DISTRIBUTIONAL ENERGY-BASED MODELS FOR UNCERTAINTY-AWARE STRUCTURED LLM REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

When Large Language Models produce structured outputs such as travel plans, code solutions, or multi-step proofs, individual reasoning steps may appear correct while the output as a whole is globally invalid or non-optimal. Since valid candidates often exist in the candidate pool, reliable selection is a key bottleneck. We propose a decomposed energy function that combines a learned quality scorer with deterministic constraint penalties for verifying structured LLM outputs. The quality scorer is a heterogeneous ensemble of low-rank adapters on a single frozen encoder (3% trainable parameters); the ensemble mean ranks candidates, while its standard deviation estimates epistemic uncertainty. Across five benchmarks (GSM8K, MuSR, TravelPlanner, TACO, and Knights & Knaves), our Distributional EBM, whose quality scorer has 149M parameters, reranks a pool from 7–26B open generators and reaches 67.7% on MuSR, compared with 68.0% for Claude Sonnet 4.6, while reducing TravelPlanner constraint violations by 53% relative to Opus 4.6. The results show that structural verification and pretraining-scale priors are complementary: deterministic verification is strongest when constraints are checkable, capturing signals frontier models cannot self-detect, while frontier priors remain stronger for narrative inference and code semantics. A cross-dataset confounding analysis confirms genuine quality discrimination on four tasks and identifies a model-identity shortcut on TACO, the code-generation benchmark. Last-layer retraining reduces the dominant generator's selection share from 94% to 34% while retaining 88.6% pass@1 on TACO.

1 INTRODUCTION

Large Language Models have advanced rapidly, achieving strong performance across translation, summarisation, code generation, and open-ended dialogue (Brown et al., 2020; Achiam et al., 2023). A central driver of this progress is reasoning: the ability to decompose a problem into intermediate steps and arrive at a justified conclusion. Chain-of-thought (CoT) prompting (Wei et al., 2022; Kojima et al., 2022) unlocked this capability by eliciting step-by-step reasoning traces, and subsequent work has extended it through self-consistency (Wang et al., 2022), least-to-most decomposition (Zhou et al., 2022), and tree-structured search (Yao et al., 2023).

Yet reasoning in current models remains fundamentally limited by the autoregressive generation mechanism: each token is committed in sequence, with no global scoring function to evaluate whether the output as a whole is consistent. Local correctness at each step does not compose into global correctness for structured outputs: cross-step violations are aggregation errors that exist only at the whole-output level (a budget exceeded only when all days are summed, a return-type mismatch propagating between functions, a single arithmetic error invalidating the rest of a derivation), and per-step scoring cannot detect them by construction, no matter how accurate. Each step receives a local correctness signal yet cross-step violations remain invisible (Fig. 1, left), so the wrong plan is delivered with no warning. GPT-4-Turbo satisfies all constraints on TravelPlanner only 4.4% of the time (Xie et al., 2024), and even frontier models frequently produce code that fails on edge cases in competitive programming benchmarks (Chen et al., 2021; Li et al., 2023).

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

The bottleneck, however, is not generation but *selection*: pass@N substantially exceeds pass@1, the correct candidate is usually somewhere in the pool of N samples from the same LLM, yet existing scorers cannot reliably identify it (Brown et al., 2024; Stiennon et al., 2020). Process reward models (Lightman et al., 2024; Khalifa et al., 2025) score individual reasoning steps rather than complete outputs. Energy-based models (EBMs) offer a more natural framework for whole-output scoring, assigning low energy to good candidates and high energy to bad ones (LeCun et al., 2006), and recent work applies this principle to rerank reasoning traces (Jiang et al., 2025; Chen et al., 2025; Tan & Hong, 2025). Yet existing methods share two limitations: (i) they produce point-estimate scores with no indication of reliability (Gleave & Irving, 2022), offering no signal for when to trust, regenerate, or abstain; and (ii) they operate on flat reasoning chains with no mechanism for verifying domain-specific constraints exactly.

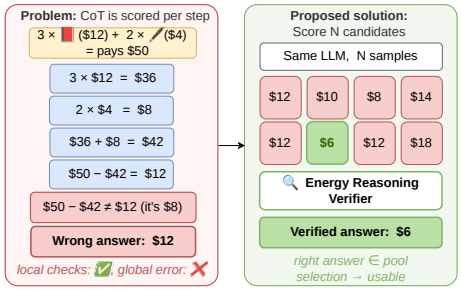


Figure 1: **Left:** CoT scores each step locally, so cross-step violations stay invisible. **Right:** Distributional energy scores all N candidates globally and checks constraints analytically.

In this paper, we show that this selection problem can be addressed with a small Distributional EBM rather than by scaling the generator. We propose a *decomposed, distributional* energy function $E = \mu_{\text{quality}} + \lambda \cdot E_{\text{constraint}}$ (Fig. 1, right) combining a learned quality term μ_{quality} with deterministic analytical constraint penalties $E_{\text{constraint}}$. The same compact architecture works across five fundamentally different reasoning tasks (math, narrative inference, multi-constraint planning, executable code, formal logic). The quality term μ_{quality} is realised as a heterogeneous ensemble of K low-rank adapters (LoRA; Hu et al., 2022) on a single frozen encoder, structurally diverse so that ensemble disagreement reflects genuine epistemic uncertainty rather than superficial variation from random seeds (Fort et al., 2019; Lakshminarayanan et al., 2017; Gal & Ghahramani, 2015). The ensemble mean ranks candidates; the standard deviation σ_{quality} provides an interpretable uncertainty signal. We also experiment with a two-pass strategy that uses uncertainty and constraint violations to trigger regeneration or abstention. Crucially, the Distributional EBM is architecturally independent of the candidate generators, so its judgement does not collapse when the generators' self-reported reasoning is wrong.

The following are the main **contributions** of this paper:

- **A novel modular energy decomposition with zero-shot transfer.** A transferable learned quality scorer E_{quality} and swappable deterministic penalties $E_{\text{constraint}}$: E_{quality} wins beyond pretraining-scale priors, $E_{\text{constraint}}$ wins on checkable constraints. The quality scorer transfers zero-shot (MuSR-trained reaches 93.9% on GSM8K without seeing math); new tasks need only new constraint functions.
- **A novel distributional energy via heterogeneous LoRA ensemble.** A parameter-efficient ensemble of structurally diverse low-rank adapters produces an energy *distribution* rather than a point estimate, enabling confounding diagnostics and σ-gated abstention.
- **Honest confounding analysis with causal verification.** On 4/5 tasks the quality scorer learns content-based quality with no model-identity shortcut (cross-model energy spreads < 0.25, picks near-uniform across generators); on TACO, a causal style-swap test exposes a formatting shortcut, fixed by last-layer retraining (Kirichenko et al., 2023), trading 3.4 pp of aggregate pass@1 for a 95% reduction in cross-model energy spread. Multi-model candidate diversity substantially improves selection.

Paper outline. Sec. 2 reviews chain-of-thought reasoning, energy-based models, reward modelling, and test-time compute. Sec. 3 presents the energy decomposition (3.1), the distributional quality scorer with formal propositions (3.2), training (3.3), and analytical constraint penalties (3.4). Sec. 4 reports the experimental setup (4.1) and results across all five benchmarks (4.2). Sec. 5 distils lessons learned and outlines future directions. The appendix provides proofs (§A), model and implementation details (§B, §C), dataset details and per-model candidate quality (§D, §E), and extended results: ablations (§F.1), energy diagnostics (§F.2), per-difficulty breakdowns (§F.3), selective prediction (§F.4), two-pass analysis (§F.5), per-constraint frontier comparison (§F.6), the full transfer matrix (§F.8), extended confounding analysis (§F.9), and broader impact (§G).

2 Related Work

Chain-of-thought prompting (Wei et al., 2022; Kojima et al., 2022) elicits step-by-step reasoning; self-consistency (Wang et al., 2022), least-to-most prompting (Zhou et al., 2022), and Tree of Thoughts (Yao et al., 2023) extend it via marginalisation, decomposition, and branching search. Process reward models give finer-grained supervision: Uesato et al. (2023) and Lightman et al. (2024) show step-level outperforms outcome-level verification, and Khalifa et al. (2025) train models that generate reasoning traces before scoring, with a fraction of process labels. All produce point-estimate scores on flat reasoning chains, with no mechanism for verifying that structured multi-section outputs satisfy global constraints jointly.

Energy-based models assign scalar energies to input-output pairs (LeCun et al., 2006) and have been applied to text generation (Deng et al., 2020); Blondel et al. (2025) prove autoregressive language models are themselves EBMs, and Carbone (2025) survey the broader landscape. For reasoning, Jiang et al. (2025) rerank CoT on GSM8K/MATH (90.7%, 55M params); Chen et al. (2025) apply energy-based calibration to implicit CoT; Tan & Hong (2025) connect RL-tuned language models to EBMs theoretically. Across this work, energy is a deterministic scalar with no reliability signal; Gleave & Irving (2022) highlight this gap for reward models but use standard ensemble variance rather than structurally heterogeneous adapters. Our pairwise-preference learning connects to RLHF (Christiano et al., 2017; Ouyang et al., 2022) and DPO (Rafailov et al., 2023) but targets post-hoc verification of complete outputs; we further produce a *distributional* energy via a heterogeneous ensemble, decompose energy into learned and analytical components, and target structured multi-section outputs. A complementary line studies test-time compute allocation. Snell et al. (2025) show verifier-guided search can outperform scaling model parameters, DeepSeek-AI (2025) show reinforcement learning alone can incentivise advanced reasoning, and Brown et al. (2024) identify candidate *selection* as the key bottleneck: pass@N exceeds pass@1 but existing methods close the gap only partially. For structured planning, Lee et al. (2025) reach 95.6% on TravelPlanner via evolutionary search with Gemini 1.5 Flash, Xia et al. (2025) use reward modelling for agent generalisation, and Wang et al. (2025) combine multiple LLMs via mixture-of-agents. We address the selection bottleneck via energy-based reranking and the verification gap via analytical constraint penalties, using ensemble uncertainty to decide when selection is reliable.

3 Methodology

We propose a Distributional EBM (Fig. 2) with two stages: a quality scorer trained on contrastive candidate pairs (top), and inference-time candidate ranking (bottom).

3.1 Problem Formulation

Let x denote an input problem and let $\{y_1, \ldots, y_N\}$ denote N candidate structured outputs generated by frozen base LLMs via temperature sampling. We propose the decomposed energy function:

$$E(x, y) = \mu_{\text{quality}}(x, y) + \lambda \cdot E_{\text{constraint}}(x, y), \tag{1}$$

and the selected candidate is the energy minimiser over the candidate pool $\mathcal{Y} = \{y_1, \ldots, y_N\}$, $y^* = \arg\min_{y \in \mathcal{Y}} E(x, y)$, where $\mu_{\text{quality}}(x, y)$ is the ensemble mean energy from the learned quality scorer, $E_{\text{constraint}}(x, y)$ is a sum of analytical penalty terms, and $\lambda > 0$ controls their relative weight. The two components serve complementary roles: E_{quality} captures general reasoning quality learned from training data, while $E_{\text{constraint}}$ encodes task-specific rules that can be verified exactly without any learning.

3.2 Distributional Quality Scorer

We instantiate the quality scorer as an ensemble of K EBMs sharing a frozen pretrained bidirectional Transformer encoder (Devlin et al., 2019; Warner et al., 2025). We choose an encoder over the now-standard causal-LM verifier so that a single `[CLS]` forward pass scores the whole `[problem]||[candidate]` pair jointly with bidirectional attention, at one to three orders of magnitude fewer parameters than CLM-based reward models or PRMs. To each member k we attach a lightweight low-rank adapter (Houlsby et al., 2019; Hu et al., 2022) that modifies a subset of the encoder's attention matrices, together with an independent energy head. Given a problem x and candidate y, the input is tokenised as input_k = `[CLS]` x `[SEP]` y `[SEP]`. The frozen encoder, augmented by adapter k, produces contextualised representations. The `[CLS]` representation $\mathbf{h}^k_{\text{CLS}}$

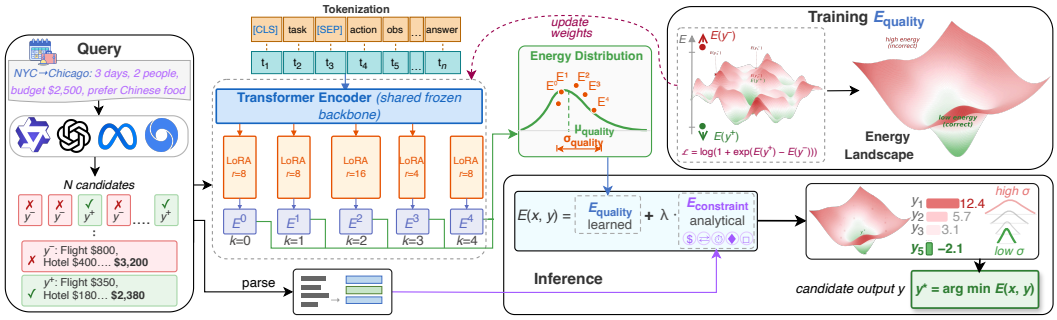


Figure 2: Method overview. **Top:** Training. Correct (y^+) / incorrect (y^-) candidate pairs pass through the frozen ModernBERT backbone with one of K=5 heterogeneous LoRA adapters; each adapter's MLP head maps `[CLS]` to a scalar E^k trained via Bradley-Terry. The K energies form a distribution: μ_{quality} ranks, σ_{quality} quantifies confidence. **Bottom:** Inference. Four LLMs generate N=32 candidates, scored in parallel by the ensemble and by analytical constraints ($E_{\text{constraint}}$), ranked by $E = \mu_{\text{quality}} + \lambda \cdot E_{\text{constraint}}$.

is passed through an MLP energy head producing a scalar $E^k_{\text{quality}}(x, y)$ (architecture in §C). Each adapter and energy head is trained independently; the frozen backbone is shared. The distributional output for candidate y is the ensemble mean $\mu_{\text{quality}}(x, y) = \frac{1}{K}\sum_{k=1}^{K} E^k_{\text{quality}}(x, y)$ and standard deviation $\sigma_{\text{quality}}(x, y) = \text{std}(\{E^1_{\text{quality}}, \ldots, E^K_{\text{quality}}\})$. The mean μ_{quality} is used for ranking (Eq. (1)); the standard deviation σ_{quality} quantifies epistemic uncertainty.

Diversity comes from three sources: *structural heterogeneity* (the K adapters differ in rank, scaling factor, and target attention modules; Tab. 6 in §C; the rationale for the specific values is given there), so disagreement reflects genuine epistemic uncertainty rather than seed variation (Fort et al., 2019; Lakshminarayanan et al., 2017); independent random initialisation per adapter; and 80% data bagging (Breiman, 1996). Trainable parameters per member (∼3% of backbone) consist only of the LoRA weights and the energy head.

Theoretical justification. A natural question is what the distributional energy (μ_{quality}, σ_{quality}) provides beyond a point-estimate scorer. We present two results that together justify the combination of decomposition and ensemble.

Proposition 3.1 (Ensemble ranking under correlated adapters)**.** *Let each adapter k correctly rank a random (correct, incorrect) pair with probability $q > \frac{1}{2}$, and let $\rho \in [0, 1]$ denote the pairwise rank correlation between adapters. Under a Gaussian approximation for $K \geq 5$, the probability that the ensemble mean $\mu_{quality}$ correctly ranks the pair, and its $K \to \infty$ limit (for any $\rho > 0$), are:*

$$P_{\text{ensemble}}(K, q, \rho) = \Phi\left(\frac{\sqrt{K}\,(q-\frac{1}{2})}{\sqrt{q(1-q)(1+(K-1)\rho)}}\right), \quad P^{\infty}(q, \rho) = \Phi\left(\frac{q-\frac{1}{2}}{\sqrt{q(1-q)\,\rho}}\right) < 1, \tag{2}$$

where Φ is the standard normal CDF and $P^{\infty} < 1$ is a finite ceiling that no ensemble size can surpass unless $\rho = 0$.

Three consequences follow (proof and detailed analysis in §A). $\partial P/\partial \rho < 0$, so structurally heterogeneous adapters (lower ρ) strictly outperform same-architecture ensembles; when $\rho = 1$, $P_{\text{ensemble}} \to q$ regardless of K; and for any $\rho > 0$ the ceiling in Eq. (2) is binding for moderate q (e.g., narrative reasoning) and slack only when q is near unity (e.g., K&K).

Proposition 3.1 justifies the ensemble for ranking. The correlation it depends on is measured rather than assumed: the inter-adapter error-correlation on decision-relevant (correct, incorrect) pairs is 0.51 to 0.71 across the five benchmarks, well below 1 (Tab. 14, §F.1). The following result justifies the *decomposition* for uncertainty: because $E_{\text{constraint}}$ is deterministic, σ_{quality} reflects purely the epistemic uncertainty of the learned quality component, uncontaminated by constraint uncertainty.

Proposition 3.2 (Partitioned variance under energy decomposition)**.** *Let $E(x, y) = \mu_{quality}(x, y) + \lambda \cdot E_{constraint}(x, y)$ where $E_{constraint}$ is a deterministic analytical function. The predictive variance of the total energy decomposes as:*

$$\text{Var}(E) = \text{Var}(E_{quality}) + \underbrace{\text{Var}(E_{constraint})}_{=0} + \underbrace{2\,\text{Cov}(E_{quality}, E_{constraint})}_{=0} = \sigma^2_{quality}. \tag{3}$$

For a monolithic scorer E_{mono} that jointly learns both components, the predictive variance is $\mathrm{Var}(E^{mono}_{quality}) + \mathrm{Var}(E^{mono}_{constraint}) + 2\,\mathrm{Cov}(E^{mono}_{quality}, E^{mono}_{constraint})$, which equals $\sigma^2_{quality}$ only in the degenerate case that $E^{mono}_{constraint}$ is learned exactly; otherwise the variance mixes constraint-prediction error into the uncertainty signal and is no longer interpretable as quality uncertainty.

This makes σ_{quality} an *interpretable* uncertainty signal: it measures quality uncertainty alone, with zero contribution from constraint verification. That is precisely what makes it safe to combine with $E_{\text{constraint}}$ as an independent second input to the triage rule (Sec. F.5.1), which accepts when σ_{quality} and $E_{\text{constraint}}$ are both low, regenerates when either is high, and abstains when σ_{quality} is extreme. We write σ-triage for brevity; the rule reads both signals. Proof and full implications are in §A.

3.3 Training

We train each member independently with the Bradley–Terry pairwise loss (Bradley & Terry, 1952) on bagged 80% subsets $\mathcal{D}^k$ of training problems, where Y_x^+ and Y_x^- are the correct/incorrect outputs for x. The objective for member k is:

$$\mathcal{L}^k = \frac{1}{|\mathcal{D}^k|} \sum_{x \in \mathcal{D}^k} \frac{1}{|P_x|} \sum_{(y^+, y^-) \in P_x} \log\Big(1 + \exp\big(E^k_{\text{quality}}(x, y^+) - E^k_{\text{quality}}(x, y^-)\big)\Big), \tag{4}$$

where $P_x \subset Y_x^+ \times Y_x^-$ is a set of contrastive pairs sampled per problem, capped at a maximum count to prevent easy problems from dominating the gradient. Members are optimised separately ($\mathcal{L}^1, \ldots, \mathcal{L}^K$); no parameters are shared across members beyond the frozen backbone. The loss pushes the energy of correct outputs below incorrect ones; the logistic form saturates naturally for well-separated pairs, avoiding the need for explicit energy regularisation.

Contrastive pairs are split at the *problem level* to prevent data leakage; labelling and hyperparameters are in Sec. 4.1. The trained ensemble is applied **zero-shot** to out-of-domain tasks; task-specific verification is handled entirely by $E_{\text{constraint}}$.

3.4 Analytical Constraint Penalties

We define $E_{\text{constraint}}$ as a sum of analytical penalty terms, each encoding a specific task constraint: $E_{\text{constraint}}(x, y) = \sum_j w_j \cdot C_j(x, y)$, where C_j is the penalty for constraint j and w_j is its weight. Each C_j returns 0 when the constraint is satisfied and a positive value proportional to the degree of violation. The constraint terms are deterministic: each is a direct computation over the parsed structured output with no learned parameters. They are also modular: adding or removing a constraint changes $E_{\text{constraint}}$ without affecting E_{quality} or requiring retraining. The specific constraint terms for each evaluation domain are defined in §E.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate across five benchmarks that all require multi-step reasoning but are verified in different ways: math (**GSM8K**; Cobbe et al., 2021), narrative inference (**MuSR**; Sprague et al., 2024), multi-constraint planning (**TravelPlanner**; Xie et al., 2024), executable code (**TACO**; Li et al., 2023), and formal logic (**K&K**; Xie et al., 2025). Prior reranking work (Jiang et al., 2025; Lee et al., 2025) has typically focused on math or planning alone; we cover all five with the same quality-scoring architecture. We additionally use **MATH-500** (Hendrycks et al., 2021) only to evaluate zero-shot transfer; it is not part of the five-benchmark headline comparison. For math, GSM8K is partially contaminated (Yang et al., 2023; Oren et al., 2024), so we rely on MuSR (released 2024, post-dating all four of our generators' pretraining) as the uncontaminated headline. TravelPlanner contributes the five analytical constraints (budget, connectivity, completeness, preferences, diversity) that define $E_{\text{constraint}}$; K&K has an exact statement-consistency constraint checker; TACO is verified by execution, running each program against every test case for its problem and stopping at the first failure. Primary metrics are pass@1 for GSM8K, MuSR, TACO and K&K, and mean violation score for TravelPlanner, the mean fraction of the five constraints violated, from 0 to 1, lower is better. More details on the datasets (train-test splits, candidate splits, etc.) are given in §E and §D.

Candidate generators. A candidate is one complete output (solution, plan or program) sampled from a generator LLM; the Distributional EBM selects among a problem's candidates. We draw candidates from four open-weight LLMs via OpenRouter: **Qwen-2.5-7B-Instruct** (Hui et al., 2024),

LLaMA-3.1-8B-Instruct (Grattafiori et al., 2024), **Gemma-4-26B-IT** (Team et al., 2026), and **GPT-OSS-20B** (Agarwal et al., 2025). Each produces 8 candidates per problem via temperature sampling (T=0.7, top-p=0.95 (Holtzman et al., 2019)), for a total of N=32 candidates per problem (4×8).

Baselines. We compare against three groups of baselines. **(1) Heuristic selection** from the candidate pool: ***Greedy*** decoding at T=0, i.e. what a user gets without any selector (as close to deterministic as floating-point non-determinism allows); ***Random***, a uniformly sampled candidate; ***Self-consistency*** (Wang et al., 2022), a majority vote on the extracted answers; and ***Oracle***, which *uses the ground-truth labels to pick the best candidate present and is therefore the pool's ceiling, not a deployable method*. **(2) State-of-the-art frontier models** (***Claude Opus 4.6*** and ***Sonnet 4.6*** (Anthropic), ***Qwen-72B-Instruct*** (Hui et al., 2024)) which generate one deterministic answer at T=0 rather than selecting from a pool.**(3) External learned selectors** scoring the same 32-candidate pool as our quality scorer (except EORM, evaluated on its own *256-candidate pool*): ***EORM*** (Jiang et al., 2025) (the closest prior energy-based method), ***Skywork-Reward-LLaMA-3.1-8B-v0.2*** (Liu et al., 2024) (an 8B reward model), ***Math-Shepherd PRM*** (Wang et al., 2024) (a 7B step-level Process Reward Model), and ***LLM-as-Judge*** (Zheng et al., 2023) using Claude Sonnet 4.6 (frontier model scoring each candidate 1-10). *Every learned baseline and frontier model in this comparison is substantially larger than our 149M-parameter quality scorer: Skywork at 8B (54×), Math-Shepherd PRM at 7B (47×), Qwen-72B (∼483×), and the frontier Claude models at a much larger scale.* The frontier single-shot LLMs are not selectors and are compared on cost in Tab. 1.

Quality-scorer backbone. The primary backbone is ModernBERT-base (Warner et al., 2025) (149M parameters, 8,192 token context); the long context accommodates MuSR narratives (3-6K characters) and full TravelPlanner itineraries without truncation. DeBERTa-v3-base (He et al., 2021) (86M parameters, 512 token context) is evaluated as an ablation; backbone comparison, adapter configurations, and DeBERTa-specific results are in §B and §F.1. The heterogeneous LoRA configuration follows Tab. 6 in §C with K=5 adapters; LoRA and energy-head dropout are both 0.2.

Training. Each quality scorer is trained per dataset with the Bradley–Terry loss in Eq. (4). Training candidates are labelled by the task's own verification: answer matching for GSM8K and MuSR, constraint-violation scoring for TravelPlanner, test execution for TACO, and exact match plus statement-consistency checking for K&K. We optimise with AdamW (Loshchilov & Hutter, 2019) and cosine annealing (Loshchilov & Hutter, 2016) at a learning rate of 5×10^{-5} (2×10^{-5} for K&K); per-dataset batch sizes and stopping criteria are in Tab. 7, and every run fits on a single NVIDIA L40S (48 GB).

Inference. The energy in Eq. (1) has one weight, λ. GSM8K, MuSR and TACO have no constraint term, so μ_{quality} alone ranks their candidates. K&K uses λ=1.0, since its exact constraint checker needs no upweighting, and TravelPlanner uses λ=2.0, chosen on a held-out validation split, because mean violation is the primary metric and heavier constraint weighting improves it. Candidates are shuffled with a fixed seed (42) before scoring to remove ordering bias.

4.2 Experimental Results

Our main experiments address the following four questions: **(i)** Is the bottleneck in structured LLM reasoning selection rather than generation? **(ii)** What do the quality scorer and constraint checker each contribute across tasks? **(iii)** Does the quality scorer learn genuine reasoning quality or exploit model-identity shortcuts? **(iv)** Does the quality scorer transfer across reasoning domains without retraining? Extended experiments in the Appendix (§F) ask how selection changes with ensemble and candidate-pool size, generator diversity, correlated adapter errors, energy separation and calibration, and task difficulty (§F.1–§F.3). We then test whether uncertainty supports abstention and whether uncertainty or constraint violations can trigger useful regeneration (§F.4–§F.5); compare constraint-level behaviour, matched constraint-checker access, and accuracy–cost trade-offs against frontier models (§F.6–§F.7); and assess cross-domain transfer and model-identity confounding (§F.8–§F.9).

(i) Selection vs. generation. Tab. 1 compares the Distributional EBM with pool heuristics, external selectors, and single-shot frontier models across all five datasets. For each dataset, we report the oracle and the strongest method from each baseline group alongside the Distributional EBM. On **GSM8K**, the Distributional EBM reaches 97.0%, matching Opus 4.6 and Sonnet 4.6 (97.0% each) while exceeding self-consistency (96.1%) and Math-Shepherd (94.2%). On **MuSR**, the Distributional EBM reaches 67.7%, improving over self-consistency (64.4%) but trailing Opus 4.6 (72.0%) and

Table 1: Selection accuracy across five benchmarks. Shaded : our 149M-parameter quality scorer reranking 32 candidates from a four-generator 7B-class pool; row labels list each baseline's parameter count (∼frontier ≈ several hundred B). Frontier models are single-shot at T=0. **Bold** = best non-oracle, underlined = second-best; figures marked $^\flat$ or * are computed on a different candidate pool or subset and are left unmarked. mean viol. = mean constraint violation (lower better, primary TravelPlanner metric). Cost is per-problem inference cost relative to the Distributional EBM (§F.7), shown because every pool method pays for 32 generations and the comparison is only fair at matched cost.

Strategy	GSM8K (pass@1 ↑)	MuSR (pass@1 ↑)	TravelPlanner‡ (mean viol. ↓, 0–1)	TACO (pass@1 ↑)	K&K (pass@1 ↑)	Cost (× Distributional EBM cost ↓)
Greedy (T=0)	91.8%	61.2%	0.098	86.4%	**100.0%**	≈0.03×
Random	92.6%	60.6%	0.231	46.6%	58.7%	1×
Self-consistency	96.1%	64.4%	0.096	36.4%	56.4%	1×
EORM (Jiang et al., 2025) (55M)♯	90.7%	–	–	–	–	≈8×
Skywork-Reward (8B) (Liu et al., 2024)	82.1%	62.3%	0.114*	70.5%	96.4%	1×
Math-Shepherd PRM (7B) (Wang et al., 2024)	94.2%	–	–	–	–	1×
LLM-as-Judge (∼frontier, 32 calls/problem) (Zheng et al., 2023)	99.2%♭	**73.5%**	0.084*	73.9%	99.9%	16–29×
Claude Opus 4.6 (∼frontier, single-shot)	**97.0%**	72.0%	0.098	**100.0%**	99.7%	≥4.2×
Claude Sonnet 4.6 (∼frontier, single-shot)	**97.0%**	68.0%	0.103	95.5%	98.7%	≥2.5×
Qwen-72B-Instruct (72B)	95.4%	54.9%	0.121	64.8%	39.0%	≥0.06×
Oracle	98.5%	86.1%	0.028	100.0%	100.0%	1×
Distributional EBM (Ours, 149M)	**97.0%**	67.7%	**0.046**	88.6%	99.9%	**1**×

Notes: ‡TP pass@1 (zero-violation) compresses methods to a 0–30% band: ours 28.0%, random 14.0%, self-consistency 12.0%. *Skywork and LLM-as-Judge TP violations are on a 20-problem subset matched to ground-truth scores (full pass@1: 21.1%, 55.0%). ♯EORM uses 256 candidates from 5 LLMs (8× our budget).

♭LLM-as-Judge GSM8K is on a more lenient pool, see §E. "–": baseline does not apply (math-only PRM/EORM).

LLM-as-Judge (73.5%). On **TravelPlanner**, the Distributional EBM obtains a mean violation of 0.046, compared with 0.098 for Opus 4.6 and 0.096 for self-consistency; LLM-as-Judge reports 0.084 on a 20-problem subset. On **TACO**, the Distributional EBM reaches 88.6%, outperforming greedy decoding (86.4%) and LLM-as-Judge (73.9%) but trailing Opus 4.6 (100.0%). On **K&K**, the Distributional EBM reaches 99.9%, matching LLM-as-Judge (99.9%), exceeding Opus 4.6 (99.7%), and falling just short of greedy decoding (100.0%). Overall, the Distributional EBM improves over the strongest pool heuristic on four of five datasets and matches or exceeds the strongest single-shot frontier model on three, with remaining gaps on MuSR and TACO. The gap to the oracle places much of the remaining error in selection rather than generation: better candidates are already present in the pool, but practical selectors do not consistently recover them. The oracle itself requires ground-truth outcomes during selection and is therefore unattainable at inference time.

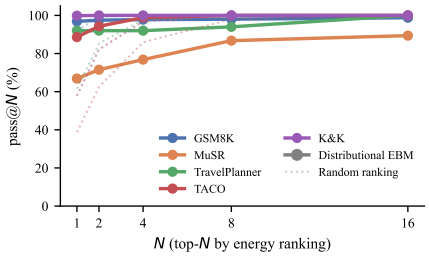


Figure 3: Pass@N (Chen et al., 2021) curves. The Distributional EBM front-loads correct candidates: pass@2 reaches 97.5% on GSM8K and 71.5% on MuSR. TravelPlanner uses a relaxed violation threshold (θ=0.15).

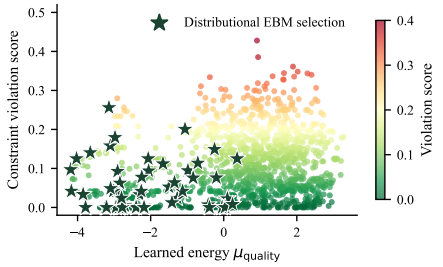


Figure 4: Energy decomposition on TravelPlanner (n=1,487 parseable candidates; both axes: lower ↓ better). Stars mark Distributional EBM selections. μ_{quality} discriminates horizontally; $E_{\text{constraint}}$ vertically. Selections cluster in the low-energy, low-violation corner.

(ii) Quality scorer vs. constraint checker. TravelPlanner provides the clearest comparison between the learned quality scorer and the constraint checker. The quality scorer alone obtains 0.058 mean violation, the constraint checker alone obtains 0.038, and the Distributional EBM obtains 0.046. The constraint checker is therefore best on the primary TravelPlanner metric. Combining the two improves upon the quality scorer alone but degrades the constraint-checker-only result. Fig. 4 shows that the components rank candidates differently, but this disagreement does not imply that their combination is uniformly better. On GSM8K, MuSR, and TACO, no constraint checker is used during selection. The Distributional EBM therefore performs reranking using the quality scorer alone. Its improvements over pool heuristics on these three tasks isolate the contribution of learned candidate ranking from that of hand-designed constraints. On K&K, the Distributional EBM combines the quality scorer with an exact constraint checker and reaches 99.9% pass@1, within 0.1 percentage points of the 100.0% pool ceiling. K&K therefore serves as a ceiling-case validation of constraint-aware selection.

Table 2: Confounding analysis; per-model pick distribution and cross-model correct-energy spread. Four datasets show near-uniform picks and spreads below 0.25. TACO is the sole outlier before DFR; after DFR its picks and spread return to the range of the others.

Dataset	Qwen	LLaMA	Gemma	GPT-OSS	Spread ↓
GSM8K	26%	24%	25%	25%	0.04
MuSR	36%	23%	18%	23%	0.18
TravelPlanner	22%	28%	26%	24%	0.24
K&K	28%	28%	27%	18%	0.11
TACO (pre-DFR)	1%	1%	**94%**	4%	**6.67**
TACO (post-DFR)	22%	24%	34%	20%	**0.34**

Table 3: Zero-shot transfer (quality-scorer pass@1). Shaded = in-domain, **bold** = best per column, underlined = second-best. TP reports zero-violation pass@1 except the K&K row ($^\diamond$), which reports mean violation.

Source	GSM8K	MuSR	TP	TACO	K&K
GSM8K	**97.0**	64.9	0.0	39.8	92.4
MuSR	93.9	**67.7**	0.0	36.4	89.0
TP	93.1	60.9	28.0	56.4	33.5
TACO	92.5	58.9	**43.3**	**88.6**	31.1
K&K	94.9	64.9	0.1$^\diamond$	51.1	**99.9**
SC	96.1	64.4	12.0	36.4	56.4
Rand	92.6	60.6	14.0	46.6	58.7

Taken together, the results support a division of labour rather than universal complementarity: the quality scorer supplies the full ranking signal on GSM8K, MuSR, and TACO, while the constraint checker drives TravelPlanner performance on the primary metric. Giving frontier models access to the same task-specific constraint checkers closes or reverses the performance gap, but at substantially higher inference cost; matched-checker and regeneration results are reported in Tabs. 22 and 24 and discussed in Secs. F.5 and F.6. The Distributional EBM's advantage therefore lies in lower-cost selection rather than exclusive access to checker feedback.

(iii) Reasoning quality vs. model-identity shortcuts. A quality scorer trained on a heterogeneous candidate pool may infer generator identity instead of assessing correctness (Geirhos et al., 2020), a failure that aggregate pass@1 does not reveal. We screen for this shortcut using two training-free diagnostics. The *pick distribution* is the fraction of test problems for which the selected candidate comes from each generator; the *cross-model correct-energy spread* is the range of generator-wise mean energies over correct candidates. A concentrated pick distribution alone may simply identify the strongest generator, but concentration together with a large energy spread indicates that equally correct candidates are treated differently by origin.

As Tab. 2 shows, TACO is the only benchmark that fails both diagnostics: the quality scorer selects Gemma on 83 of 88 problems (94%), and the correct-energy spread is 6.67. This concentration is not decisive by itself because Gemma is the strongest generator; selecting a random Gemma candidate already reaches 89.8%, compared with 92.0% for the quality scorer. We therefore intervene on style: for 40 execution-verified correct Gemma solutions, rewriting only the formatting in each other generator's style raises energy by 1.95 on average while preserving correctness. The shortcut also changes selections: on seven problems, the quality scorer chooses an incorrect Gemma solution despite a correct alternative in the pool. Further normalization and input-ablation results are reported in §F.9. Deep feature reweighting (DFR; Kirichenko et al., 2023) retrains only the five energy heads (0.4% of parameters) on group-balanced data. It reduces TACO's correct-energy spread from 6.67 to 0.34 and redistributes picks from 1/1/94/4% to 22/24/34/20%, while pass@1 changes from 92.0% to 88.6% (Tab. 2). Because the original quality scorer exceeded random selection among Gemma candidates by only 2.2 pp, the 3.4 pp decrease is consistent with removing the shortcut rather than losing substantial content-based discrimination. The other four benchmarks have near-uniform pick distributions and correct-energy spreads of 0.04–0.24. K&K shows no shortcut despite a larger generator-accuracy gap than TACO, indicating that model-specific code format, rather than generator strength alone, drives the confounding.

(iv) Cross-domain transfer. To test whether the quality scorer captures transferable reasoning signals, we evaluate each dataset's checkpoint on every other dataset zero-shot (*i.e.*, no retraining). Transfer is asymmetric and regime-dependent (Tab. 3). MuSR→GSM8K reaches 93.9% zero-shot, whereas GSM8K→MuSR reaches 64.9%, near self-consistency (64.4%). Reasoning-trained quality scorers fall below random on TACO, while TACO→TravelPlanner transfers positively; no checkpoint beats self-consistency on MATH-500 (Hendrycks et al., 2021), exposing sensitivity to task format. Overall, the quality scorer transfers within compatible reasoning regimes, but its rankings do not generalise reliably across changes in output structure or verification signal. Full transfer matrices, cross-pool results, and calibration analysis are in §F.8.

Limitations. *(i) Generator ceiling*: when no correct candidate exists, even perfect scoring cannot succeed (14% of MuSR; 58% of TravelPlanner at strict zero-violation). *(ii) Format dependence*: a GSM8K quality scorer achieves 65.6% on MATH-500 (Hendrycks et al., 2021), *below* random

(68.4%), because it learned format signals rather than reasoning quality. *(iii) Calibration limits*: on MuSR, σ-AUROC is anti-correlated (0.44), making selective prediction counterproductive. *(iv) Manual constraint design*: $E_{\text{constraint}}$ must be authored per task; scaling directions in §G. *(v) Single-round feedback*: the two-pass loop adds only one regeneration round and is therefore less powerful than iterative refinement (Lee et al., 2025; Madaan et al., 2023). Full ablations are in §F.1. Handing the exact constraint checker to a frontier model in a self-verification loop matches or exceeds the Distributional EBM's accuracy on the checkable tasks (Tab. 24), so the contribution is cost and latency rather than accuracy. σ-triage is neutral to negative on three of five tasks (§F.5), and because all adapters share one frozen encoder, a backbone blind spot can produce confident agreement on a wrong answer; a multi-backbone ensemble is the natural remedy and is left to future work.

5 Conclusion

Across the five reasoning tasks, three lessons emerge: *(i)* the Distributional EBM, whose quality scorer has 149M parameters, orchestrates cheap 7–26B open generators and outperforms the pool-matched self-consistency control on four of five benchmarks while matching frontier Claude models on constraint-heavy planning at a fraction of their cost, demonstrating that the bottleneck in structured reasoning is selection, not parameter scale; *(ii)* the two routes are complementary: structural verification wins when constraints are checkable, while pretraining-scale priors win where they are not (web-scale corpora encode narrative and code-semantic priors a small quality scorer cannot match; analytical checking exposes hallucinations frontier models cannot self-detect); *(iii)* honest confounding analysis confirms the quality scorer learns content on four of five tasks, with the lone code-formatting shortcut on TACO fixable via last-layer retraining, at a 3.4 pp aggregate cost offset by large gains on hard problems. The transferability of quality signals (a MuSR quality scorer reaching 93.9% on GSM8K zero-shot) suggests structured reasoning shares a common quality signature when training data is difficult enough to force genuine discrimination. Three directions follow: replacing the heterogeneous LoRA ensemble with epistemic deep learning for an axiomatically grounded energy in a single forward pass; extending single-round feedback to iterative energy-guided refinement; and richer structured feedback for tasks where two-pass currently fails on opaque signals (*e.g.*, code execution errors).

AI use statement

We used generative AI tools as general-purpose assistants for writing support (copy-editing and LaTeX formatting) and for code assistance in building the experimental pipeline. Generative AI was not used to generate the research idea, to design the method, or to produce any experimental result reported here: every number in this paper comes from the runs described in Sec. 4 and the appendix. Frontier language models appear in this work as objects of study, as candidate generators and as baselines, which is distinct from their use as authoring tools. We have reviewed all AI-assisted work, verified generated code against the execution-based and constraint-based checks described in §C, and we take responsibility for the final content of this paper, including all text, claims, and artefacts.

Ethics statement

This work studies verification of structured outputs from language models on five public benchmarks; no human subjects, personal data, or newly collected data are involved. The released artefact is the Distributional EBM, which performs selection rather than generation and therefore has lower direct misuse risk than a released LLM. The principal ethical consideration is automation bias: an output that passes an automated checker may receive less human scrutiny than it warrants, particularly where the uncertainty signal is poorly calibrated, as we report for MuSR (σ-AUROC 0.44). Positive and negative societal impacts, the shortcut-learning risk surfaced by our own confounding analysis, and recommended deployment mitigations are discussed in §G.

Reproducibility statement

The energy decomposition and training objective are specified in Sec. 3, with full proofs of both propositions in §A. Model details, the heterogeneous LoRA configuration, the energy-head architecture, and the per-dataset training configuration are given in §B and §C; dataset splits, labelling procedures, and per-model candidate statistics are in §D and §E. All five benchmarks are publicly available and all four candidate generators are open-weight models accessed through a public API, with sampling

parameters and the evaluation seed stated in Sec. 4.1. Anonymised code for MuSR, TACO and Knights & Knaves, covering training, inference and all analyses, is provided as supplementary material.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Anthropic. The claude 3 model family: Opus, sonnet, haiku. URL https://api.semanticscholar.org/CorpusID:268232499.

Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *ArXiv*, abs/1607.06450, 2016. URL https://api.semanticscholar.org/CorpusID:8236317.

Mathieu Blondel, Michael E. Sander, Germain Vivier-Ardisson, Tianlin Liu, and Vincent Roulet. Autoregressive language models are secretly energy-based models: Insights into the lookahead capabilities of next-token prediction. *ArXiv*, abs/2512.15605, 2025. URL https://api.semanticscholar.org/CorpusID:283920326.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Davide Carbone. Hitchhiker's guide on the relation of energy-based models with other generative models, sampling and statistical physics: a comprehensive review. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=VTgixSbrJI.

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

Zhikang Chen, Sen Cui, De-Heng Ye, Yu Zhang, Yatao Bian, and Tingting Zhu. Think consistently, reason efficiently: Energy-based calibration for implicit chain-of-thought. *ArXiv*, abs/2511.07124, 2025. URL https://api.semanticscholar.org/CorpusID:282911219.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

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

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *ArXiv*, abs/2501.12948, 2025. URL https://api.semanticscholar.org/CorpusID:284488789.

Yuntian Deng, Anton Bakhtin, Myle Ott, and Arthur Szlam. Residual energy-based models for text generation. *ArXiv*, abs/2004.11714, 2020. URL https://api.semanticscholar.org/CorpusID:212945787.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective. *ArXiv*, abs/1912.02757, 2019. URL https://api.semanticscholar.org/CorpusID:208637294.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, 2015. URL https://api.semanticscholar.org/CorpusID:160705.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard S. Zemel, Wieland Brendel, Matthias Bethge, and Felix Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2:665 – 673, 2020. URL https://api.semanticscholar.org/CorpusID:215786368.

Adam Gleave and Geoffrey Irving. Uncertainty estimation for language reward models. *ArXiv*, abs/2203.07472, 2022. URL https://api.semanticscholar.org/CorpusID:247450945.

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

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. *ArXiv*, abs/1706.04599, 2017. URL https://api.semanticscholar.org/CorpusID:28671436.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *ArXiv*, abs/2111.09543, 2021. URL https://api.semanticscholar.org/CorpusID:244346093.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv: Learning*, 2016. URL https://api.semanticscholar.org/CorpusID:125617073.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Xiaodong Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *ArXiv*, abs/2103.03874, 2021. URL https://api.semanticscholar.org/CorpusID:232134851.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *ArXiv*, abs/1904.09751, 2019. URL https://api.semanticscholar.org/CorpusID:127986954.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, 2019. URL https://api.semanticscholar.org/CorpusID:59599816.

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

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*, 2024.

Eric Hanchen Jiang, Haozheng Luo, Shengyuan Pang, Xiaomin Li, Zhenting Qi, Hengli Li, Cheng-Fu Yang, Zongyu Lin, Xinfeng Li, Hao Xu, et al. Learning to rank chain-of-thought: Using a small model. *arXiv preprint arXiv:2505.14999*, 2025.

Muhammad Khalifa, Rishabh Agarwal, Lajanugen Logeswaran, Jaekyeom Kim, Hao Peng, Moontae Lee, Honglak Lee, and Lu Wang. Process reward models that think. *Trans. Mach. Learn. Res.*, 2026, 2025. URL https://api.semanticscholar.org/CorpusID:277999741.

Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Zb6c8A-Fghk.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, Fujie Huang, et al. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Kuang-Huei Lee, Ian Fischer, Yueh-Hua Wu, Dave Marwood, Shumeet Baluja, Dale Schuurmans, and Xinyun Chen. Evolving deeper llm thinking. *arXiv preprint arXiv:2501.09891*, 2025.

Rongao Li, Jie Fu, Bo-Wen Zhang, Tao Huang, Zhihong Sun, Chen Lyu, Guang Liu, Zhi Jin, and Ge Li. Taco: Topics in algorithmic code generation dataset. *arXiv preprint arXiv:2312.14852*, 2023.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Chris Liu, Liang Zeng, Jiacai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang Liu, and Yahui Zhou. Skywork-reward: Bag of tricks for reward modeling in llms. *ArXiv*, abs/2410.18451, 2024. URL https://api.semanticscholar.org/CorpusID:273549327.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv: Learning*, 2016. URL https://api.semanticscholar.org/CorpusID:14337532.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=S37hOerQLB.

Yonatan Oren, Nicole Meister, Niladri Chatterji, Faisal Ladhak, and Tatsunori Hashimoto. Proving test set contamination in black-box language models. In *International Conference on Learning Representations*, volume 2024, pp. 16354–16372, 2024.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations*, volume 2025, pp. 10131–10165, 2025.

Zayne Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. Musr: Testing the limits of chain-of-thought with multistep soft reasoning. In *International Conference on Learning Representations*, volume 2024, pp. 14670–14728, 2024.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020.

Zhiquan Tan and Yinrong Hong. A theoretical lens for rl-tuned language models via energy-based models. *arXiv preprint arXiv:2512.18730*, 2025.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

Jonathan Uesato, Nate Kushman, Ramana Kumar, H. Francis Song, Noah Yamamoto Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process-based and outcome-based feedback, 2023. URL https://openreview.net/forum?id=MND1kmmNy0O.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Y Zou. Mixture-of-agents enhances large language model capabilities. In *International Conference on Learning Representations*, volume 2025, pp. 33944–33963, 2025.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9426–9439, 2024.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *ArXiv*, abs/2203.11171, 2022. URL https://api.semanticscholar.org/CorpusID:247595263.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, et al. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 2526–2547, 2025.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Yu Xia, Jingru Fan, Weize Chen, Siyu Yan, Xin Cong, Zhong Zhang, Yaxi Lu, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Agentrm: Enhancing agent generalization with reward modeling. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19277–19290, 2025.

Chulin Xie, Yangsibo Huang, Chiyuan Zhang, Da Yu, Xinyun Chen, Bill Yuchen Lin, Bo Li, Badih Ghazi, and Ravi Kumar. On memorization of large language models in logical reasoning. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pp. 2742–2785, 2025.

Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. Travelplanner: A benchmark for real-world planning with language agents. In *International Conference on Machine Learning*, 2024. URL https://api.semanticscholar.org/CorpusID:267406800.

Shuo Yang, Wei-Lin Chiang, Lianmin Zheng, Joseph E. Gonzalez, and Ion Stoica. Rethinking benchmark and contamination for language models with rephrased samples. *ArXiv*, abs/2311.04850, 2023. URL https://api.semanticscholar.org/CorpusID:265050721.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022.

A PROOFS OF THEORETICAL RESULTS

This section provides full proofs and the detailed consequences/implications of Proposition 3.1 and Proposition 3.2 from Sec. 3.2.

A.1 PROOF OF PROPOSITION 3.1: ENSEMBLE RANKING UNDER CORRELATED ADAPTERS

For each adapter k, define the binary correctness indicator

$$V_k = \mathbf{1}\big[E^k_{\text{quality}}(x, y^+) < E^k_{\text{quality}}(x, y^-)\big] \in \{0, 1\},$$

so that $\mathbb{E}[V_k] = q$ and $\mathrm{Var}(V_k) = q(1-q)$. The exchangeable correlation assumption is $\mathrm{Corr}(V_j, V_k) = \rho$ for all $j \neq k$, equivalently $\mathrm{Cov}(V_j, V_k) = \rho\, q(1-q)$.

Let $V = \sum_{k=1}^{K} V_k$ be the total correct-vote count. Then

$$\mathbb{E}[V] = Kq, \qquad \mathrm{Var}(V) = K\, q(1-q) + K(K-1)\, \rho\, q(1-q) = Kq(1-q)\big[1 + (K-1)\rho\big].$$

The ensemble mean $\mu_{\text{quality}}(x, y^+) < \mu_{\text{quality}}(x, y^-)$ iff $V > K/2$, i.e., a strict majority of adapters vote correctly. Under the Gaussian approximation (which the central limit theorem supports for $K \geq 5$ in the parameter range we consider),

$$V \overset{\text{approx}}{\sim} \mathcal{N}\Big(Kq,\ Kq(1-q)\big[1 + (K-1)\rho\big]\Big),$$

and standardising,

$$\begin{aligned} P_{\text{ensemble}}(K, q, \rho) = \mathbb{P}(V > K/2) &= \Phi\left(\frac{Kq - K/2}{\sqrt{Kq(1-q)(1 + (K-1)\rho)}}\right) \\ &= \Phi\left(\frac{\sqrt{K}\,(q - \frac{1}{2})}{\sqrt{q(1-q)(1 + (K-1)\rho)}}\right), \end{aligned}$$

which is Eq. (2).

For the limit, write the argument of Φ as

$$\frac{\sqrt{K}\,(q - \frac{1}{2})}{\sqrt{q(1-q)(1 + (K-1)\rho)}} = \frac{(q - \frac{1}{2})}{\sqrt{q(1-q)\big(\frac{1}{K} + \frac{K-1}{K}\rho\big)}}.$$

As $K \to \infty$, $\frac{1}{K} \to 0$ and $\frac{K-1}{K} \to 1$, so for any $\rho > 0$ the argument converges to $(q - \frac{1}{2})/\sqrt{q(1-q)\rho}$, giving Eq. (2):

$$P^{\infty}(q, \rho) = \Phi\left(\frac{q - \frac{1}{2}}{\sqrt{q(1-q)\rho}}\right) < 1 \quad \text{for any finite } q \in (\tfrac{1}{2}, 1).$$

The ceiling is strictly below 1 because the argument of Φ is finite. Only when $\rho = 0$ (perfectly independent adapters) does $\sqrt{K}$ in the numerator dominate, sending $P_{\text{ensemble}} \to 1$.

Detailed consequences. Three consequences follow.

(C1) **Diversity amplification.** $\partial P_{\text{ensemble}}/\partial \rho < 0$ for $q > \frac{1}{2}$: lower inter-adapter correlation strictly improves ensemble accuracy. This is non-obvious because one might expect that highly correlated (i.e., individually strong) adapters would produce a stronger ensemble; the proof shows the opposite.

(C2) **Diminishing returns of K at high correlation.** When $\rho = 1$, all adapters are perfectly correlated and $P_{\text{ensemble}} \to q$ regardless of K: adding more identical adapters provides zero benefit. When $\rho < 1$, ensemble accuracy increases with K but saturates at the ceiling $P^{\infty}(q, \rho)$.

(C3) **Ceiling binding regime.** Empirically, at $K{=}5$, the gap $P^{\infty} - P_{\text{ensemble}}(K{=}5)$ is small in our regime (less than 1 pp for typical (q, ρ) pairs we observe), so additional adapters yield strongly diminishing returns once $K \gtrsim 5$. The ceiling is binding for tasks where per-adapter accuracy q is moderate (e.g., narrative reasoning where $q \approx 0.6$) and slack only for tasks where q is near unity (e.g., K&K with high adapter accuracy).

The implication is that ensemble diversity is necessary but not sufficient: a fundamental ceiling is determined by the joint (q, ρ) of the adapter architecture and training distribution.

A.2 Proof of Proposition 3.2: partitioned variance under energy decomposition

We have $E(x, y) = \mu_{\text{quality}}(x, y) + \lambda \cdot E_{\text{constraint}}(x, y)$ where $E_{\text{constraint}}$ is a deterministic analytical function of (x, y) with no learnable parameters and no stochastic components. Treating μ_{quality} as a random variable (over the ensemble's epistemic uncertainty), the variance of a sum of a random and a deterministic quantity is:

$$\begin{aligned}\text{Var}(E) &= \text{Var}(\mu_{\text{quality}} + \lambda\, E_{\text{constraint}}) \\ &= \text{Var}(\mu_{\text{quality}}) + \lambda^2 \underbrace{\text{Var}(E_{\text{constraint}})}_{=0} + 2\lambda \underbrace{\text{Cov}(\mu_{\text{quality}}, E_{\text{constraint}})}_{=0} = \sigma^2_{\text{quality}},\end{aligned}$$

since the variance of any deterministic function is 0 and the covariance of any random variable with a deterministic quantity is 0. This gives Eq. (3).

For the monolithic comparison, suppose a single scorer E_{mono} is trained jointly to predict both quality and constraint satisfaction, $E_{\text{mono}}(x, y) = E^{\text{mono}}_{\text{quality}}(x, y) + E^{\text{mono}}_{\text{constraint}}(x, y)$, where now both components are random variables under the scorer's epistemic uncertainty. Then

$$\text{Var}(E_{\text{mono}}) = \text{Var}(E^{\text{mono}}_{\text{quality}}) + \text{Var}(E^{\text{mono}}_{\text{constraint}}) + 2\,\text{Cov}(E^{\text{mono}}_{\text{quality}}, E^{\text{mono}}_{\text{constraint}}).$$

This equals $\sigma^2_{\text{quality}}$ only in the degenerate case where the monolithic scorer learns $E^{\text{mono}}_{\text{constraint}}$ exactly (so $\text{Var}(E^{\text{mono}}_{\text{constraint}}) = 0$ and the covariance term vanishes); otherwise the variance mixes constraint-prediction error into the uncertainty signal.

Detailed implications. Three implications follow.

(I1) **Interpretable uncertainty.** σ_{quality} measures exactly how uncertain the ensemble is about reasoning quality, with zero contribution from constraint verification. A monolithic scorer's uncertainty conflates both sources and cannot distinguish "uncertain about quality" from "uncertain about constraints."

(I2) **Strictly lower predictive variance.** The decomposition yields strictly lower predictive variance than a monolithic scorer whenever the analytical constraints have lower error than a learned approximation. This holds by construction for TravelPlanner (database-verified routes) and K&K (statement-consistency checker), where $E_{\text{constraint}}$ is exact.

(I3) **Natural decision rule.** The σ-triage mechanism (Sec. F.5.1) is the natural decision rule under this partitioned variance:

- *Accept* when both σ_{quality} and $E_{\text{constraint}}$ are low (quality reliable, constraints satisfied).
- *Regenerate with constraint feedback* when $E_{\text{constraint}}$ is high (constraints violated, regardless of σ_{quality}).
- *Regenerate with quality feedback* when σ_{quality} is high but $E_{\text{constraint}}$ is low (constraints fine, quality uncertain).
- *Abstain* when σ_{quality} exceeds a threshold (quality completely unknown).

B Model Details

This section details the candidate generators (Tab. 4), quality-scorer backbones (Tab. 5), ensemble architecture (Tab. 6), and training configuration (Tab. 7).

B.1 Candidate Generators

Gemma-4-26B-IT consistently produces the highest candidate accuracy across all four benchmarks (96.7% on GSM8K, 66.7% on MuSR, 67.9% on TravelPlanner, 84.6% on TACO; see Tab. 9). LLaMA-3.1-8B-Instruct produces the lowest accuracy but contributes the most diverse error patterns: on GSM8K, it provides 1,309 unique negative problems, 4-10$\times$ more than the other generators. GPT-OSS-20B is a reasoning model that outputs extended thinking traces before the final answer; we set `max_tokens=4096` to capture both the reasoning and the response.

Table 4: Candidate generator models. All models are frozen throughout and accessed via the OpenRouter API. Active parameters are listed for mixture-of-experts models.

Model	Architecture	Parameters	Context	Role
Qwen-2.5-7B-Instruct (Hui et al., 2024)	Dense Transformer	7B	128K	7B baseline; pass-2 generator
LLaMA-3.1-8B-Instruct (Grattafiori et al., 2024)	Dense Transformer	8B	128K	8B baseline; highest error diversity
Gemma-4-26B-IT (Team et al., 2026)	Mixture-of-Experts	26B (4B active)	128K	Strongest generator across all tasks
GPT-OSS-20B (Agarwal et al., 2025)	Mixture-of-Experts	20B (3.6B active)	128K	Reasoning model (extended thinking)

B.2 Frontier Models

Three frontier models are evaluated as single-shot baselines on TravelPlanner and TACO: Claude Opus 4.6 and Claude Sonnet 4.6 (Anthropic), and Qwen-72B (Hui et al., 2024). All generate a single deterministic output at $T{=}0$ with `max_tokens=2048`. Opus 4.6 also serves as the primary LLM judge for the TravelPlanner quality validation (Sec. 4.2).

B.3 Quality-Scorer Backbones

Table 5: Quality-scorer backbone comparison.

Backbone	Parameters	Max context	Trainable / backbone
ModernBERT-base (Warner et al., 2025)	149M	8,192	$\sim$3%
DeBERTa-v3-base (He et al., 2021)	86M	512	$\sim$3%

ModernBERT-base is the primary backbone for all experiments. Its 8,192-token context accommodates MuSR narratives (3,000-6,000 characters) and TravelPlanner itineraries without truncation. DeBERTa-v3-base is evaluated as an ablation; its 512-token context truncates long inputs but produces better-calibrated uncertainty estimates (ECE 0.10 vs. 0.32 on GSM8K) and stronger zero-shot transfer (96.1% on GSM8K from MuSR training vs. 93.9% for ModernBERT).

C Implementation Details

Energy head architecture. Each adapter k has an independent energy head: $E^k_{\text{quality}}(x, y) = \text{MLP}_k\big(\text{LayerNorm}(\mathbf{h}^k_{\text{CLS}})\big)$, where MLP_k consists of Linear($d{\to}d$), GELU (Hendrycks & Gimpel, 2016), Dropout(p), Linear($d{\to}1$), preceded by LayerNorm (Ba et al., 2016).

Table 6: Heterogeneous LoRA ensemble configuration. All adapters share a single frozen encoder backbone. Structural diversity in rank, scaling, and target modules forces disagreement that reflects epistemic uncertainty. ModernBERT targets are shown; DeBERTa uses query, key, value, and output projections analogously.

Adapter	Rank	Alpha	Dropout	Target Modules (ModernBERT)
0	8	16	0.2	Wqkv
1	8	16	0.2	Wqkv, Wo
2	16	32	0.2	Wqkv
3	4	8	0.2	Wqkv
4	8	16	0.2	Wqkv, Wi

Configuration rationale. The three axes of Tab. 6 each spread the members along a different dimension, so that no two learn the same function, which is what lowers the inter-adapter correlation that Proposition 3.1 relies on. Rank $\{4, 8, 16\}$ spans a small-to-moderate range of capacities; powers of two are the conventional LoRA setting and keep every member lightweight (3% trainable parameters in total). Scaling factor $\{8, 16, 32\}$ varies the effective update size and the scaling-to-rank ratio, so members move off the frozen features by different amounts. Target modules {Wqkv; Wqkv+Wo; Wqkv+Wi} vary where the adapter acts: attention projections alone, plus the attention output projection, plus the first feed-forward layer. The values are set by one principle rather than per benchmark: within a small parameter budget, make the members as different as possible. All sit in standard LoRA ranges, and that the choice produces diverse members is measured, not assumed (Tab. 14). For a new domain the recipe is to span each axis with a few standard values and confirm the members disagree, rather than to search for specific numbers.

Table 7: Training configuration per dataset. All use AdamW with weight decay 0.01, cosine annealing, 10% warmup, gradient clipping at L_2 norm 1.0, and per-adapter bagging (80% of problems).

Dataset	Train problems	Candidates	Usable problems	Pairs	Batch size	Max length	Early stop patience
GSM8K	7,473	32 (multi)	2,464	39,424	32	8,192	2
MuSR	605	32 (multi)	544	8,704	8 (MB) / 16 (DB)	4,096 (MB) / 512 (DB)	2
TravelPlanner	175	32 (multi)	175	2,800	32	8,192	2
TACO	601	32 (multi)	559	7,096	32	5,120	2
Knights & Knaves	6,200	32 (multi)	4,426	70,816	16	1,536	3

Energy head architecture. Each adapter's energy head maps the `[CLS]` representation to a scalar energy: LayerNorm$(d) \to$ Linear$(d, d) \to$ GELU $\to$ Dropout$(0.2) \to$ Linear$(d, 1)$, where d=768 for both ModernBERT-base and DeBERTa-v3-base. Trainable parameters per adapter: LoRA weights ($\sim$0.3-1.2M depending on rank and target modules) plus energy head ($\sim$0.6M), totalling $\sim$3% of backbone parameters per member. The full K=5 ensemble adds $\sim$15% trainable overhead relative to the frozen backbone.

DeBERTa adapter configuration. DeBERTa-v3-base uses disentangled attention projections instead of fused QKV. The heterogeneous adapters target: adapter 0: `query_proj, value_proj`; adapter 1: `query_proj, value_proj, key_proj`; adapter 2: `query_proj, value_proj` (r=16); adapter 3: `query_proj, value_proj` (r=4); adapter 4: `query_proj, value_proj, output_proj`.

Training dynamics. On GSM8K (ModernBERT multi-model), all five adapters converge within 3-4 epochs, with best validation losses of 0.055-0.058. On MuSR (ModernBERT multi-model), adapters train for 3-4 epochs with best validation losses of 0.64-0.65 (well below the random baseline of $\log 2 \approx 0.693$, confirming that a learnable signal is present despite the difficulty of the task). DeBERTa adapters consistently train for more epochs (5 on MuSR, 2-5 on GSM8K) than ModernBERT, reflecting more stable optimisation at the shorter 512-token context. All training uses fp16 automatic mixed precision; every run fits on a single NVIDIA L40S (48 GB), with peak memory usage $\sim$31 GB at batch size 32. No run is distributed across GPUs: each adapter is trained as an independent single-GPU job, and the three L40S instances available for this work were used to execute such jobs in parallel rather than to shard a single run. Wall-clock time per adapter is $\sim$20 minutes per epoch on GSM8K and $\sim$5 minutes per epoch on MuSR.

D Training Data Details

D.1 GSM8K

Multi-model candidates: 7,473 problems $\times$ 32 candidates (8 per generator). Classification by answer matching yields an overall candidate accuracy of 95% (dominated by contamination). Of 7,473 problems, 2,464 have both correct and incorrect candidates across the four generators (usable for contrastive training); 5,009 have all-correct pools (excluded). LLaMA-3.1-8B contributes 1,309 unique negative problems (91.4% accuracy), providing 4-10$\times$ more negatives than the other generators. Contrastive pairs are capped at 16 per problem, yielding 39,424 total pairs split 80/20 at the problem level.

D.2 MuSR

756 problems split 80/20: 605 train, 151 test. Multi-model candidates: 32 per problem (8 per generator, except Qwen which contributes 16 on some subsets). Average candidate accuracy: 57%. Of 605 train problems, 544 (90%) have both correct and incorrect candidates, providing 3$\times$ better coverage than GSM8K despite being 12$\times$ smaller. This coverage advantage is the primary reason MuSR produces stronger training signal per problem.

D.3 TRAVELPLANNER

175 train, 50 test problems from the validation split (sole-planning mode). Multi-model candidates: 32 per problem. Classification uses a continuous violation score in $[0, 1]$ aggregating 8 normalised constraint dimensions. Positives are the top 25% (8 of 32) per problem; negatives are the bottom 75% (24 of 32). All 175 train problems have both positives and negatives by construction. Gemma-4-26B dominates with 67.9% of its candidates in the positive set; LLaMA achieves only 8.6%.

D.4 TACO

695 fully-clean problems from `likaixin/TACO-verified`, filtered to drop `VERY_HARD` (0% solve rate). 601 train, 94 test (88 test problems usable after filtering all-correct and all-incorrect pools). Classification by automated test execution: candidates pass if every test case returns correct output within a 4-second timeout. Positives capped at 8 per problem; negatives capped at 24, ranked by status severity. Overall candidate accuracy: 31.9% (train), with Gemma-4-26B at 84.6% producing 58% of all positives. A confounding analysis (§F.9) confirmed this imbalance caused a model-identity shortcut; DFR (Kirichenko et al., 2023) retraining on group-balanced data mitigates it.

E DATASET DETAILS

E.1 EXPLORATORY DATA ANALYSIS

We characterise the four benchmarks along three axes: scale (split sizes and candidate counts), length (problem and candidate token counts under the ModernBERT tokenizer), and difficulty (a dataset-specific proxy: number of reasoning steps for GSM8K, MuSR subset and number of choices, trip duration in days for TravelPlanner, and official difficulty label for TACO).

Table 8: Dataset overview. Candidate accuracy is the fraction of correct candidates across all generators.

Dataset	Train	Test	N	Cand. accuracy	Label type	$E_{\text{constraint}}$	Primary metric
GSM8K	7,473	1,319	32	95%	Binary (answer match)	None	pass@1
MuSR	605	151	32	57%	Binary (answer match)	None	pass@1
TravelPlanner	175	50	32	Top-25%	Continuous (viol. score)	5 constraints	Mean violation
TACO	601	88	32	32%	Binary (execution)	None	pass@1
Knights & Knaves	6,200	700	32	58%	Binary (exact match)	Statement consistency	pass@1

E.2 PER-MODEL CANDIDATE QUALITY

Table 9: Per-model candidate accuracy on the training set. Gemma-4-26B consistently dominates; LLaMA-3.1-8B contributes the most diverse errors.

Model	GSM8K	MuSR	TravelPlanner	TACO	K&K
Gemma-4-26B-IT	96.7%	66.7%	67.9%	84.6%	99.9%
GPT-OSS-20B	96.1%	65.3%	9.8%	25.8%	99.0%
Qwen-2.5-7B	95.1%	51.4%	13.6%	16.9%	19.7%
LLaMA-3.1-8B	91.4%	49.7%	8.6%	10.5%	11.7%

On GSM8K, the high candidate accuracy (91-97%) across all models reflects likely contamination. On MuSR, the gap between Gemma (67%) and Qwen/LLaMA ($\sim$50%) is substantial, providing the diversity of error patterns that drives contrastive learning. On TACO, Gemma produces 58% of all positive candidates. The initial quality scorer selected Gemma in 98% of test problems; after DFR deconfounding, picks are distributed uniformly (22/24/34/20%).

E.3 GSM8K

GSM8K (Cobbe et al., 2021) comprises 8,792 grade-school math word problems (7,473 train, 1,319 test). Each problem is a short narrative ending with a question; the solution is a chain-of-thought derivation concluding with a numeric answer after the #### delimiter. On GSM8K, $E_{\text{constraint}}$ does not apply (no structured constraints beyond answer correctness), so the Distributional EBM uses only the quality scorer for reranking.

The primary metric is *pass@1 after reranking*. Secondary metrics include pass@N (Chen et al., 2021) curves, energy separation (mean energy gap between correct and incorrect candidates), Kendall τ rank correlation, σ-AUROC, and expected calibration error (ECE; Guo et al., 2017).

Contamination. GSM8K was released in 2021 and is widely included in LLM pretraining (Yang et al., 2023). Random candidate accuracy of 92.6% and self-consistency at 96.1% are both implausibly high for grade-school math, suggesting that generators are reproducing memorised answers rather than reasoning. We report GSM8K for completeness and backward comparability, but emphasise MuSR (released 2024, post-dating all generators' pretraining) as the clean signal.

E.4 MuSR

MuSR (Sprague et al., 2024) (Multi-step Soft Reasoning) consists of 756 narrative reasoning problems. Three subsets test distinct reasoning capacities: *murder mysteries* (250 problems, 2 choices) require identifying a perpetrator from a narrative with multiple suspects; *object placements* (256 problems, 3 choices) require tracking object locations across a multi-paragraph narrative; *team allocation* (250 problems, 3 choices) require matching people to roles based on described traits. Narratives span 3,000-6,000 characters and require multi-hop inference over entities and attributes.

$E_{\text{constraint}}$ does not apply; the Distributional EBM uses only the quality scorer for reranking. The primary metric is pass@1.

E.5 TravelPlanner

TravelPlanner (Xie et al., 2024) is a multi-constraint planning benchmark. Given a natural-language query specifying a trip (origin, destinations, dates, budget, preferences), the model produces a day-by-day JSON itinerary. All experiments use *sole-planning mode*: human-curated reference information is provided directly to the LLM.

We define five analytical constraint terms following the original benchmark categories (Xie et al., 2024), each verified against the sandbox database:

- *Budget*: $C_{\text{budget}} = \text{excess}/\text{budget}$, normalised to $[0, 1]$.
- *Connectivity*: fraction of inter-city legs with invalid transport routes.
- *Completeness*: fraction of missing required fields (accommodation, meals, attractions per day).
- *Preferences*: fraction of unsatisfied user preferences.
- *Diversity*: fraction of repeated entities across days.

Additionally, *hallucination count* (entities not in the sandbox), *structure violations* (route reasonableness), and *parse success* are tracked. The violation score is the mean of all 8 normalised dimensions. The primary metric is mean violation score; secondary metric is pass@1 at θ=0 (zero violations).

LLM-as-Judge candidate pool (GSM8K). LLM-as-Judge in Tab. 1 is evaluated on a slightly more lenient GSM8K candidate pool than the rest of the comparison: random pass@1 is 92.0% (vs. 92.6% on our pool) and oracle@32 is 99.2% (vs. 98.5%). The judge's headline 99.2% essentially matches its pool's oracle, so the row reflects pool ceiling rather than a clean head-to-head with our quality scorer.

Table 10: Qualitative examples from each benchmark. Each row shows a truncated problem, a candidate excerpt, and the verification outcome. Correct candidates receive low energy; incorrect candidates receive high energy.

Dataset	Problem (truncated)	Candidate excerpt	Verdict
GSM8K	Janet's ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins for her friends every day with four. She sells every remaining duck egg at the farmers' market for $2 each. How much does she make every day?	Janet has 16 eggs per day. She eats 3 and bakes 4, so she uses $3 + 4 = 7$ eggs. She has $16 - 7 = 9$ eggs left. She sells them for $2 each: $9 \times 2 = 18$. `#### 18`	Correct E=−2.95
MuSR	A murder mystery narrative (3,200 chars): three suspects at a dinner party. Motive, alibi, and physical evidence described across 6 paragraphs. "Who is the most likely murderer?" Choices: (A) Alice, (B) Bob, (C) Carol.	Let me trace the evidence step by step. Alice has an alibi confirmed by two witnesses... Bob was seen near the kitchen... The knife found matches Bob's set... `#### 2`	Correct (B) E=−0.31
TravelPlanner	Please provide a travel plan departing from Las Vegas to Denver for 3 days, March 12-14, 2022. Budget: $1,600. Solo trip.	`[{"days": 1, "current_city": "from Las Vegas to Denver", "transportation": "Flight F3877057", "lunch": "The Urban Socialite, Denver", "dinner": "Kloof Street House, Denver", "accommodation": "Luxury Studio Suite..."}, ...]`	0 violations viol. score: 0.00
TACO	A flower shop has n bouquets with a_i flowers each. Make the maximum number of large bouquets (two or more initial bouquets, odd total). Each initial bouquet used at most once.	`def solve(): n = int(input()); a = list(map(int, input().split())); odd = sum(1 for x in a if x % 2); even = n - odd; print(min(odd, even) + max(0, (odd - even) // 2) if odd > even else odd)`	Wrong answer test 1 failed
K&K	Oliver says "Ethan is a knight." Ethan says "Oliver is a knave or I am a knight." (2 characters)	If Oliver is a knight, his statement is true ⇒ Ethan is a knight. Then Ethan's statement "Oliver is a knave or I am a knight" = "F or T" = T. Consistent. `{"Oliver": "knight", "Ethan": "knight"}`	Correct E=−4.47

E.6 TACO

TACO (Li et al., 2023) (Topics in Algorithmic Code generation) consists of competitive programming problems sourced from Codeforces, AtCoder, and related platforms. We use the verified subset (`likaixin/TACO-verified`), filtered to drop `VERY_HARD` problems (0% solve rate). Each problem includes a description, input/output format, constraints, and test cases.

Verification is execution-based: candidate code is extracted from the chain-of-thought response, executed against all test cases with a 4-second timeout per test, and labelled correct only if every test passes. $E_{\text{constraint}}$ does not apply; the binary execution label serves as both the training signal and the evaluation metric. The primary metric is pass@1.

Contamination note. TACO problems are sourced from public competitive-programming sites. Per-difficulty pass@1 (particularly `HARD`) is the cleanest signal for genuine reasoning; `EASY` problems are likely seen during pretraining.

E.7 Knights and Knaves

Knights and Knaves (K&K) puzzles (Xie et al., 2025) are formal deductive reasoning problems. Each puzzle presents n characters (2-8), each either a knight (always tells the truth) or a knave (always lies), each making exactly one statement. The task is to determine the unique assignment of knight/knave labels satisfying all statements simultaneously.

We use the dataset of Xie et al. (2025) from HuggingFace (`K-and-K/knights-and-knaves`): 6,200 train and 700 test problems (100 per tier, 2ppl-8ppl). Candidates are classified by exact match against the ground-truth boolean assignment. A statement-consistency constraint checker independently verifies $E_{\text{constraint}}$=0 ⇔ correct on all 34,120 candidates checked, confirming perfect verification.

Token lengths are short (p50∼900 tokens for problem + candidate CoT); we use `max_length=1536`. Per-model accuracy reveals an extreme split: Gemma-4-26B and GPT-

OSS-20B achieve ∼99% accuracy, while Qwen-2.5-7B (19.7%) and LLaMA-3.1-8B (11.7%) rarely solve puzzles beyond 3 characters. This split creates a strong model-identification confound in the multi-model setting (see Sec. 4.2).

E.8 Qualitative Examples

Tab. 10 shows one problem-candidate pair per dataset, illustrating the input format, reasoning style, and verification signal. The examples illustrate several properties of the benchmarks. GSM8K candidates are short chain-of-thought derivations with a numeric answer delimiter; the verification signal is binary answer matching. MuSR candidates involve multi-hop narrative reasoning over thousands of characters; correct answers require tracking entities and evidence across paragraphs. TravelPlanner candidates are structured JSON itineraries verified against 8 constraint dimensions; even "correct" candidates (zero violations) may vary in quality, motivating the continuous scoring. TACO candidates are executable Python code verified by automated test execution; the most common failure mode is `wrong_answer` (56.6% of negatives), where syntactically valid code produces incorrect output on edge cases.

F Extended Results and Analysis

Evaluation regime. The ablations, energy diagnostics and selective-prediction results in this appendix (Tabs. 11 to 13, 15 and 20) were computed under the shuffled-candidate evaluation used during development, which predates the deterministic $T{=}0$ re-evaluation reported in Tab. 1. Absolute pass@1 therefore differs from the main table by up to 0.8 pp (GSM8K 96.9% vs. 97.0%, MuSR 66.9% vs. 67.7%). The transfer results (Tabs. 26 and 27) use the $T{=}0$ evaluation and match Tab. 1 directly. Every comparison within a given table is computed under a single regime.

Extended results ask fourteen questions across four themes: **Design sensitivity** (Sec. F.1)

Q1: How does ensemble size affect ranking accuracy and uncertainty estimation?
Q2: How does candidate-pool size affect selection quality?
Q3: How much does generator diversity contribute beyond sampling from a single generator?

Energy and decision diagnostics (Secs. F.2 to F.5)

Q4: How correlated are the adapters' decision-relevant ranking errors?
Q5: How clearly does the energy function separate correct from incorrect candidates?
Q6: Can per-adapter energy calibration improve uncertainty estimates without changing selection?
Q7: How does selection performance vary with task difficulty?
Q8: Does ensemble uncertainty identify errors well enough to support abstention?
Q9: When does uncertainty- or constraint-triggered regeneration improve pass@1?

Matched-checker and cost comparisons (Secs. F.6 and F.7)

Q10: On which constraint dimensions do the Distributional EBM and frontier models differ?
Q11: What changes when frontier models receive the same constraint-checker feedback?
Q12: Where does the Distributional EBM lie on the accuracy–cost trade-off?

Generalisation and confounding (Secs. F.8 and F.9)

Q13: How well does a quality scorer transfer across domains without retraining?
Q14: Does the quality scorer exploit model identity, and can this shortcut be removed?

F.1 Ablation Studies

We isolate five design choices: ensemble size (K), candidate pool size (N), single-model vs. multi-model training, backbone architecture, and energy decomposition. All ablations use ModernBERT unless noted; DeBERTa results are included where the 512-token context does not truncate candidates.

Ensemble size (Tab. 11, Fig. 5). ***(Q1:** How does ensemble size affect ranking accuracy and uncertainty estimation?)* Increasing K from 1 to 5 has negligible effect on pass@1 across all datasets: GSM8K stays at 96.9-97.2%, MuSR fluctuates between 64.9% and 66.9%, and TACO is constant

at 88.6%. The one exception is TravelPlanner, where $K \geq 3$ provides a +4 pp lift (24.0%→28.0%), likely because the continuous violation labels create noisier gradients that benefit from ensemble averaging. The ensemble is justified not by accuracy but by uncertainty estimation: σ-AUROC is undefined at $K=1$ (single adapter has no variance) and improves to 0.87 at $K=5$ on K&K. DeBERTa shows the same pattern on GSM8K (96.4-96.6% across K) and is not evaluated on TravelPlanner, TACO, or K&K due to its 512-token context truncating candidates.

Candidate pool size (Tab. 12, Fig. 6). ***(Q2:** How does candidate-pool size affect selection quality?)* The quality scorer's advantage over random grows with pool size on binary-label tasks. On GSM8K, the margin increases from +4.0 pp at $N=4$ to +4.3 pp at $N=32$. On TACO, pass@1 scales from 75.0% ($N=4$) to 88.6% ($N=32$), closing 86% of the random-to-oracle gap. On MuSR, the advantage peaks at $N=8$ (+14.5 pp over random) then diminishes at $N=16$ and $N=32$ as the pool becomes saturated with similar candidates, indicating diminishing returns on long-context reasoning tasks.

Single-model vs. multi-model (Tab. 13). ***(Q3:** How much does generator diversity contribute beyond sampling from a single generator?)*

Table 11: Ensemble size (K) sweep across all datasets and both backbones. MB = ModernBERT, DB = DeBERTa. All multi model. TACO values are pre-DFR (the identity shortcut saturates at $K=1$, making the sweep uninformative for genuine quality discrimination; after DFR deconfounding, pass@1 is 88.6%). DeBERTa (512-token context) is not evaluated on TravelPlanner, TACO, or K&K because candidate outputs exceed its context window.

K	Backbone	GSM8K	MuSR	TravelPlanner	TACO	K&K
1	MB	97.2%	66.9%	24.0%	92.0%	99.9%
3	MB	97.0%	64.9%	28.0%	92.0%	100.0%
5	MB	96.9%	66.9%	28.0%	92.0%	99.9%
1	DB	96.5%	62.3%	–	–	–
3	DB	96.4%	63.6%	–	–	–
5	DB	96.6%	62.9%	–	–	–

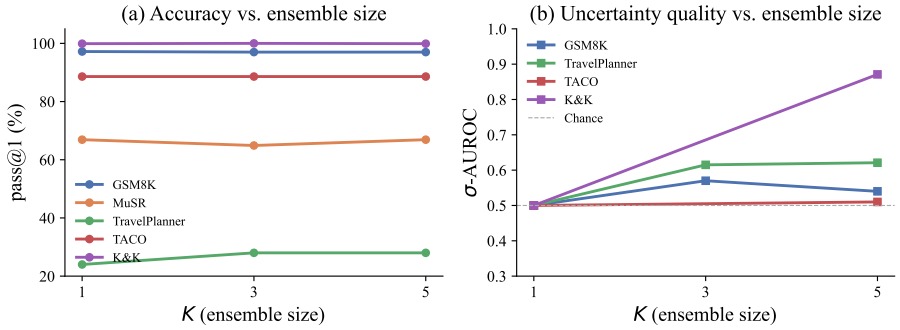


Figure 5: Effect of ensemble size K. **(a)** Pass@1 is stable across K on all datasets; the ensemble does not improve top-1 accuracy. **(b)** σ-AUROC improves from chance (0.5 at $K=1$) to 0.87 on K&K at $K=5$, confirming that structurally heterogeneous adapters produce informative uncertainty. The ensemble is justified by uncertainty estimation, not accuracy.

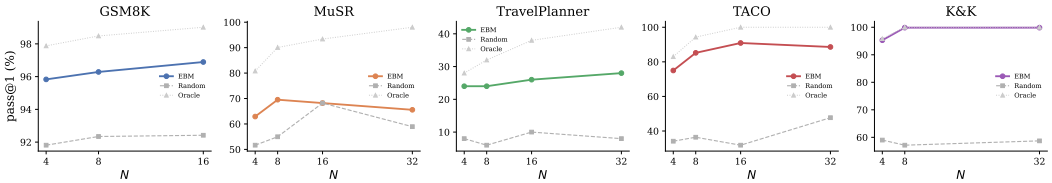


Figure 6: Candidate pool scaling. The quality scorer's advantage over random grows with N on GSM8K; on MuSR the advantage peaks at $N=8$ and diminishes as the pool saturates.

Multi-model candidate diversity provides the largest gain of any design choice on every dataset (Tab. 13). The effect is most dramatic on K&K (+59.8 pp) and TACO (+52.2 pp), where the generator

Table 12: Candidate pool size (N) ablation (ModernBERT, multi-model). **Bold** marks the best quality-scorer result per dataset. TACO values are pre-DFR; after deconfounding the N=32 result is 88.6%.

	GSM8K		MuSR		TACO	
N	Quality scorer	Random	Quality scorer	Random	Quality scorer	Random
4	95.8%	91.8%	62.9%	51.7%	75.0%	34.1%
8	96.3%	92.3%	**69.5%**	55.0%	85.2%	36.4%
16	96.9%	92.4%	68.2%	68.2%	90.9%	31.8%
32	**96.9%**	92.6%	66.9%	56.3%	**92.0%**	47.7%

Table 13: Single-model vs. multi-model comparison (ModernBERT, K=5). Single-model uses Qwen-2.5-7B only (N=16). Multi-model uses N=16 (4 models $\times$ 4 candidates); the main results (Tab. 1) use N=32, explaining small differences (*e.g.*, MuSR 66.9% here vs. 67.7%). **Bold** marks the better result per column.

	GSM8K	MuSR	TravelPlanner	TACO	K&K
Single-model Distributional EBM	93.1%	53.6%	8.0%	36.4%	40.1%
Multi-model Distributional EBM	**96.9%**	**66.9%**	**28.0%**	**88.6%**	**99.9%**
Δ	+3.8 pp	+13.3 pp	+20.0 pp	+52.2 pp	+59.8 pp

accuracy gap between the strongest and weakest model is largest. On MuSR, single-model training (Qwen only, 53.6%) barely exceeds random (51.0%); multi-model reaches 66.9% because the four generators provide both diverse error patterns for contrastive learning and higher oracle coverage (98.0% vs. 86.8%). On GSM8K, the gain is smallest (+3.8 pp) because all models are 91-97% correct (contamination), leaving little room for multi-model diversity to help.

Measured inter-adapter correlation. ***(Q4:** How correlated are the adapters' decision-relevant ranking errors?)* Proposition 3.1 makes the inter-adapter correlation ρ the lever of the ensemble, and every diversity choice (heterogeneous configuration, independent optimisation, bagged data splits) is a move to lower it. Tab. 14 measures ρ directly at K=5. The quantity the proposition uses is the correlation of the members' Bradley-Terry ranking mistakes over (correct, incorrect) pairs, not the correlation of raw energies; measured that way it is 0.51 to 0.71 across the benchmarks, below the raw energy-correlation (0.62 to 0.98) on every dataset because members agree on easy candidates but disagree on the decision-relevant hard pairs. Member accuracy q tracks task difficulty (K&K 0.94, TACO 0.87, GSM8K 0.78, MuSR 0.62) and the usefulness of σ_{quality} follows q as the (q, ρ) regime predicts: strong on K&K (σ-AUROC 0.87) and weak on MuSR (0.44). TravelPlanner has $q < \frac{1}{2}$ under its continuous labels, so the proposition does not apply there and its error-ρ is shown for reference only.

Table 14: Measured inter-adapter correlation at K=5 (off-diagonal mean of the $K \times K$ matrix per evaluation pool). Lower ρ means more diverse members and a stronger ensemble (Proposition 3.1); higher q means more accurate members. Energy-ρ correlates the raw per-adapter energies; error-ρ correlates the members' ranking mistakes over (correct, incorrect) pairs, the quantity Proposition 3.1 uses.

Dataset	$q \uparrow$	Energy-ρ (Pearson) $\downarrow$	Energy-ρ (Spearman) $\downarrow$	Error-ρ (Proposition 3.1) $\downarrow$	n / pairs
TACO	0.87	0.846	0.792	0.509	88 / 9,720
MuSR	0.62	0.800	0.807	0.642	151 / 31,730
GSM8K	0.78	0.837	0.819	0.604	1,319 / 53,434
K&K	0.94	0.979	0.937	0.706	700 / 162,235
TravelPlanner	0.27	0.618	0.598	(0.553)	50 / 3,246

F.2 Energy Separation and Calibration

***(Q5:** How clearly does the energy function separate correct from incorrect candidates?)*

Tab. 15 and Fig. 8 report diagnostic metrics for the Distributional EBM across all five benchmarks; Fig. 7 visualises the correct and incorrect energy distributions.

K&K exhibits the largest energy gap (9.00) and highest Kendall τ (0.62), reflecting the clean binary signal from the statement-consistency constraint checker and the high candidate diversity. TACO's gap (2.49) and τ (0.44) are reported after DFR deconfounding; the pre-DFR values (4.30, 0.78)

were inflated by the model-identity shortcut. MuSR has the smallest gap (0.19) and lowest τ (0.14), consistent with the difficulty of discriminating correct from incorrect narrative reasoning where both candidates contain plausible multi-hop inferences. TravelPlanner achieves the highest τ (0.49) despite a moderate gap (2.00), because its continuous violation labels produce a smoother energy landscape than binary correctness labels. σ-AUROC ranges from 0.44 (MuSR, anti-correlated) to 0.87 (K&K), confirming that ensemble uncertainty is task-dependent: reliable on tasks with clear correct/incorrect boundaries, unreliable on nuanced reasoning. ECE is moderate across all datasets (0.21-0.43), with MuSR the best-calibrated (0.21) and GSM8K the worst (0.43).

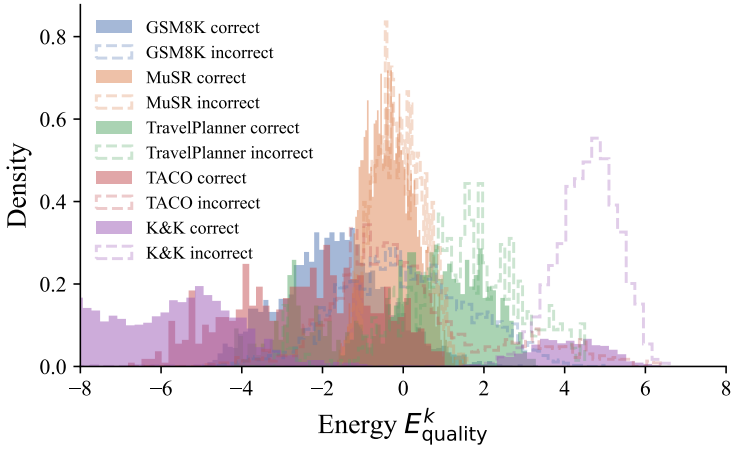


Figure 7: Energy distributions for correct (filled) and incorrect (dashed outline) candidates on GSM8K and MuSR (ModernBERT, multi-model). GSM8K shows clear bimodal separation; MuSR distributions overlap substantially, consistent with the smaller energy gap (0.19 vs. 1.76) and the difficulty of discriminating correct from incorrect narrative reasoning.

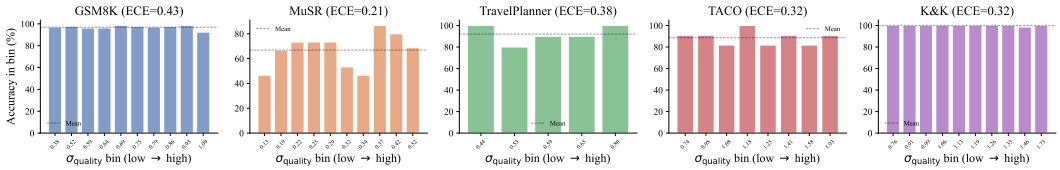


Figure 8: Calibration reliability diagrams. Bars show accuracy within each σ_{quality} bin. On GSM8K, accuracy is uniformly high across all bins (ECE = 0.43, the quality scorer is poorly calibrated). On MuSR, accuracy fluctuates non-monotonically with σ_{quality}, confirming that the uncertainty signal is not reliably informative on this task.

Table 15: Energy separation metrics (ModernBERT, multi-model, K=5). Energy gap = $\bar{E}$(incorrect) − $\bar{E}$(correct); positive means the quality scorer correctly assigns lower energy to correct candidates. **Bold** marks the best value per metric (higher is better for gap, τ, AUROC; lower for ECE).

Metric	GSM8K	MuSR	TravelPlanner	TACO	K&K
$\bar{E}$(correct)	−1.71	−0.32	−1.26	−2.44	−4.47
$\bar{E}$(incorrect)	0.04	−0.12	0.74	0.04	4.52
Energy gap	1.76	0.19	2.00	2.49	**9.00**
Kendall τ	0.27	0.14	0.49	0.44	**0.62**
σ-AUROC	0.54	0.44	0.62	0.51	**0.87**
ECE	0.43	**0.21**	0.28	0.32	0.32

Energy-scale calibration across adapters. ***(Q6:*** *Can per-adapter energy calibration improve uncertainty estimates without changing selection?)* Bradley-Terry training constrains only within-adapter energy differences, so each member's energies are free up to an additive offset (and, in practice, a scale). Selection is the argmin of the mean energy, so a per-member offset shifts every candidate equally and cannot change the pick; only σ_{quality}, the across-member spread, is affected. We therefore centre and unit-scale each member's energies on the validation split before recomputing σ_{quality}. Of seven variants tested (raw, centring, standardisation, unit-scaling, min-max, robust, quantile), none

moves pass@1 by more than 1 pp, and σ-AUROC sharpens most where it was weakest (Tab. 16); K&K and TravelPlanner are already near their ceiling and do not move.

Table 16: σ-AUROC before and after centring and unit-scaling each adapter's energies on the validation split (higher is better), with the change in pass@1.

Dataset	Raw	Recalibrated	Δ pass@1
GSM8K	0.54	0.58	+0.1 pp
TACO	0.51	0.57	+1.1 pp
TravelPlanner	0.62	0.62	none
K&K	0.87	0.87	none

F.3 Per-Difficulty Analysis

*(**Q7:** How does selection performance vary with task difficulty?)*

This section breaks down results by task difficulty to reveal where the Distributional EBM excels and where it struggles.

TACO (Tab. 17). The deconfounded quality scorer degrades gracefully with difficulty: 93.8% on EASY, 90.9% on MEDIUM, and 80.0% on HARD. It matches Sonnet 4.6 on MEDIUM and HARD problems, and substantially outperforms Qwen-72B at every difficulty level (by 16-60 pp). Opus 4.6 achieves 100% across all tiers, reflecting the advantage of a much larger model on competitive programming.

Table 17: TACO pass@1 by difficulty (multi-model). The quality scorer matches or exceeds Sonnet 4.6 on MEDIUM problems. n = number of test problems.

Difficulty	Quality scorer	Opus 4.6	Sonnet 4.6	Qwen-72B	n
EASY	93.8%	100.0%	100.0%	77.1%	48
MEDIUM	90.9%	100.0%	90.9%	45.5%	11
MEDIUM_HARD	81.8%	100.0%	90.9%	27.3%	11
HARD	80.0%	100.0%	80.0%	20.0%	5
UNKNOWN	76.9%	100.0%	92.3%	84.6%	13

Knights and Knaves (Tab. 18). The multi-model Distributional EBM achieves 100% pass@1 on all tiers except 8-character puzzles (99.0%), demonstrating near-perfect discrimination even as combinatorial complexity grows exponentially. Gains over greedy scale with difficulty: +32 pp on 2ppl to +55 pp on 7ppl. The single-model Distributional EBM (Qwen-only) shows a different pattern: gains are largest on the hardest tiers (+23.1 pp on 8ppl, +14.5 pp on 6ppl), suggesting that the quality scorer adds the most value precisely where the generator is weakest. Qwen-72B degrades steeply from 75% (2ppl) to 10% (8ppl), falling below random (58.7%) from 5ppl onward.

Table 18: K&K pass@1 by puzzle size (number of characters). The multi-model Distributional EBM achieves 100% on all tiers except 8ppl. **Bold** marks the best result per tier. Single-model Distributional EBM gains are largest on the hardest tiers.

	Multi-model (N=32)			Single-model (N=16)			Frontier	
Tier	Greedy	Distributional EBM	Lift	Greedy	Distributional EBM	Lift	Qwen-72B	Random
2ppl	68.0%	**100.0%**	+32.0 pp	58.7%	58.7%	+0.0 pp	75.0%	58.7%
3ppl	56.0%	**100.0%**	+44.0 pp	45.7%	48.9%	+3.2 pp	63.0%	58.7%
4ppl	60.0%	**100.0%**	+40.0 pp	25.8%	38.7%	+12.9 pp	48.0%	58.7%
5ppl	59.0%	**100.0%**	+41.0 pp	28.2%	29.5%	+1.3 pp	32.0%	58.7%
6ppl	55.0%	**100.0%**	+45.0 pp	17.7%	32.3%	+14.5 pp	29.0%	58.7%
7ppl	45.0%	**100.0%**	+55.0 pp	14.6%	19.5%	+4.9 pp	16.0%	58.7%
8ppl	52.0%	**99.0%**	+47.0 pp	10.3%	33.3%	+23.1 pp	10.0%	58.7%

MuSR (Tab. 19). MuSR's three subsets differ in structure: murder mysteries are 2-choice (50% random baseline), while object placements and team allocation are 3-choice (33% random). Average candidate accuracy is highest on murder mysteries (62.2%) and lowest on object placements (52.1%),

reflecting the difficulty of spatial reasoning relative to narrative deduction. The quality scorer's gains are largest on the 3-choice subsets where the random baseline is lower and there is more room for discrimination.

Table 19: MuSR per-subset accuracy (multi-model). Murder mysteries are 2-choice (50% random); objects and team are 3-choice (33% random).

Subset	Avg. candidate accuracy	Random baseline	n choices
Murder mysteries	62.2%	50%	2
Object placements	52.1%	33%	3
Team allocation	57.2%	33%	3

F.4 Selective Prediction via σ-Abstention

*(**Q8:** Does ensemble uncertainty identify errors well enough to support abstention?)*

Tab. 20 reports pass@1 on the retained set after abstaining on the $X\%$ of problems with highest σ_{quality}. On K&K, selective prediction reaches 100% by 20% abstention, reflecting the strong σ-AUROC (0.87): the ensemble reliably identifies the single wrong prediction. On TravelPlanner, abstaining on 40% of problems raises pass@1 from 28.0% to 36.7% (+8.7 pp), confirming that σ_{quality} captures meaningful quality uncertainty on continuous-label tasks. On GSM8K, selective prediction is flat (96.9-97.4%) because the model is already near-ceiling: there are only $\sim$40 wrong predictions out of 1,319. On MuSR, accuracy *decreases* with abstention (66.9%$\rightarrow$61.9% at 30%), consistent with the anti-correlated σ-AUROC (0.44): the ensemble is more confident on wrong predictions than right ones, so abstaining on high-σ_{quality} problems removes easy ones. On TACO (after DFR), selective prediction provides only +2.3 pp at 50% abstention (88.6%$\rightarrow$90.9%), down from +5.3 pp pre-DFR, because the pre-DFR σ was capturing model-identity uncertainty rather than quality uncertainty.

Table 20: Selective prediction: pass@1 on retained problems after abstaining on the most uncertain (highest σ_{quality}) fraction. **Bold** marks where a dataset reaches 100% accuracy. TACO values are post-DFR; selective prediction is weak (+2.3 pp at 50% abstention) because the deconfounded σ no longer captures model-identity uncertainty. MuSR pass@1 fluctuates non-monotonically with abstention (61.9–66.9%); the anti-correlated σ-AUROC (0.44) means high-σ problems are not preferentially incorrect, so abstention does not concentrate on the wrong predictions.

Abstain %	GSM8K (MB)	MuSR (MB)	TravelPlanner	TACO	K&K
0%	96.9%	66.9%	28.0%	88.6%	99.9%
10%	97.4%	66.7%	28.9%	88.6%	99.8%
20%	97.3%	65.0%	30.0%	88.6%	**100.0%**
30%	97.2%	61.9%	31.4%	88.5%	**100.0%**
40%	97.2%	64.4%	36.7%	90.4%	**100.0%**
50%	97.1%	66.7%	36.0%	90.9%	**100.0%**

F.5 Two-Pass σ-Triage Analysis

*(**Q9:** When does uncertainty- or constraint-triggered regeneration improve pass@1?)*

This section reports the two-pass σ-triage loop as an additional experiment outside the main selection pipeline (Sec. 3, Fig. 2): the mechanism, its thresholds, results on all tasks, and the decision surface.

F.5.1 TWO-PASS INFERENCE WITH σ-TRIAGE

We extend single-pass reranking with a two-pass inference loop driven by ensemble uncertainty and constraint satisfaction, recovering when no good candidate exists or the quality scorer is uncertain (Alg. 1). Multiple frozen LLMs each generate N_m candidates via temperature sampling, which are pooled into the first-pass set $\mathcal{Y}_1$ and scored. Each candidate receives μ_{quality} and σ_{quality} from the K=5 ensemble and $E_{\text{constraint}}$ from the analytical constraint checker. Let $y_1^* = \arg\min_{y \in \mathcal{Y}_1} E(x, y)$ denote the first-pass selection. The system then takes one of three actions, governed by thresholds $\theta_\sigma < \theta_{\text{abstain}}$:

$$\text{action}(y_1^*) = \begin{cases} \text{abstain} & \text{if } \sigma > \theta_a, \\ \text{regenerate} & \text{if } \sigma > \theta_\sigma \ \vee \ C > 0, \\ \text{accept} & \text{otherwise}, \end{cases} \tag{5}$$

where $\sigma = \sigma_{\text{quality}}(y_1^*)$, $C = E_{\text{constraint}}(y_1^*)$, $\theta_a = \theta_{\text{abstain}}$, and the cases are evaluated in order. On regenerate, constraint violations are formatted as natural-language feedback and used to draw N_2 additional candidates $\mathcal{Y}_2$. The pooled set $\mathcal{Y} = \mathcal{Y}_1 \cup \mathcal{Y}_2$ is rescored, and the final selection is $y^* = \arg\min_{y \in \mathcal{Y}} E(x, y)$, subject to a final abstention check on $\sigma_{\text{quality}}(y^*)$.

The σ-triage rule (Eq. (5)) uses thresholds θ_σ=0.8 and θ_{abstain}=1.5, both chosen by grid search on a held-out validation split to maximise the pass@1 lift from regeneration while keeping the abstention rate below 10%.

F.5.2 RESULTS

Two-pass σ-triage (Eq. (5)) helps when pass-1 has headroom and feedback is structured: on TravelPlanner (Qwen-only generator pool, used to isolate the feedback effect from candidate diversity), it improves pass@1 from 74.0% to 80.0% (+6.0 pp). On TACO (deconfounded), two-pass is neutral (88.6%); on GSM8K, it is mildly counterproductive (-1.7 pp) precisely because the generator has memorised the answer and feedback distracts retrieval, a diagnostic feature: σ-triage detects which regime each task falls into rather than applying a one-size-fits-all loop. For selective prediction (Geirhos et al., 2020), σ_{quality} is task-dependent: on TravelPlanner, abstaining on 40% of problems raises pass@1 from 28.0% to 36.7% (+8.7 pp); on MuSR, σ does not signal correctness (AUROC 0.44), so the quality scorer cannot triage reliably. The when-it-helps regime appears to depend on constraint-checkability and label discreteness rather than scoring task; we discuss the pattern in §F.5. Fig. 9 visualises the $(\mu_{\text{quality}}, \sigma_{\text{quality}})$ space; full results are in §F.5 and §F.4.

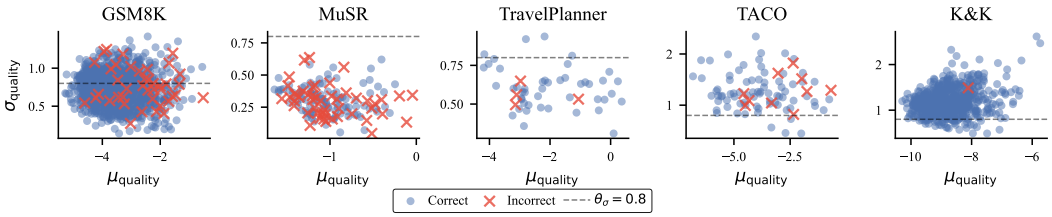


Figure 9: σ-triage decision surface (μ_{quality} vs. σ_{quality} per problem). On GSM8K, incorrect selections (red) are interspersed with correct ones. On MuSR, all $\sigma_{\text{quality}} < \theta_\sigma$=0.8 (dashed), so triage never triggers. On TravelPlanner, the wider σ_{quality} spread reflects continuous violation labels.

Alg. 1 provides the full inference pseudocode for the two-pass σ-triage mechanism.

Tab. 21 reports two-pass σ-triage results across all datasets and pool configurations. The mechanism helps when three conditions hold: (a) pass-1 has headroom (not already near-ceiling), (b) the feedback is structured enough to guide regeneration, and (c) the generator can act on it. TravelPlanner single-model satisfies all three: constraint violations (budget overruns, missing meals, invalid routes) provide concrete feedback, yielding +6.0 pp. TACO multi-model (confounded) triages all 88 problems but adopts no pass-2 candidate, so pass@1 is unchanged. After DFR deconfounding, TACO two-pass is neutral (88.6%→88.6%) because the deconfounded quality scorer evaluates each candidate on quality alone, so re-pooling low-quality pass-2 candidates does not corrupt the ranking. GSM8K two-pass reduces pass@1 by 1.7 pp because doubt-inducing feedback causes the contaminated generator to second-guess its (memorised) correct answers.

Two-pass helps when pass-1 has headroom, feedback is structured (*e.g.*, specific constraint violations), and the generator can act on it. TravelPlanner single-model satisfies all three conditions. On K&K

Algorithm 1 Inference with σ-triage

Require: Problem x, frozen LLMs $\{G_1, \ldots, G_M\}$, ensemble $\{f_k\}_{k=1}^{K}$, constraint set $\{C_j\}$, thresholds θ_σ, θ_{abstain}, weight λ
Ensure: Selected candidate y^* or ABSTAIN

1: **Pass 1: Generate.** For each LLM G_m, sample N_m candidates via temperature sampling. Pool: $\mathcal{Y}_1 = \{y_1, \ldots, y_{N_1}\}$.
2: **for** each $y_i \in \mathcal{Y}_1$ **do**
3: Parse y_i. If parse fails, set $E_{\text{constraint}}(x, y_i) = +\infty$.
4: Score with all K adapters: $\{E^1(x, y_i), \ldots, E^K(x, y_i)\}$.
5: $\mu_{\text{quality}}(x, y_i) \leftarrow \text{mean}(\{E^k\})$, $\sigma_{\text{quality}}(x, y_i) \leftarrow \text{std}(\{E^k\})$.
6: $E_{\text{constraint}}(x, y_i) \leftarrow \sum_j w_j \cdot C_j(x, y_i)$.
7: $E(x, y_i) \leftarrow \mu_{\text{quality}}(x, y_i) + \lambda \cdot E_{\text{constraint}}(x, y_i)$.
8: **end for**
9: $y^* \leftarrow \arg\min_{y_i} E(x, y_i)$.
10: **if** $E_{\text{constraint}}(x, y^*) > 0$ **or** $\overline{\sigma_{\text{quality}}} > \theta_\sigma$ **then**
11: **Pass 2: Regenerate.** Format constraint violations of y^* as feedback.
12: Generate N_2 candidates conditioned on feedback: $\mathcal{Y}_2$.
13: Score all $y_i \in \mathcal{Y}_2$ (repeat lines 3-7). Pool: $\mathcal{Y} = \mathcal{Y}_1 \cup \mathcal{Y}_2$.
14: $y^* \leftarrow \arg\min_{y_i \in \mathcal{Y}} E(x, y_i)$.
15: **end if**
16: **if** $\sigma_{\text{quality}}(x, y^*) > \theta_{\text{abstain}}$ **then**
17: **return** ABSTAIN
18: **else**
19: **return** y^*
20: **end if**

Table 21: Two-pass σ-triage results across datasets. θ_σ=0.8 unless noted, N_2=8 pass-2 candidates from Qwen-2.5-7B with constraint/error feedback. **Bold** marks the only configuration where two-pass improves pass@1. "–" = not applicable (no problems triaged or no pass-2 candidates adopted).

Dataset	Pool	Triage rate	Pass-1	Two-pass	Δ	Pass-2 adopted
GSM8K	Single (MB v1)	201/1319 (15%)	93.1%	91.4%	−1.7 pp	56/201
MuSR	Multi	0/151 (0%)	67.7%	67.7%	0.0 pp	–
TravelPlanner	Single	18/50 (36%)	74.0%	**80.0%**	+6.0 pp	9/18
TravelPlanner	Multi	17/50 (34%)	92.0%	92.0%	0.0 pp	1/17
TACO (pre-DFR)	Multi	88/88 (100%)	92.0%	92.0%	0.0 pp	0/88
TACO (DFR)	Multi	84/88 (95.5%)	88.6%	88.6%	0.0 pp	–
TACO	Single	0/88 (0%)	36.4%	36.4%	0.0 pp	–
K&K	Single (θ_σ=0.5)	396/499 (79.4%)	85.2%	85.0%	−0.2 pp	29/396
K&K	Multi (θ_σ=0.5)	700/700 (100%)	99.9%	99.9%	0.0 pp	0/700

single-model (θ_σ=0.5), two-pass is near-neutral (−0.2 pp overall) but shows a tier-dependent pattern: it helps on easy puzzles (2-4ppl: +2 to +4 pp) where constraint feedback gives actionable corrections, but hurts on hard puzzles (7-8ppl: −10 to −13 pp) where pass-2 candidates look confident to the ensemble but are more often wrong. On K&K multi-model, all 700 problems trigger triage (mean σ_{quality}=0.931) due to high inter-model disagreement, but only 1 problem has a constraint violation receiving actionable pass-2 feedback; the remaining 699 already satisfy all constraints, so two-pass has zero effect (99.9%→99.9%). On TACO multi-model, all 88 problems trigger triage ($\sigma_{\text{pool}} > 0.8$ due to high inter-model disagreement), but pass-2 candidates from Qwen-7B with opaque error feedback are uniformly poor and none is adopted, so pass@1 is unchanged. Execution errors give the generator no actionable signal to condition on.

When does σ-triage help? Two patterns are visible. *(a) Constraint-checkability.* σ-triage adds value where regeneration receives *actionable* feedback: TravelPlanner's per-constraint violation messages (budget over by $X, missing meal on day 2) translate directly into pass-2 prompts (+6.0 pp); TACO's opaque `wrong_answer` and doubt-inducing feedback on contaminated GSM8K do not. *(b) Quality-distribution multimodality.* Triage benefits tasks where candidates cluster into discrete modes the ensemble can disagree about (K&K's statement-consistency states); on tasks where candidates lie

on a continuous quality manifold (MuSR's narrative inferences), σ measures presentation variation rather than correctness. σ-triage therefore selects which regime each task falls into rather than applying a one-size-fits-all loop.

Regeneration with matched feedback. To compare like with like against the frontier self-verification loop of Tab. 24, we give our own regeneration loop (Eq. (5)) the same failing-case feedback (the failing test input on TACO, the violated constraints on TravelPlanner) and run it for a few rounds across all four generators. On TACO, pass@1 rises from 88.6% to 93.2% at about \$0.006 per problem, with no new model or constraint checker, recovering most of the gap to the frontier loop at 6 to 67 times lower cost. TravelPlanner shows no gain: its multi-generator pool is already near the oracle (0.040 vs. 0.028), so every regenerated plan is worse than the existing pick and the Distributional EBM correctly keeps it.

Table 22: Our regeneration loop given the same failing-case feedback as the frontier loop of Tab. 24, a few rounds across all four generators. "Fixed / adopted" counts problems whose regenerated candidate was selected and correct.

Task	Metric	Base	After regen.	Fixed / adopted	Cost / problem	Frontier loop (Opus)
TACO	pass@1 ↑	88.6%	**93.2%**	4/10 (3 at round 1)	∼\$0.006	\$0.039
TravelPlanner	mean viol. ↓	0.040	0.040	0/9	∼\$0.009	\$0.60

F.6 TravelPlanner: Per-Constraint Frontier Comparison

*(**Q10:** On which constraint dimensions do the Distributional EBM and frontier models differ?)*

Tab. 23 decomposes the TravelPlanner violation score by constraint dimension, revealing where the Distributional EBM and frontier models have complementary strengths. Frontier models (Opus, Sonnet, Qwen-72B) systematically fail on connectivity (24-28% satisfaction) because they hallucinate transport routes not in the reference database. The analytical constraint checker verifies routes against the database directly, giving the Distributional EBM near-perfect connectivity (86% satisfaction). Conversely, frontier models outperform on budget (Opus: 0.040 vs. Distributional EBM: 0.030), reflecting stronger numerical reasoning. Preferences are the hardest dimension for all methods, including oracle (0.180 mean violation).

Table 23: Per-violation-dimension mean scores on TravelPlanner (50 test problems). Lower is better. **Bold** marks the best (lowest) value per row.

Dimension	Oracle	Constraint checker	Quality scorer + constraint checker	Quality scorer	Opus 4.6	Sonnet 4.6	Qwen-72B	Random
Budget	0.018	**0.012**	0.030	0.176	0.040	0.078	0.062	0.238
Completeness	0.009	0.011	0.031	**0.003**	0.086	0.048	0.110	0.260
Hallucination	0.005	0.011	0.011	**0.002**	0.015	0.037	0.070	0.119
Connectivity	**0.000**	0.003	0.058	0.062	0.418	0.393	0.375	0.232
Preferences	**0.180**	0.187	0.213	0.220	0.213	0.213	0.280	0.387
Diversity	0.011	0.080	0.022	**0.000**	0.007	0.054	0.074	0.148

Frontier models systematically fail on connectivity (24-28% satisfaction) because they hallucinate transport routes not present in the reference database. The analytical constraint checker verifies routes against the database directly, giving the Distributional EBM a structural advantage on this dimension. Conversely, frontier models outperform on budget (Opus 4.6: 0.040 vs. Distributional EBM: 0.030), reflecting their stronger numerical reasoning. Preferences are the hardest dimension for all methods, including oracle (0.180 mean violation even for the best candidate).

Independent LLM-judge validation. To check that the violation score reflects genuine plan quality rather than artefacts of the constraint checker, two independent LLM judges (Opus 4.6 and Sonnet 4.6 (Anthropic), $T{=}0$) score 120 candidates on a 1–5 scale using the original TravelPlanner criteria (Xie et al., 2024). The judges agree strongly with each other (Spearman $\rho{=}0.67$, $p < 10^{-16}$) and both correlate negatively with the violation score (Opus: $\rho{=}{-}0.46$; Sonnet: $\rho{=}{-}0.38$; both $p < 10^{-5}$). Opus 4.6 rates plans selected by the Distributional EBM higher than oracle plans on average (3.86 vs. 3.67), suggesting the quality scorer captures holistic coherence beyond what constraints alone measure.

Frontier models given the same constraint checker. *(**Q11:** What changes when frontier models receive the same constraint-checker feedback?)* The natural agentic baseline on the checkable tasks

is a frontier model with tool access to our constraint checker in a self-verification loop: the model proposes an output, the constraint checker returns the violated constraints or the first failing test (never the expected answer), and the model revises, up to 32 rounds, stopping when the constraint checker reports zero violations. Round 0 reproduces the paper's zero-shot prompt, including TravelPlanner's reference information, so the frontier model is not handicapped. On accuracy the loop closes the gap: K&K ties at the ceiling, TACO reaches 98.9 to 100.0% against 88.6%, and on TravelPlanner Opus reaches 0.020 mean violation against 0.046 (Sonnet 0.055). The cost is the point: the loop issues multiple full frontier generations per problem, about one round on saturated K&K but 7 rounds (Opus) and 12 rounds (Sonnet) on TravelPlanner, each re-sending the roughly 15k-token reference information, at \$0.60 to \$0.77 per problem against about \$0.003 for the Distributional EBM, 4 to 250 times more across tasks (Tab. 24). Since $E_{\text{constraint}}$ is part of the Distributional EBM, a frontier model driven by it is our pipeline with a far costlier reasoning model; the experiment confirms that the 149M quality scorer reaches the same selection quality at a fraction of the cost and one local forward pass in place of 6 to 12 frontier calls, and it is why we make no accuracy claim for the quality scorer over frontier reasoning.

Table 24: Frontier models given the same constraint checker in a self-verification loop (up to 32 rounds), against the Distributional EBM on the same problems (K&K n=700, TACO n=88, TravelPlanner n=50). Accuracy is the Tab. 1 metric; cost is actual per-problem API billing, with mean rounds in parentheses.

Task	Distributional EBM	Opus + constraint checker	Sonnet + constraint checker	Distributional EBM \$	Opus \$ (rounds)	Sonnet \$ (rounds)
K&K (pass@1 ↑)	99.9	100.0	100.0	0.003	0.024 (1)	0.013 (1)
TACO (pass@1 ↑)	88.6	100.0	98.9	0.005	0.039 (1)	0.031 (2)
TravelPlanner (mean viol. ↓)	0.046	0.020	0.055	0.003	0.60 (7)	0.77 (12)

F.7 Cost-Matched Comparison

*(**Q12:** Where does the Distributional EBM lie on the accuracy–cost trade-off?)*

Tab. 25 reports per-problem cost against accuracy for every selector in Tab. 1, on the five test splits. Open-generator tokens are priced with each model's own tokenizer from the stored candidates; LLM-as-Judge from actual API billing (one call per candidate); every selector reranks the same 32-candidate pool, so generation (\$0.0016 to \$0.0051 per problem) is a shared, compute-matched cost and only the selector cost differs. The Distributional EBM's learned selection overhead is a single 149M encoder forward pass per candidate with no decoding, about \$0.003 per problem, identical to self-consistency; LLM-as-Judge costs \$0.04 to \$0.14, 16 to 29 times more, and is the only selector that edges the Distributional EBM on raw accuracy. Zero-shot rows are lower bounds: their generations were not stored, so they are priced at the pool's mean output length, and the frontier models generate more. At a matched budget the Distributional EBM sits on the accuracy-cost frontier: for the same \$0.003 as self-consistency it leads on MuSR, TACO and K&K, and on K&K it matches Opus 4.6 at roughly one fifth of Opus's cost.

Table 25: Per-problem cost (USD) against accuracy (Tab. 1) on the five test splits. Every pool method reranks the same 32 candidates, so generation is a shared cost; "≥" marks zero-shot lower bounds.

Method	GSM8K ↑	MuSR ↑	TP (viol.) ↓	K&K ↑	TACO ↑	\$/problem ↓
Qwen-72B zero-shot	95.4	54.9	0.121	39.0	64.8	≥0.0001–0.0005
Sonnet-4.6 zero-shot	97.0	68.0	0.103	98.7	95.5	≥0.0041–0.0129
Opus-4.6 zero-shot	97.0	72.0	0.098	99.7	100.0	≥0.0068–0.0215
Self-consistency	96.1	64.4	0.096	56.4	36.4	0.0016–0.0051
LLM-as-Judge (Sonnet)	99.2	73.5	0.084	99.9	73.9	0.0416–0.1427
Oracle (pool bound)	98.5	86.1	0.028	100.0	100.0	0.0016–0.0051
Distributional EBM (ours)	97.0	67.7	0.046	99.9	88.6	**0.0016–0.0051**

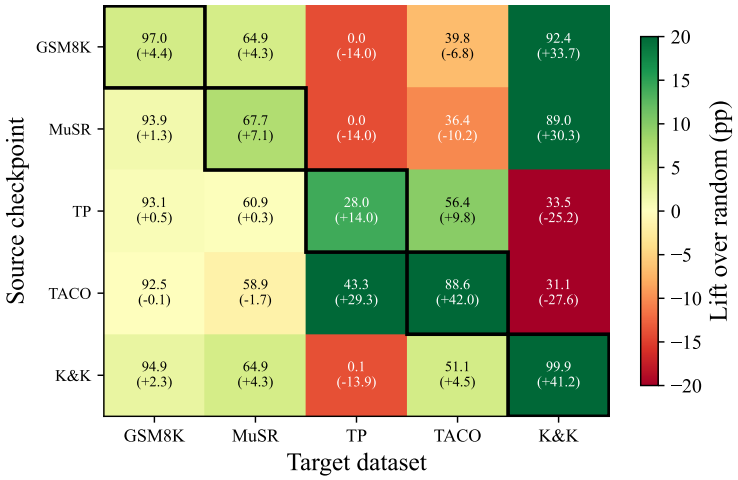


Figure 10: Zero-shot transfer heatmap (ModernBERT): each cell shows pass@1 and (lift over random). Colour encodes lift: green = positive transfer, red = negative. Black borders mark in-domain entries.

Table 26: Full zero-shot transfer matrix (quality-scorer pass@1, ModernBERT multi-model unless noted). Shaded cells mark in-domain (diagonal) entries. **Bold** highlights notable transfer results. Baselines for reference: self-consistency (SC) and random.

Source → Target	GSM8K	MuSR	MATH-500	TP	TACO	K&K
GSM8K (MB)	97.0%	64.9%	65.6%	0.0%	39.8%	92.4%
GSM8K (DB)	97.1%	63.6%	62.7%	0.0%	45.5%	68.4%
MuSR (MB)	93.9%	67.7%	67.6%	0.0%	36.4%	89.0%
MuSR (DB)	**96.1%**	62.9%	67.8%	0.0%	39.8%	88.4%
TP (MB)	93.1%	60.9%	67.2%	28.0%	56.4%	33.5%
TACO (MB)	92.5%	58.9%	67.4%	43.3%	88.6%	31.1%
K&K (MB)	94.9%	64.9%	0.0%	0.136$^{\diamond}$	51.1%	**99.9%**
Self-consistency	96.1%	64.4%	72.6%	12.0%	36.4%	56.4%
Random	92.6%	60.6%	68.4%	14.0%	46.6%	58.7%

F.8 Zero-Shot Transfer: Full Matrix

*(**Q13:** How well does a quality scorer transfer across domains without retraining?)*

Tab. 26 reports the full zero-shot transfer matrix, including both ModernBERT and DeBERTa checkpoints and the MATH-500 benchmark. Fig. 10 visualises the transfer asymmetry as lift over random.

Transfer is strongest within reasoning tasks but asymmetric: MuSR(DB)→GSM8K reaches 96.1%, matching in-domain self-consistency, whereas GSM8K→MuSR remains below the in-domain result. Transfer breaks under larger format and signal shifts: no checkpoint beats self-consistency on MATH-500, and reasoning-trained quality scorers fall below random on TACO. By contrast, TACO and TravelPlanner transfer positively to one another, while the backbone-dependent K&K results show that context length also affects transfer.

Cross-pool transfer (Qwen-train → multi-test). We also evaluate a Qwen-only-trained quality scorer on the multi-generator test pool (Tab. 27). It reaches 69.5% on MuSR, exceeding both multi-generator training (67.7%) and its single-generator result (53.6%), but reaches only 90.5% on GSM8K, below random (92.6%) and multi-generator training (97.0%). Cross-generator transfer therefore succeeds on MuSR but not on the near-saturated GSM8K training distribution.

Table 27: Cross-pool transfer: Qwen-only-trained quality scorer evaluated on the multi-model candidate test pool. *Single-train, single-test* repeats the entries from the corresponding row of Tab. 13 for context. All numbers are pass@1 with N=32 candidates per problem; multi-test pool is the same as the headline evaluation.

Configuration	MuSR	GSM8K
Single-train, single-test (Qwen-only)	53.6%	93.1%
Single-train, multi-test (cross-pool)	**69.5%**	90.5%
Multi-train, multi-test (in-domain)	67.7%	**97.0%**

F.9 Confounding Analysis

*(**Q14:** Does the quality scorer exploit model identity, and can this shortcut be removed?)*

We screen all five datasets for model-identity confounding by computing two diagnostics: (i) *per-model pick distribution* (does the quality scorer disproportionately select one generator?) and (ii) *cross-model correct energy spread* (do correct candidates from different generators receive similar energies?). Tab. 2 in Sec. 4.2 summarises the results.

Four datasets show near-uniform pick distributions and energy spreads below 0.25, indicating no model-identity confounding. On MuSR, the quality scorer picks the *weaker* model (Qwen, 51% accuracy) most often (36%), the opposite of what fingerprinting would produce. On K&K, despite the most extreme accuracy split in our study (Gemma/GPT-OSS at 99% vs. Qwen/LLaMA at 15%), picks remain uniform and the energy spread is just 0.11.

TACO is the sole outlier: the initial quality scorer selected Gemma on 94% of test problems with an energy spread of 6.67, indicating severe model-identity confounding. The six-phase analysis below details how this was detected, diagnosed, and mitigated.

TACO: detailed analysis. A six-phase confounding analysis on the TACO multi-model quality scorer revealed that the initial model exploited a model-identity shortcut rather than genuine reasoning-quality discrimination.

Phase 1-2: Detection. The quality scorer selected a Gemma-4-26B candidate on 83/88 test problems (94.3%). A naive "always pick random Gemma" baseline achieved 89.8% pass@1, leaving only +2.2 pp of genuine quality signal. Correct solutions from different models received dramatically different energies: Gemma correct at -4.47 vs. LLaMA correct at $+2.20$, a 6.67-point gap for identical correctness labels.

Phase 3: Style normalisation ablation. Progressive removal of formatting features (headers, LaTeX, IO normalisation, comments) collapsed both the cross-model energy gap (6.67 $\rightarrow \sim$1.0) and the within Gemma correct/incorrect gap (2.41 $\rightarrow$ 0.55), confirming that quality and identity features are carried by the same tokens.

Phase 4: Causal style-swap test. For 40 problems with correct Gemma solutions, three other models rewrote each solution in their own style with variable names, algorithm, and logic locked. Execution-verified identical code received +1.95 higher energy on average when presented in another model's formatting (LLaMA: +2.66, GPT-OSS: +2.20, Qwen: +0.97). This is causal proof of style bias (Geirhos et al., 2020).

Phase 5-6: Decomposition and case studies. Code-only scoring retained most of the model-identity gap (3.6-point spread), while explanation-only scoring collapsed to 62.5% pass@1. The identity signal lives primarily in code patterns (IO methods, function structure), not explanation text. All 5 non-Gemma picks by the original quality scorer occurred on EASY problems; on 7 problems the quality scorer picked an incorrect Gemma solution despite correct alternatives being available.

Mitigation: DFR. Last-layer retraining (DFR; Kirichenko et al., 2023) on group-balanced data (4 models $\times$ 2 labels, capped at 396 per cell) retrains only the K=5 energy heads (0.4% of parameters) while freezing the backbone and LoRA adapters. Tab. 28 reports the effect. This collapses the cross-model correct energy spread from 6.67 to 0.34 (95% reduction) while retaining 88.6% pass@1 with picks distributed uniformly across generators (22/24/34/20%). The deconfounded quality scorer

Table 28: Effect of DFR on TACO. Energy-head retraining only (0.4% of parameters); backbone and adapters frozen. Picks are selection shares over Qwen/LLaMA/Gemma/GPT-OSS. The 3.4 pp pass@1 cost is the formatting shortcut being removed: only 2.2 pp of the pre-DFR 92.0% exceeded a fixed heuristic that selects a random Gemma candidate (89.8%).

	Pre-DFR	Post-DFR
Picks (%)	1/1/94/4	**22/24/34/20**
Energy spread	6.67	**0.34**
pass@1 (n=88)	92.0%	88.6%

reaches 80.0% on HARD (n=5) and 76.9% on UNKNOWN (n=13); per-difficulty results are in Tab. 17. It is unchanged under two-pass re-pooling (88.6% → 88.6%; pre-DFR 92.0% → 92.0%), consistent with no pass-2 candidate being adopted by either quality scorer.

G Broader Impact

This paper presents work whose goal is to advance machine learning, specifically the verification of structured outputs from large language models. We outline expected positive and negative societal impacts.

Positive impacts. Uncertainty-aware verification of structured LLM outputs has beneficial applications in safety-critical domains where undetected errors propagate silently: *automated planning* (travel itineraries, logistics, scheduling) where a single budget or routing violation invalidates the entire plan; *code generation* where edge-case failures can cause silent data corruption or service outages; *multi-step mathematical and scientific reasoning* where intermediate errors compound; and *decision support* in regulated settings (legal, medical, financial) where outputs must satisfy domain rules verifiable in closed form. The decomposition $E = \mu_{\text{quality}} + \lambda \cdot E_{\text{constraint}}$ is auditable by design: $E_{\text{constraint}}$ is a deterministic constraint checker over explicit task constraints, and the distributional $(\mu_{\text{quality}}, \sigma_{\text{quality}})$ exposes *when the quality scorer is uncertain*, enabling principled abstention rather than confidently wrong outputs. Compared with scaling generators, the compact Distributional EBM is also more energy-efficient and democratises access to reliable structured reasoning without 100B+-parameter inference.

Negative impacts and risks. Overreliance on automated verification carries three risks. *(i) Spurious shortcuts.* Our confounding analysis (Sec. 4.2, §F.9) demonstrates that quality scorers can learn high-confidence but content-irrelevant signals (the TACO formatting shortcut), producing reliable-looking but unreliable outputs. Without the kind of cross-dataset auditing we report, such shortcuts go undetected. *(ii) Distribution shift.* A quality scorer trained on one candidate distribution may degrade silently when deployed against differently-styled generations or out-of-domain inputs (illustrated by the GSM8K→MATH-500 below-random transfer; see Tab. 26). *(iii) Automation bias.* Plans, code, or proofs that pass an automated constraint checker may be accepted with less human review than warranted, particularly when σ_{quality} is uncalibrated (e.g., MuSR σ-AUROC = 0.44; see Limitations in Sec. 4.2).

Mitigations. We recommend: (a) running confounding diagnostics (cross-model pick distributions and energy spreads) at deployment time, with last-layer retraining (DFR; Kirichenko et al., 2023) when shortcuts surface; (b) treating σ_{quality} as a regime-aware signal rather than a universal abstention threshold (§F.4 shows it is task-dependent); (c) preserving human review for high-stakes deployments and using σ-triage to route uncertain cases to humans rather than auto-accepting them; (d) documenting candidate-pool composition and constraint definitions when releasing checkpoints, since both interact materially with quality-scorer behaviour. The released artefact is the Distributional EBM, which performs selection rather than generation and therefore has lower direct misuse risk than a released LLM.

Scaling $E_{\text{constraint}}$. The decomposition $E = \mu_{\text{quality}} + \lambda \cdot E_{\text{constraint}}$ requires a domain expert to author each constraint term, which is the principal scalability bottleneck of the framework. Three directions seem viable. *(a) LLM-generated specifications:* a frontier model can be prompted to extract executable constraints from a natural-language task specification, with the Distributional

EBM then ranking candidates against the auto-generated constraint checker; the Distributional EBM inherits the spec-extractor's errors, but an exact deterministic constraint can be cheaper to author than to learn. *(b) Learned constraint heads:* a small model trained on labelled violation data can approximate $E_{\text{constraint}}$ for domains where authoring is hard but labelled examples exist (CI logs for code, structured grading rubrics for student work, regulatory checklists in compliance settings). *(c) Constraint DSLs:* domain-specific languages for stating constraints, analogous to integrity constraints in relational databases, lower the authoring burden without removing the human in the loop. We treat these as engineering frontiers rather than methodological gaps; the propositions of Sec. 3.2 hold regardless of how $E_{\text{constraint}}$ is obtained, as long as it is deterministic at inference time.