate prediction head, timings cover the backbone and exclude the head’s small matrix product. Refused denotes the fraction of rejected requests.

System	Median	Mean	80th percentile	Refused
27B				
StartLux-Decision-27B-Instant	140.7	519.4	301.9	0
Kev 27B	126.6	451.3	250.3	0
JEV-27B (AutoTrust AI Lab)	138.8	489.0	261.4	0
Decision 2.0 Vega (vLLM Semantic Router)	161.4	509.3	261.1	0
Perplexity Decider v1.1	143.6	537.8	292.8	0.08%
9B				
StartLux-Decision-9B-Instant	53.8	185.6	106.9	0
JEV-9B (AutoTrust AI Lab)	50.9	175.3	91.8	0
Decision 2.0 Lux (vLLM Semantic Router)	55.6	181.4	92.4	0
4B				
StartLux-Decision-4B-Instant	40.6	135.2	75.6	0
FlyMyJev-4B (FlyMy.AI)	42.0	139.8	69.3	0.08%
Decision 2.0 Nox (vLLM Semantic Router)	41.0	130.7	63.7	0
H2O-Lightning-4B (H2O.ai)	41.9	133.4	66.6	0
2B and 3B				
StartLux-Decision-2B-Instant	25.7	83.2	44.9	0
d1-3B (Liquid AI)	20.2	72.6	35.8	0
Decision 2.0 Sol (vLLM Semantic Router)	26.2	80.1	39.7	0
0.6B to 1.7B				
StartLux-Decision-0.8B-Instant	21.1	68.1	36.1	0
Decision 2.0 Kai, 0.6B (vLLM Semantic Router)	15.0	45.4	21.1	0.08%
Bosun 3.1 1.7B (Hanno Labs)	21.1	63.9	30.6	0
Decision 2.0 Eos, 0.8B (vLLM Semantic Router)	21.6	66.0	31.3	0

Table 8. MLX supports local BF16 and quantized inference across model sizes. Median single-readout latency in milliseconds at the specified prompt length, measured using mlx-lm on an Apple M5 Pro with 64 GB of unified memory. Measurements use three runs after warm-up, with one request in flight. A dash indicates unavailable BF16 results for the 27B models due to memory constraints. BF16 baseline rows are shaded.

Precision	Prompt length (tokens)	Instant					AutoThink		
		0.8B	2B	4B	9B	27B	4B	9B	27B
BF16	256	17	34	83	166	–	86	158	–
	1,024	59	139	522	1,087	–	587	965	–
	4,096	277	844	2,250	4,461	–	2,257	4,116	–
	8,192	752	1,942	5,085	9,836	–	4,800	8,752	–
8-bit	256	26	36	95	186	1,384	94	171	974
	1,024	66	152	602	1,124	4,294	466	1,087	3,839
	4,096	535	876	2,295	4,569	17,008	2,339	4,404	15,824
	8,192	1,080	2,002	5,815	9,850	32,002	4,875	9,053	38,069
4-bit	256	26	35	89	177	946	88	169	711
	1,024	64	144	571	1,041	3,308	527	993	3,023
	4,096	488	851	2,547	4,276	13,993	2,275	4,102	12,496
	8,192	1,067	1,992	5,571	9,374	28,697	4,730	8,435	33,465

Table 9. FP8 and GGUF Q8_0 largely preserve decision performance, while Q4_K_M can incur larger losses. Results report DI 0.3 public scores and DecisionPivot mean chance-corrected accuracy across seven scoring layers on 7,800 core examples; the latter differs from pair accuracy in Section 4.3. BF16 and FP8 use vLLM, while GGUF formats use llama.cpp b11549, so cross-format comparisons also reflect backend differences. BF16 DI 0.3 runs are separate from Table 4; a dash denotes an unavailable result.

Model	DI 0.3		DecisionPivot			
	BF16	FP8	BF16	FP8	Q8_0	Q4_K_M
Instant						
0.8B	53.81	53.24	9.5	8.7	9.7	8.7
2B	59.18	59.16	23.5	23.6	23.7	21.9
4B	67.84	67.87	47.5	46.2	46.2	43.7
9B	69.98	69.84	53.2	52.8	53.2	50.6
27B	72.89	72.91	80.5	80.5	80.7	80.2
AutoThink						
4B	68.11	68.09	65.3	65.6	65.6	63.5
9B	70.47	70.19	63.8	66.4	64.7	64.2
27B	73.52	73.20	88.6	87.3	89.5	84.3

7. Conclusion

StartLux-Decision combines efficient single-pass prediction with confidence-guided reasoning through a unified probability interface. RLCD² improves reasoning termination and efficiency through proper-score rewards and a soft length penalty, while largely preserving held-out decision accuracy. Evaluations on Decision Index and DecisionPivot demonstrate strong decision performance, while policy-aware calibration mitigates reliability distortions induced by selection and routing. Support for vLLM, llama.cpp, and MLX enables deployment across GPU and local inference environments. Future work will focus on generalization beyond represented task families and robust probability reliability under distribution shift. Together, these results advance efficient and reliable decision modeling by integrating direct prediction, selective reasoning, and calibrated confidence.

References

- Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. A survey on LLM-as-a-Judge. *arXiv preprint arXiv:2411.15594*, 2024.
- Lijuan Tang and Yuemeng Zheng. Typed decision models: An early evidence audit and evaluation checklist. *arXiv preprint arXiv:2609.32160*, 2026.
- Cloudflare. Clef. Hugging Face model card, <https://huggingface.co/Cloudflare/clef>, 2026. Accessed 11 October 2026.
- Perplexity AI. pplx-decider-v1.1-27b. Hugging Face model card, <https://huggingface.co/perplexity-ai/pplx-decider-v1.1-27b>, 2026. Accessed 11 October 2026.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022.
- Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.
- Diogo Almeida. Introducing System One models & Jev. TypeSafe AI blog, <https://typesafe.ai/blog/introducing-system-one-models-and-jev>, September 2026. Posted 15 September 2026.
- StartLux. DecisionPivot: Benchmarking Reliable Decision-Making at the Control Points of LLM Systems. Dataset <https://huggingface.co/datasets/Lie24/DecideBench>, 2026. Companion paper.
- Benchmark Heaven. JevBench: Benchmark for AI decision models. Repository <https://github.com/fstandhartinger/jevbench>; board <https://benchmarkheaven.com/jev-models>, 2026. Living benchmark; this report uses release v1.4.2 of 25 September 2026.
- Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. LLM2Vec: Large language models are secretly powerful text encoders. In *Conference on Language Modeling (COLM)*, 2024.
- Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, et al. Do NOT think that much for 2+3=? On the overthinking of long reasoning models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 9487–9499, 2025.
- Xialie Zhuang, Peixian Ma, Zhikai Jia, Zane Cao, and Shiwei Liu. Effective learning for small reasoning models: An empirical study on 0.5b reasoning LLMs. *arXiv preprint arXiv:2506.13404*, 2025.
- Zhimin Gao and Pichao Wang. OpenJev-RLCD: A working RLCD implementation, 2026. URL <https://arxiv.org/abs/2609.38850>.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645: 633–638, 2025. doi: 10.1038/s41586-025-09422-z.
- Peixian Ma, Xialie Zhuang, Chengjin Xu, Xuhui Jiang, Ran Chen, and Jian Guo. SQL-R1: Training natural language to SQL reasoning model by reinforcement learning. In *Advances in Neural Information Processing Systems 38*, 2025.

- Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.
- Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1.5: Scaling reinforcement learning with LLMs. *arXiv preprint arXiv:2501.12599*, 2025.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.
- Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.
- Tobias Deußner, Lorenz Sparrenberg, and Rafet Sifa. Evaluating and benchmarking the System One model Jev. *arXiv preprint arXiv:2609.37647*, 2026.
- Riccardo Porcedda. Jev thinks “I don’t know”, but doesn’t say it: Introducing Sys1Cal-v1 dataset for probability calibration. *arXiv preprint arXiv:2609.35342*, 2026.
- Romain Lacombe, Kerrie Wu, and Eddie Dilworth. Don’t think twice! Over-Reasoning impairs confidence calibration. *arXiv preprint arXiv:2508.15050*, 2025. ICML 2025 Workshop on Reliable and Responsible Foundation Models.
- Keyi Li, Yihao He, and Quanyi Li. Beyond calibration: Do a typed-decision model’s probabilities obey the probability axioms? *arXiv preprint arXiv:2609.33209*, 2026.
- Neha Gupta, Harikrishna Narasimhan, Wittawat Jitkrittum, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. Language model cascades: Token-level uncertainty and beyond. In *International Conference on Learning Representations*, 2024.
- Gianluca Bailo. System Switch: When should a fast decision model stop and think? a closed-loop study with typed-decision models and a vision reasoner in Doom. Code and data, <https://github.com/dexmac221/system-switch>, 2026. Not peer-reviewed; the README says “arXiv link to follow”.

Appendix A. Data Sources

This appendix documents the training mixtures, label provenance, and data integrity checks for the pipeline in Section 2.3.

A.1 Record Format

Each training record represents a typed decision about an input state, specifying candidate options, target probabilities, label provenance, and a parent-group identifier. Records derived from the same state share a parent group, enabling group-level splitting and decontamination. Overlength records are excluded rather than truncated, and a record-level hash deterministically permutes option order to reduce positional shortcuts.

A.2 Data Composition Across Training Stages

Lux-Release, used in Stage 1, combines public training splits, community decision corpora, programmatically generated examples, knowledge and commonsense multiple-choice questions, and smaller collections of tool-use, safety, and challenging decisions. Table A.1 reports its composition and label provenance, covering approximately 2.7 million records for the 0.8B–9B models and 0.8 million for 27B.

Table A.1. Composition and label provenance of Lux-Release across model scales. The 27B mixture places greater emphasis on benchmark training splits and knowledge-oriented multiple-choice questions. Label types: H = human annotation or source gold label; T = model-generated or model-assigned label; P = programmatically computed label.

Source family	Share of records		Labels
	0.8B–9B	27B	
Public training splits: classification, inference, question answering, safety, preference, and tool use	43.6%	18.6%	H, P, T
Community corpora of typed decisions	24.4%	17.2%	T, P, H
Training and development splits of 15 DI benchmarks	10.0%	26.4%	H
Knowledge and commonsense multiple-choice questions	7.1%	22.3%	H
Programmatically generated questions	5.8%	2.5%	P
Small challenging decision datasets	0.9%	1.4%	T, P
Tool use, our in-house pool, safety, and palettes	8.2%	11.5%	H, source labels
Total records	~2.7M	~0.8M	–

Lux-Rounds, used in Stage 2, contains approximately 1.7 million records primarily from benchmark training splits, related public datasets, and additional decision families. Other sources include programmatically generated and model-written examples, select-all questions, teacher-labeled decisions, a subset of *Lux-Release*, and reconstructed GSM8K training problems. *Lux-Think*, used in Stage 3, contains approximately 4,000–21,000 supervised reasoning examples per model (Section 2.3).

A.3 Benchmark-Specific Training Coverage

Table A.2 summarizes task-related Stage 2 training sources for Decision Index 0.2.1 benchmarks in the preview-model mixture. T marks the official training split of a benchmark, which is separate from its evaluation items, and no evaluation item is used in training for either the preview or the released models.

A.4 Licensing and Data Usage

The inference code is released under Apache-2.0, while model weights are distributed under CC BY-NC 4.0 for non-commercial use, with separate licensing available for commercial applications. Training sources have varying

Table A.2. Benchmark-specific training coverage in the Stage 2 mixture of the preview models. T = official training split of the benchmark or its source; G = independently generated or model-written data; R = related public data; – = no targeted Stage 2 data. This earlier mixture is not an exhaustive record of the final released checkpoints.

Benchmarks	Stage 2 sources
BANKING77, CLINC150, HoVer, ContractNLI, ANLI, VAST, NLI4CT, RAGTruth, iSarcasmEval, GSM8K, New Yorker, WinoGrande, HellaSwag, cfcolor, Amazon ESCI, ACOS, When2Call	T
API-Bank	R
BFCL	R
GPQA Diamond, MMLU-Pro	R (layout only)
Habermas Machine, ForecastBench, Humicroedit	T
appliance simulator, PhishNChips, CLadder, MuSR, ChessBench, POP909-CL, BPoMP, BRIGHT, BBH	G
SATA-Bench	G, T (upstream sources)
FinEntity	R
ToolRet, HLE, CRUXEval	–

licensing and usage conditions, including public datasets and commercially accessed model-generated examples. Model-generated examples are not redistributed.

A.5 Generated and Model-Written Data

Programmatically generated examples pair independent templates with deterministically computed labels. No evaluation items are supplied to the generators, and the procedural templates are independent of those used in DecisionPivot. Model-written examples are retained only when two verifier models independently recover the proposed label without seeing the writer’s answer.

A.6 Data Integrity Checks

Data Decontamination. We screen training records against Decision Index, public JevBench items, DecisionPivot, and other held-out sets, excluding related records by parent group to reduce training–evaluation overlap.

Shortcut Detection. We audit dataset sources for answer-position and option-format shortcuts using content-blind baselines, such as selecting a fixed position or the longest option without reading the input state. These audits identify shortcuts in the original GSM8K evaluation format, motivating the reconstructed questions in DI 0.3.

Appendix B. Training Implementation and Additional Analyses

Stage 1 and RLCD² optimize all language-model parameters, whereas Stage 2 and reasoning supervision use adapters subsequently merged into the model weights. The pretrained vision tower remains frozen throughout training (Section 2).

B.1 Bidirectional Prompt Attention

Bidirectional prompt attention is enabled by removing the causal mask from full-attention layers during single-pass prediction, while linear-attention layers retain causal recurrence. With bidirectional attention disabled, the modified implementation produces outputs bit-identical to those of the original causal implementation. Runtime integration and cross-backend validation are described in Section 6.1.

Table B.1. Targeted ablations of the StartLux-Decision training design. Results use the 4B model unless otherwise specified.

Design aspect	Alternative configuration	Observation
Direct probability optimization	Sampled policy-gradient optimization of the single-pass objective in Appendix B.4	Calibration error of 5.3 points, versus 3.9 with exact gradients
Parameter-update strategy	Train Stages 1 and 2 using a single adapter on the unmodified base model	DI 0.2.1 scores of 63.3 and 58.2, versus 64.8 with full-parameter training
Reasoning termination	Increase the generation limit fourfold for a model exhibiting repetitive reasoning loops	Termination increases from 11% to 13%, with unchanged accuracy
RLCD ² initialization	Apply policy-gradient training to a checkpoint that frequently produces empty reasoning traces	Reasoning is skipped on approximately two-thirds of decisions; 57 reasoning tokens per answer on average

B.2 Additional Training Ablations

Table B.1 summarizes additional ablations of probability optimization, parameter updates, reasoning termination, and RLCD² initialization.

B.3 Calibration and Resolution under Proper Scoring Rules

The Brier decomposition separates the expected score into calibration, resolution, and outcome uncertainty:

$$\mathbb{E} S_B = -\mathbb{E} \|p - \bar{t}(p)\|^2 + \mathbb{E} \|\bar{t}(p) - \bar{t}\|^2 - \mathbb{E} \|t - \bar{t}\|^2, \quad (7)$$

where $\bar{t}(p) = \mathbb{E}[t \mid p]$ is the conditional mean target and $\bar{t} = \mathbb{E}[t]$ is the overall mean target. The terms represent negative miscalibration, positive resolution, and negative outcome uncertainty, respectively. For a fixed evaluation distribution, outcome uncertainty is independent of the model. A higher expected score can therefore reflect improved calibration, resolution, or both, without changing the predicted option.

B.4 Direct Probability Optimization

We separately evaluate direct optimization of single-pass decision probabilities, independent of RLCD² and the training procedure for the released Instant models. Unlike sampled policy-gradient optimization, this objective permits exact differentiation with respect to model parameters. We augment the logarithmic score with decision-utility and record-level terms, regularized toward a reference model:

$$\mathcal{L} = -S_L(p_\theta, t) - \frac{1}{2} p_\theta^\top U t - \frac{1}{2} \prod_{j \in g} p_{\theta, j}^\top t_j + 0.05 \text{KL}(p_\theta \parallel p_{\text{ref}}). \quad (8)$$

Here U_{ky} denotes the utility of predicting option k when option y is correct, assigning unit utility to exact matches and allowing partial credit for ordered levels. The product extends over questions j within record g , rewarding joint correctness. The utility terms can encourage sharper predictions but are not proper scoring rules in isolation; strict propriety is therefore not guaranteed for the combined objective.

In a separate 4B single-pass experiment using the same training data as cross-entropy, this objective reduces calibration error from 5.1 to 3.9 points (approximately 23%) and Brier loss from 0.278 to 0.271. DI 0.2.1 scores remain comparable, at 52.49 for the baseline and 52.57 for the optimized objective. Sampled-answer policy gradients yield a similar DI score of 52.55 but no calibration improvement, favoring exact gradients for calibration in this setting.

The optimized model produces sharper raw predictions, with the fitted temperature increasing from 1.37 to 1.60. Since the contributions of individual objective terms are not isolated, the calibration improvement is attributed to their combined effect.

Appendix C. Policy-Aware Calibration: Proofs and Details

We formalize selection- and routing-induced calibration effects and provide evaluation protocols and threshold-certification results. Experiments use publicly available model weights, except for the external actor-to-reasoner analysis based on public system records.

C.1 Setting and Notation

Let x be an input, y its correct option, and p_0 and p_1 the single-pass and post-reasoning distributions. Define confidence $c(p) = \max_k p_k$ and prediction $a(p) = \arg \max_k p_k$. A think-gate routes an input to reasoning if $c(p_0) < t$. Marginal calibration concerns individual predictions; policy-aware calibration concerns predictions returned after selection or routing.

For best-of- N , let θ be latent problem easiness with density $h(\theta)$. Conditional on θ , the candidate correctness–score pairs (Y_i, S_i) are independent and identically distributed, with score density $f(s | \theta)$ and CDF $F(s | \theta)$. Let $m(s | \theta) = \Pr(Y_i = 1 | S_i = s, \theta)$ and $m(s) = \Pr(Y_i = 1 | S_i = s)$. Assuming continuous scores to avoid ties, the policy returns $I^* = \arg \max_i S_i$.

C.2 Theoretical Properties

Proposition 1 (Selection by the reported score). Suppose (i) $m(s | \theta)$ is non-decreasing in θ , and (ii) $F(s | \theta)$ is non-increasing in θ for each score s . Then $a_N(s) = \Pr(Y_{I^*} = 1 | S_{I^*} = s)$ is non-increasing in N , with $a_1(s) = m(s)$. Thus, a larger candidate pool cannot improve accuracy at a fixed score. If the individual-response verifier is calibrated, $m(s) = s$, selected responses are weakly overconfident; strict overconfidence requires stronger conditions.

Proof. Conditional on θ , the density of the maximum score is $Nf(s | \theta)F(s | \theta)^{N-1}$. Conditional independence gives

$$a_N(s) = \int m(s | \theta) w_N(\theta; s) d\theta, \quad w_N(\theta; s) \propto h(\theta) f(s | \theta) F(s | \theta)^{N-1}.$$

Increasing N by one multiplies the posterior weight by $F(s | \theta)$, non-increasing in θ . This downward shift in monotone likelihood-ratio order cannot increase the expectation of non-decreasing $m(s | \theta)$. ■

Using the verifier’s mean score on competing candidates as a proxy for easiness, responses scored 0.8–0.9 by the 9B verifier achieve 75.6%, 93.4%, and 98.3% accuracy in low-, medium-, and high-easiness strata. This pattern is consistent with, but does not establish, the first assumption.

Proposition 2 (Routing by an additional score). Let $G = \{c(p_0) < t\}$, $Q = c(p_1)$, and $Z = \mathbf{1}\{a(p_1) = y\}$. If $\Pr(Z = 1 | Q = q, c(p_0) = u)$ is non-decreasing in u , then

$$\Pr(Z = 1 | Q = q, G) \leq \Pr(Z = 1 | Q = q) \leq \Pr(Z = 1 | Q = q, G^c).$$

The inequalities reverse for non-increasing dependence. If p_1 is marginally calibrated and both routes have positive probability at q , calibration within the reasoning route is equivalent to $Z \perp G | Q = q$.

Proof. At fixed $Q = q$, route-specific accuracies average the same monotone function over initial confidences below or above t , giving the inequalities. Marginal calibration makes q a weighted average of both route accuracies. If one route’s accuracy equals q , the other must also equal q , which for binary Z and G is equivalent to conditional independence. ■

We diagnose residual routing information using the AUROC of initial confidence for predicting correctness within bins of post-reasoning confidence. Values above or below 0.5 indicate residual ranking direction, but AUROC 0.5 does not establish conditional independence. The analysis also applies to cascades.

Proposition 3 (Marginal certification does not imply route-wise certification). Let C be the confidence returned by a gated system, which accepts predictions when $C \geq \lambda$. Define $e = \Pr(\text{wrong} \mid C \geq \lambda)$, $e_r = \Pr(\text{wrong} \mid C \geq \lambda, r)$, and $w_r = \Pr(r \mid C \geq \lambda)$. Then

$$e = w_{\text{plain}} e_{\text{plain}} + w_{\text{thought}} e_{\text{thought}}.$$

For $w_{\text{thought}} > 0$, $e \leq \alpha$ implies only

$$e_{\text{thought}} \leq \frac{\alpha - w_{\text{plain}} e_{\text{plain}}}{w_{\text{thought}}},$$

which can exceed α when $e_{\text{plain}} < \alpha$. Conversely, certifying $e_r \leq \alpha$ on every route with positive acceptance probability guarantees the same bound $e \leq \alpha$ for the accepted predictions overall.

Proof. Apply the law of total probability conditional on acceptance. ■

Proposition 4 (Calibration conditional on policy statistics). Let T contain the information determining whether a candidate is returned, and let q be its reported confidence. If $\Pr(Z = 1 \mid q, T) = q$, where Z indicates correctness, then any return event R determined by (q, T) preserves calibration:

$$\Pr(Z = 1 \mid q, R) = \mathbb{E}[\Pr(Z = 1 \mid q, T) \mid q, R] = q.$$

Proof. The equality follows from the tower property and the measurability of R with respect to (q, T) . ■

Policy-aware maps approximate conditional correctness using candidate scores, candidate counts, or routing confidence (Section 5.5); finite-sample estimation does not guarantee exact conditional calibration.

Proposition 5 (Optimal gating under scoring and computation costs). Let S be a proper scoring rule, $g(x) \in [0, 1]$ the probability of reasoning, and $c(\tau)$ the cost of a reasoning trace. For a fixed conditional reasoning policy, define

$$J = \mathbb{E}[(1 - g(x))S(p_0(x), y) + g(x)(S(p_1(x, \tau), y) - c(\tau))].$$

Writing

$$v(x) = \mathbb{E}[S(p_1(x, \tau), y) - c(\tau) \mid x] - \mathbb{E}[S(p_0(x), y) \mid x],$$

the objective is affine in $g(x)$ and is maximized pointwise by $g^*(x) = \mathbf{1}\{v(x) > 0\}$, with either choice optimal at $v(x) = 0$. For a fixed gate and strictly proper S , the optimal post-reasoning readout given statistic $\phi(x, \tau)$ is the conditional distribution of y given ϕ under the gate-weighted distribution.

Proof. The coefficient of $g(x)$ in the conditional objective is $v(x)$, giving the optimal gate. Strict propriety yields the optimal conditional readout. ■

This characterization does not establish optimality of the released models' fixed confidence threshold.

C.3 Evaluation Protocols

Best-of- N experiments use nine verifiers (three StartLux-Decision models, two third-party decision models, and four chat models) on held-out variants of GSM8K training problems, five unseen arithmetic datasets, and reasoning tasks. Routing experiments use twelve models on diagnostic decisions spanning 39 task families, reasoning decisions, and CJ-Bench, a collection of challenging decisions in Chinese.

Signed overconfidence is mean reported confidence minus empirical accuracy, in percentage points. The route effect is the difference between routed and non-routed accuracy at matched reported confidence. Acceptance thresholds use a 95% fixed-sequence Clopper–Pearson procedure on the calibration half of each of 20 source-problem splits.

Table C.1. Policy-aware calibration reduces signed confidence–accuracy gaps across selection and routing pipelines. Overconfidence is mean reported confidence minus empirical accuracy, in percentage points. The external actor-to-reasoner row uses publicly available system records; all other results come from our own evaluations.

Pipeline	Settings	Conventional calibration	Policy-aware calibration
Best-of- N	Unseen datasets; five verifiers; gap increases with N ; $N = 2\text{--}32$; fitted at $N = 1, 4, 16$	+0.8 to +14.6	−1.2 to +0.3
Think-gate	Diagnostic decisions; nine models; reasoning rates of 10–50%	up to +16 on routed predictions	−1.5 to +2.1
Cascade	Diagnostic decisions; seven model pairs from two teams; 30% escalated	+2.3 to +8.4 on escalated predictions	−0.6 to +0.5
Select-then-think	Ten settings; fitted at the two or three smallest N	+2.1 to +7.3 at the largest N	−1.4 to +0.7 at every N
Actor → reasoner (external)	21 settings; 30% deferred	−10.7 to +18.1 on deferred predictions	−7.4 to +4.4

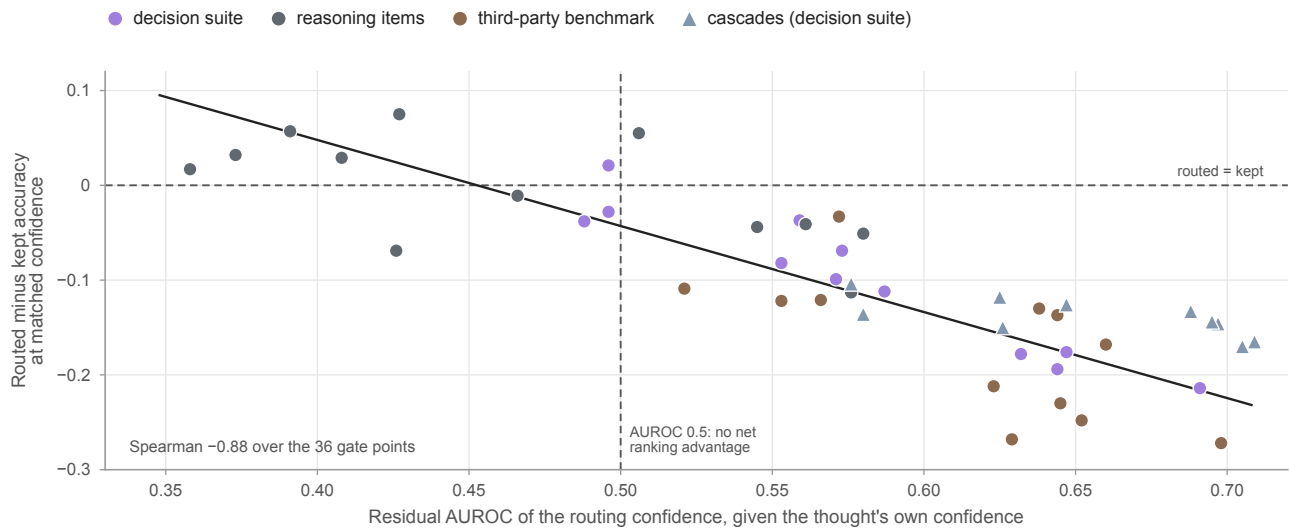


Figure C.1. Residual routing information is strongly associated with route-dependent accuracy differences. Residual AUROC measures how routing confidence ranks correctness within bins of post-reasoning confidence. The route effect is routed minus non-routed accuracy at matched confidence (Spearman -0.88 across 36 model–dataset settings; triangles mark cascades).

C.4 Results Across Decision Pipelines

Table C.1 reports calibration gaps across four decision pipelines and an external actor-to-reasoner system.

The route effect is strongly associated with residual routing-confidence information for think-gates and cascades (Figure C.1). An isotonic map of post-reasoning confidence within quantile strata of initial confidence reduces the route-specific gaps to within approximately 2 percentage points across nine models. Public records of System Switch (Bailo, 2026), where uncertain actor decisions are deferred to a reasoner, show an 8.3-point mean absolute gap between overall reasoner accuracy and accuracy on deferred items, versus 2.1 points after route-aware calibration.

C.5 Policy-Specific Threshold Certification

Table C.2 compares conventional and policy-specific threshold certification. Policy-specific thresholds reduce observed error violations, although some routes accept no predictions.

Table C.2. Policy-specific threshold certification reduces violations of target error rates among accepted predictions. Entries give accepted-prediction error rates and, in parentheses, the number of evaluation splits (out of 20) exceeding target α . Some policy-specific thresholds accept no predictions on a route.

Setting	Conventional thresholds	Policy-specific thresholds
StartLux-Decision 4B / 9B / 27B, best-of-32, unseen datasets, $\alpha = 0.05$	10.05 / 9.27 / 7.57% (19 / 20 / 20)	3.5–4.3% (0–4)
Same models; best-of-256; 0.8B generator	12.9 / 11.6 / 7.8% (20 each)	3.4 / 3.3 / 3.7% (0–1)
All 67 pairs with nonzero coverage; best-of-16, $\alpha = 0.05$	Target exceeded in 44 of 67 pairs	Target exceeded in 0 of 67 pairs
StartLux-Decision-4B, think-gate, CJ-Bench, $\alpha = 0.05$, 10% / 20% think rate	Reasoning-route error 27.9 / 27.3% (20 / 20)	No reasoning-route predictions accepted; system error 4.6% (0)
StartLux-Decision-9B, select-then-think, reasoning sets, $\alpha = 0.10$, $N = 2/4/8$	Reasoning-route error 22.2 / 18.6 / 15.8% (20 / 20 / 14)	6.9–7.8% (0)

Changes in candidate count N , routing rate, generator, or task family may require renewed calibration and threshold certification.

Appendix D. Evaluation Details

This appendix details Decision Index scoring, per-benchmark results, evaluation safeguards, and inference for long option lists and multiple questions. DecisionPivot is described by [StartLux \(2026\)](#); its data and scorer are available on [Hugging Face](#).

D.1 Decision Index Evaluation

Decision Index 0.2.1 aggregates 38 benchmarks across five weighted domains, with within-domain weights of 1.0 or 1.2. Individual scores are chance-corrected as $(s - c)/(1 - c)$ and clipped to $[0, 1]$, where s is accuracy and c is chance accuracy; unanswered requests count as incorrect. DI 0.3 retains five domains and includes 37 public benchmarks. Table D.1 reports per-benchmark results for the released models and three reference systems. Repeated evaluations with independently sampled reasoning traces vary by approximately 0.05 index points.

At 27B, AutoThink scores 11.8 points below Instant on MuSR, although invoking reasoning within AutoThink changes accuracy by only +0.5 points. In a controlled Stage 2 comparison using identical data and the same Stage 1 checkpoint (Figure 4b), causal and bidirectional models score 50.3 and 64.9 on MuSR, respectively. Three additional bidirectional seeds score 55.8–58.1, yielding gains of 5.5–14.6 points over the causal model and suggesting that attention direction contributes to the cross-variant difference. On SATA-Bench, Stage 3 and explicit reasoning account for 4.4 of the 6.7-point cross-variant difference.

Table D.1. Per-benchmark results on the public subset of DI 0.3. At least one StartLux-Decision model leads on 23 of 37 benchmarks. Scores are chance-corrected skill points (0: chance; 100: perfect accuracy). Benchmarks marked ★ have within-domain weight 1.2; others have weight 1.0. Domain weights appear in parentheses. AutoThink uses default predictions (Section 2). Reference scores are from the public leaderboard on October 7, 2026; best scores are bolded. Most tasks belong to represented training families; Table A.2 describes the earlier preview-model mixture.

Benchmark	27B		9B		4B		2B	0.8B	Reference models		
	AutoThink	Instant	AutoThink	Instant	AutoThink	Instant	Instant	Instant	Perplexity Decider v1.1 (27B)	Clef 27B	Jev (hosted)
DI 0.3	73.37	72.83	70.45	69.96	68.21	67.88	59.14	53.75	62.25	61.71	57.96
Knowledge & Reasoning (25.8%)	65.40	60.65	59.42	53.27	57.48	50.49	36.03	28.45	51.96	52.80	53.87
GPQA Diamond ★	63.3	35.4	47.6	33.3	41.5	28.6	11.6	6.8	41.5	31.3	71.4
GSM8K	96.4	80.2	93.6	64.3	94.9	63.7	41.9	24.9	72.0	52.1	66.2

Table D.1 (continued)

Benchmark	27B		9B		4B		2B	0.8B	Reference models		
	AutoThink	Instant	AutoThink	Instant	AutoThink	Instant	Instant	Instant	Perplexity Decider v1.1 (27B)	Clef 27B	Jev (hosted)
ChessBench	19.4	18.6	16.6	16.7	15.1	15.1	14.1	14.2	10.3	17.8	9.8
MuSR	49.7	61.5	46.1	53.3	46.3	51.2	44.0	35.1	53.5	74.2	46.1
SATA-Bench	52.3	59.0	53.2	57.4	53.7	57.1	53.9	50.3	27.2	32.6	25.4
CRUXEval	93.9	78.3	74.4	50.4	80.2	43.8	5.6	0.0	77.2	78.6	57.1
CLadder	89.2	88.7	85.5	85.1	80.0	83.6	67.0	58.7	58.6	88.2	45.3
HLE ★	0.0	1.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.7
MMLU-Pro ★	80.4	69.6	72.5	60.3	68.6	55.9	37.7	25.1	67.4	61.7	80.5
BBH ★	88.3	88.0	82.8	81.1	80.7	79.3	66.4	58.7	76.8	61.8	89.7
WinoGrande ★	88.0	90.2	83.4	85.3	75.4	79.2	56.0	41.3	84.8	87.1	83.9
Language Understanding (25.8%)	75.37	75.54	71.99	73.10	69.84	71.09	62.44	55.63	67.48	61.27	59.23
ContractNLI	81.3	80.6	79.6	80.4	77.9	79.0	77.2	77.4	68.7	73.0	59.1
ANLI ★	70.5	70.5	67.0	65.6	61.1	62.0	46.5	35.9	61.5	55.0	62.2
HellaSwag ★	97.0	97.4	95.1	95.5	91.3	93.1	84.4	71.2	91.0	97.6	92.7
ACOS	63.4	64.9	60.8	63.1	58.9	61.7	56.9	56.7	25.9	31.0	27.3
FinEntity	93.3	93.9	93.2	93.7	93.2	93.2	90.1	87.6	93.3	94.2	80.8
iSarcasmEval	55.0	51.0	46.6	50.4	47.7	50.5	32.9	18.9	62.7	31.0	36.3
VAST	70.8	71.7	68.0	68.6	68.6	67.6	66.2	63.9	65.1	39.0	46.9
NLI4CT	75.0	75.2	67.3	70.0	64.0	63.3	45.8	36.2	70.6	66.9	69.0
RAGTruth	68.7	71.4	66.8	67.5	63.3	66.7	60.8	53.7	65.0	57.6	51.3
Retrieval & Classification (20.0%)	73.97	74.76	73.60	74.33	71.12	73.13	69.65	66.32	61.26	62.84	55.42
BANKING77 ★	91.7	92.3	91.7	92.9	88.6	90.7	89.7	90.0	80.3	94.0	79.5
CLINC150 ★	94.3	93.8	94.0	94.2	92.7	93.5	90.5	90.2	91.5	97.4	89.2
BRIGHT ★	42.3	41.8	41.6	39.6	39.4	41.4	32.8	33.1	44.4	40.6	40.6
Amazon ESCI	53.9	56.0	51.5	53.7	51.3	52.3	50.5	48.6	50.3	46.5	43.8
PhishNChips	84.2	83.5	84.7	84.7	82.0	81.4	77.9	63.1	46.3	58.7	25.1
HoVer	76.2	80.4	76.8	80.1	71.3	78.2	75.7	70.0	48.2	31.2	45.7
Tools & Automation (18.3%)	86.75	87.71	85.83	88.14	83.98	86.44	78.62	74.61	78.88	80.71	75.09
BFCL ★	97.4	97.3	96.8	97.0	93.0	94.9	92.8	91.5	96.2	97.9	94.3
ToolRet	63.6	65.2	61.3	62.7	62.5	62.9	60.6	58.6	62.6	64.3	59.9
API-Bank ★	83.8	83.4	90.0	89.2	83.4	86.8	79.1	77.7	85.2	91.8	88.0
Home appliances	98.9	100.0	92.0	98.9	93.2	95.5	70.5	60.2	80.7	80.7	52.3
When2Call	88.6	91.7	86.0	90.9	86.2	90.4	87.2	81.0	65.1	63.3	74.6
Arts & Human Taste (10.0%)	63.14	66.20	60.55	62.98	57.07	60.11	53.65	51.00	46.96	48.88	39.08
BPoMP	95.6	96.5	93.6	94.8	91.1	92.1	81.9	80.6	90.5	94.1	81.8
Humicroedit	35.5	37.7	32.1	32.8	27.8	29.2	26.2	22.8	22.3	33.1	23.7
POP909	81.2	85.2	78.5	81.3	76.8	80.0	76.9	77.0	55.5	14.8	15.9
cfcolor	43.9	48.1	39.1	43.9	37.6	42.7	31.9	32.8	28.1	34.0	28.8
Habermas	44.9	49.2	47.0	50.0	41.4	47.0	44.4	42.2	15.8	54.5	21.5
New Yorker	77.8	80.6	73.0	75.1	67.8	69.7	60.7	50.5	69.5	62.8	62.6

D.2 GSM8K Benchmark Reconstruction

Some distractors in DI 0.2.1’s ten-choice GSM8K format provide cues to the correct answer, creating an answer-selection shortcut. DI 0.3 replaces these distractors with answers from other problems to reduce this artifact. Earlier checkpoints score 0.7–1.8 points higher on DI 0.2.1 with the original format, whereas the 4B Instant model trained only on reconstructed questions shows no such advantage.

D.3 Evaluation Reliability and Serving Configuration

Valid evaluation runs require at least 99.9% request completion; unanswered requests count as incorrect. Prefix caching and chunked prefill are disabled because they can alter decision readouts under the evaluated vLLM 0.19.1 configuration (Table D.2).

D.4 Handling Multiple Questions and Long Option Lists

Long Option Lists. A single readout supports up to 26 options, each mapped to one letter token. Longer lists use parallel groups of at most 25 options, advancing the top three from each group to a final readout. For lists of up to

Table D.2. Safeguards for reliable decision evaluation. Observed effects of vLLM 0.19.1 serving configurations relative to the reference batch runner and the corresponding safeguards. The final row shows the effect of incomplete request coverage.

Configuration	Observed issue	Measured effect	Safeguard
Prefix caching enabled	Repeated readouts of the same records differ	2.6% of readouts change by more than 0.05	Disable prefix caching
Chunked prefill with 8,192 batched tokens (default)	Readouts change when a prompt spans two scheduler steps	4B with reasoning: 65.01 instead of 66.38	Process prefill in one step
Incomplete request coverage	Unanswered requests are scored as incorrect	61.31 instead of 66.36	Require at least 99.9% completion

151 options, this procedure eliminates the correct answer before the final round in 0.2–0.7% of cases. On 151-option intent questions, the 4B model achieves 86.7% accuracy, compared with 95.7% for 26 options. Serving such lists requires multiple readouts (Section 6.2).

Multiple Questions per Document. Questions about a document are evaluated independently, so adding questions does not change existing predictions. For long documents, the evaluated serving configuration can reuse document representations when the attention pattern permits it. For a 3,200-token document with 32 questions, this increases measured throughput from 2.8 to 27 decisions per second at 27B. The gain depends on the serving configuration; bidirectional prompt attention prevents reuse of a question-independent document encoding (Section 5.1).