Distributional Energy-Based Models for Uncertainty-Aware Structured LLM Reasoning

Shireen Kudukkil Manchingal[†1] **Abhey Kalia**[1] **Fernanda Gonçalves**[1] **Shebin Rawther**[1]

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 violates budgets, fails test cases, or contradicts earlier deductions. We propose a decomposed energy function that combines a learned quality scorer with deterministic analytical 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 the standard deviation quantifies epistemic uncertainty, driving a two-pass inference loop that triggers targeted regeneration or abstention. Across five benchmarks (GSM8K, MuSR, TravelPlanner, TACO, Knights & Knaves), reranking 32 candidates from cheap 7B-class models matches Sonnet 4.6 on MuSR (67.7% vs. 68.0%) and reduces constraint violations by 53% relative to Opus 4.6 on TravelPlanner. A cross-dataset confounding analysis confirms genuine quality discrimination on four reasoning tasks and identifies a model-identity shortcut on code, mitigated via last-layer retraining. Scorers trained on difficult data transfer zero-shot: a MuSR-trained scorer achieves 93.9% on GSM8K without seeing a math problem.

1. Introduction

Large Language Models have advanced rapidly, achieving strong performance across translation, summarisation, code generation, and open-ended dialogue (Brown et al., 2020; OpenAI, 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., 2023), least-to-most decomposition (Zhou et al., 2023), and tree-structured search (Yao et al., 2023).

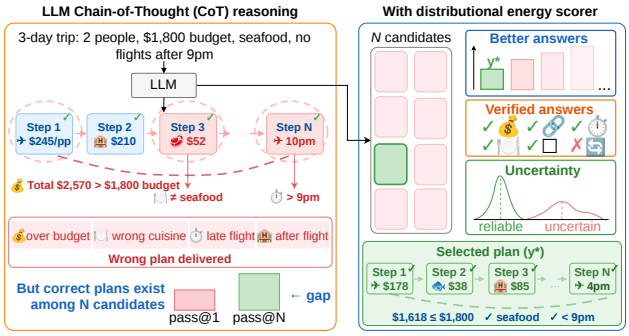

Figure 1. **Left:** Chain-of-thought scores each step locally; cross-step violations (budget, preference, temporal) are invisible and the wrong plan is delivered. **Right:** Distributional energy scores N candidates globally, verifies constraints analytically, and quantifies confidence via ensemble agreement. The selected plan satisfies all constraints with reliable confidence.

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. This limitation is starkest for structured outputs that must satisfy many constraints simultaneously, such as multi-day travel plans with budgets and route validity, competitive programming solutions that must pass all test cases, or multi-step mathematical derivations where a single arithmetic error invalidates the entire trace. As Figure 1 (left) illustrates, each step in a chain-of-thought trace may receive a local correctness signal, yet cross-step violations remain invisible to the local scorer: a budget overrun spanning all days, a cuisine preference mismatch, or an invalid transport route go undetected, and 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., 2024b), and even frontier models frequently produce code that fails on edge cases in

[1]Oxford Dynamics, Harwell Science and Innovation Campus, United Kingdom. Correspondence to: [†]Shireen Kudukkil Manchingal <shireen.manchingal@oxdynamics.com>.

Accepted to the ICML 2026 Workshop on Statistical Frameworks for Uncertainty in Agentic Systems, Seoul, South Korea, 2026.
Copyright 2026 by the author(s).

competitive programming benchmarks (Chen et al., 2021; Li et al., 2023). The bottleneck, however, is not generation: among N candidates from the same LLM, at least one is often correct (Brown et al., 2024; Stiennon et al., 2020). The bottleneck is *selection*. Process reward models (Lightman et al., 2023; 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 from a result; and (ii) they operate on flat reasoning chains with no mechanism for verifying domain-specific constraints such as budgets, route validity, or logical consistency.

In this paper, we propose a novel *decomposed, distributional* energy function for scoring structured language model outputs (Figure 1, right). The energy splits into two terms: a learned quality score E_{quality}, which captures general reasoning quality, and a sum of analytical constraint penalties $E_{\text{constraint}}$, which encodes task-specific rules deterministically. The quality scorer is a heterogeneous ensemble of K low-rank adapters (LoRA; Hu et al., 2022) mounted on a single frozen encoder. The adapters are structurally heterogeneous (details in Section 3.2), so ensemble disagreement reflects genuine epistemic uncertainty about output quality rather than superficial variation from different random seeds (Fort et al., 2019; Lakshminarayanan et al., 2017; Gal & Ghahramani, 2016). The ensemble mean ranks candidates; the standard deviation drives a two-pass inference loop in which high uncertainty or constraint violations trigger targeted regeneration with structured feedback, and extreme uncertainty triggers abstention. Because the quality scorer learns general properties of well-structured reasoning, it transfers zero-shot to new task domains, while the modular constraint penalties handle domain-specific verification without retraining.

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

- **A novel modular energy decomposition with zero-shot transfer.** We decompose verification into a transferable learned quality scorer E_{quality} and swappable deterministic constraint penalties $E_{\text{constraint}}$: the scorer transfers zero-shot across domains (MuSR→GSM8K: 93.9%), while new tasks require only new constraint functions, not retraining. On TravelPlanner, the analytical constraint component reduces violations by 53% relative to single-shot Opus 4.6.
- **A novel distributional energy via heterogeneous LoRA ensemble.** We introduce a parameter-efficient ensemble of structurally diverse low-rank adapters (3% of backbone parameters) that produces an energy *distribution* rather than a point estimate, enabling confounding diagnostics (*e.g.*, detecting model-identity shortcuts via cross-model energy spreads) and σ-gated abstention.
- **Honest confounding analysis with causal verification.** We systematically evaluate whether the scorer learns reasoning quality or model-identity shortcuts, confirming genuine discrimination on four tasks. On code (TACO), a causal style-swap test proves the shortcut, mitigated via last-layer retraining (Kirichenko et al., 2023). Multi-model candidate diversity substantially improves selection (+13.3 pp on MuSR, +52.2 pp on TACO).

Paper outline. Section 2 reviews related work on chain-of-thought reasoning, energy-based models, and test-time compute. Section 3 presents the decomposed energy function, the heterogeneous LoRA ensemble, and the two-pass σ-triage mechanism. Section 4 evaluates the method across five benchmarks, analyses confounding, and reports cross-domain transfer. Section 5 distils lessons learned and outlines future directions. The appendix provides model and implementation details (§A, §B), training data and dataset analysis (§C, §D), and extended results including ablations, energy diagnostics, per-difficulty breakdowns, selective prediction, two-pass analysis, per-constraint comparisons, full transfer matrices, and the TACO confounding analysis (§E).

2. Related Work

Chain-of-thought prompting (Wei et al., 2022; Kojima et al., 2022) elicits step-by-step reasoning, and self-consistency (Wang et al., 2023) improves accuracy by marginalising over multiple reasoning paths. Least-to-most prompting (Zhou et al., 2023) decomposes problems into sub-problems, while Tree of Thoughts (Yao et al., 2023) extends deliberate search across branching reasoning steps. Process reward models provide finer-grained supervision: Uesato et al. (2022) and Lightman et al. (2023) show that step-level verification outperforms outcome-level verification, and Khalifa et al. (2025) extend this by training models that generate reasoning traces before scoring, achieving strong results with a fraction of process labels. All of these methods produce point-estimate scores and operate on flat reasoning chains, providing no mechanism for verifying that a structured, multi-section output satisfies global constraints jointly.

Energy-based models assign scalar energies to input–output pairs, with lower energy indicating higher compatibility (LeCun et al., 2006), and have been applied to text generation (Deng et al., 2020). Blondel et al. (2025) prove that autoregressive language models are themselves EBMs, grounding their use as verifiers over autoregressive outputs, and Carbone (2024) survey the broader landscape. For reasoning specifically, Jiang et al. (2025) train an outcome-level EBM

to rerank chain-of-thought traces on GSM8K and MATH, achieving 90.7% with a 55M-parameter encoder, and Chen et al. (2025) apply energy-based calibration to implicit chain-of-thought; Tan & Hong (2025) connect RL-tuned language models to EBMs theoretically. Across this literature, the energy function is treated as a deterministic scalar mapping: a single number with no indication of scorer reliability. Gleave & Irving (2022) highlight this gap for reward models more broadly, showing that uncertainty estimation improves reward model reliability, but their approach uses standard ensemble variance rather than structurally heterogeneous adapters. Our approach to learning from pairwise preferences connects to the reward modelling literature in RLHF (Christiano et al., 2017; Ouyang et al., 2022) and direct preference optimisation (Rafailov et al., 2023), but we target post-hoc verification of complete outputs rather than online policy optimisation. Our method produces a *distributional* energy via a heterogeneous ensemble, decomposes energy into learned and analytical components, and targets structured multi-section outputs rather than flat reasoning traces.

A complementary line of work studies how to allocate compute at test time. Snell et al. (2025) show that verifier-guided search can outperform scaling model parameters, Guo et al. (2025) demonstrate that reinforcement learning alone can incentivise advanced reasoning, and Brown et al. (2024) identify candidate *selection* as the key bottleneck: repeated sampling reveals a large gap between the accuracy of a single attempt and the best among many, yet existing selection methods close only a fraction of this gap. For structured planning, Lee et al. (2025) achieve 95.6% on TravelPlanner via evolutionary search with Gemini 1.5 Flash, Xia et al. (2025) show that reward modelling improves agent generalisation across diverse tasks, and Wang et al. (2025) demonstrate that combining outputs from multiple LLMs via a mixture-of-agents framework improves over any single model. Our method addresses the selection bottleneck via energy-based reranking and the verification gap via analytical constraint penalties, while using ensemble uncertainty to decide when selection is reliable enough to trust.

3. Methodology

We score candidate outputs from a frozen LLM with a decomposed energy function that combines a distributional learned quality term with deterministic constraint penalties, and select the candidate with lowest total energy. When the ensemble is uncertain or constraints are violated, a second generation pass uses structured feedback to elicit improved candidates. Figure 2 provides an overview.

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 denote the number of ensemble members by K, the energy produced by member k by $E^k_{\text{quality}}(x, y)$, and the analytical constraint penalty by $E_{\text{constraint}}(x, y)$. The energy function decomposes as:

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

and the selected candidate is the energy minimiser over the candidate pool $\mathcal{Y} = \{y_1, \ldots, y_N\}$:

$$y^* = \arg\min_{y \in \mathcal{Y}} E(x, y), \tag{2}$$

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

The quality scorer is an ensemble of K energy-based models that share a single frozen pretrained encoder. Each member maps a (problem, candidate) pair to a scalar energy; the K energies form a distribution whose mean ranks candidates and whose standard deviation quantifies scorer confidence.

The encoder is a pretrained bidirectional Transformer (Lee & Toutanova, 2018; Warner et al., 2025), frozen throughout training and inference. Each ensemble member k adds a lightweight low-rank adapter (Houlsby et al., 2019; Hu et al., 2022) that modifies a subset of the encoder's attention matrices, along with an independent energy head. Given a problem x and candidate y, the input is tokenised as $\text{input}_k = \texttt{[CLS]}\ x\ \texttt{[SEP]}\ y\ \texttt{[SEP]}$. The frozen encoder, augmented by adapter k, produces contextualised representations. The `[CLS]` representation $\mathbf{h}^k_{\text{CLS}}$ is passed through an energy head:

$$E^k_{\text{quality}}(x, y) = \text{MLP}_k\big(\text{LayerNorm}(\mathbf{h}^k_{\text{CLS}})\big), \tag{3}$$

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). Each adapter and energy head is trained independently; the frozen backbone is shared. The distributional output for candidate y is:

$$\mu_{\text{quality}}(x, y) = \frac{1}{K} \sum_{k=1}^{K} E^k_{\text{quality}}(x, y), \tag{4}$$

$$\sigma_{\text{quality}}(x, y) = \text{std}\big(\{E^1_{\text{quality}}, \ldots, E^K_{\text{quality}}\}\big). \tag{5}$$

The mean μ_{quality} is used for ranking (Equation (1)). The standard deviation σ_{quality} drives the two-pass inference loop (Section 3.5) and provides a calibration signal for post-hoc analysis.

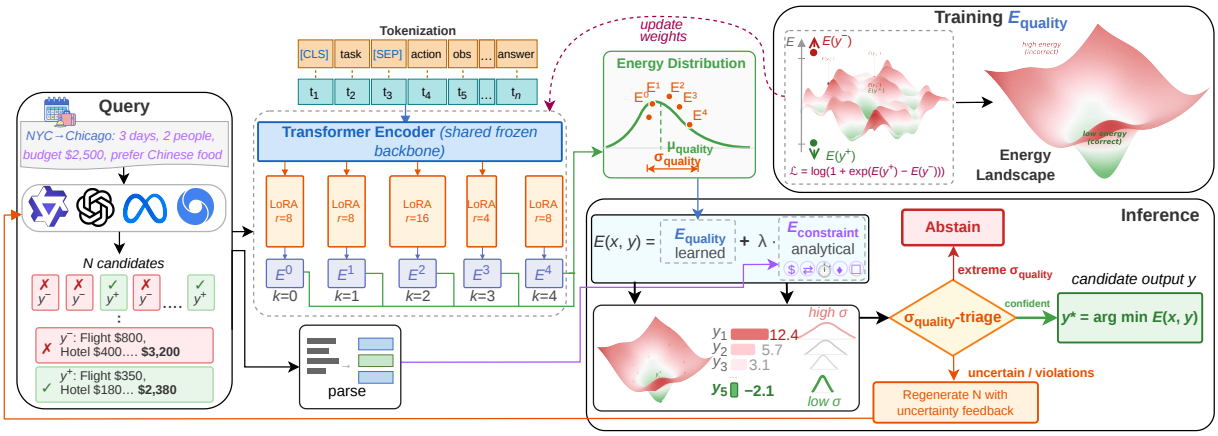

Figure 2. Method overview. **Top:** Training E_{quality}. From N candidates per problem, correct (y^+) and incorrect (y^-) pairs are formed. Each candidate is tokenised and passed through the frozen ModernBERT backbone augmented with one of $K{=}5$ heterogeneous LoRA adapters (varying rank $r \in \{4, 8, 16\}$ and target modules). Each adapter's MLP energy head maps the `[CLS]` representation to a scalar energy E^k. The Bradley–Terry loss pushes $E(y^+)$ below $E(y^-)$. The K energies form a distribution: μ_{quality} (mean) for ranking, σ_{quality} (spread) for confidence. Gradients update only the LoRA adapters and energy heads; the backbone remains frozen. **Bottom:** Inference. Four LLMs generate $N{=}32$ candidates per problem. Each candidate is scored in parallel by the $K{=}5$ ensemble (producing μ_{quality} and σ_{quality}) and by analytical constraint functions ($E_{\text{constraint}}$). Candidates are ranked by total energy $E = \mu_{\text{quality}} + \lambda \cdot E_{\text{constraint}}$. A σ_{quality}-triage step decides the outcome: if the ensemble is confident and constraints are satisfied, the best candidate is selected; if uncertainty is high or violations are detected, targeted regeneration with constraint feedback is triggered; if uncertainty remains extreme, the system abstains.

Ensemble diversity comes from three sources. First, the K adapters are *structurally heterogeneous*: they differ in rank, scaling factor, and which attention matrices they target (Table 6 in §B). This forces the adapters to operate in different regions of parameter space, so disagreement reflects genuine epistemic uncertainty rather than superficial variation from different random seeds (Fort et al., 2019; Lakshminarayanan et al., 2017). Second, each adapter is initialised with a different random seed. Third, each adapter trains on a random 80% subset of problems (data bagging; Breiman, 1996), ensuring that no two members see identical training distributions.

Because all K members share the frozen backbone, the trainable parameters per member consist only of the LoRA adapter weights and the energy head, amounting to roughly 3% of backbone parameters. The full ensemble thus adds modest overhead relative to a single scorer, while producing a distributional energy that a single model cannot.

3.3. Training

Each ensemble member is trained independently with the Bradley–Terry pairwise loss (Bradley & Terry, 1952). Let Y_x^+ and Y_x^- denote the sets of correct and incorrect outputs for problem x, and let $\mathcal{D}^k \subset \mathcal{D}$ denote the bagged 80% subset of training problems assigned to member k. The objective for member k is:

$$\mathcal{L}^k = \frac{1}{|\mathcal{D}^k|} \sum_{x \in \mathcal{D}^k} \frac{1}{|P_x|} \sum_{(y^+, y^-) \in P_x} \log\Big(1 + \exp\big(E^k_{\text{quality}}(x, y^+) - E^k_{\text{quality}}(x, y^-)\big)\Big), \quad (6)$$

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 Section 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

$E_{\text{constraint}}$ is a sum of analytical penalty terms, each encoding a specific task constraint:

$$E_{\text{constraint}}(x, y) = \sum_j w_j \cdot C_j(x, y), \quad (7)$$

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 §D.

3.5. Two-Pass Inference with σ-Triage

Single-pass reranking selects the best candidate from the initial pool, but cannot recover when no good candidate exists or when the scorer is uncertain. We extend the pipeline with a two-pass inference loop driven by ensemble uncertainty and constraint satisfaction (Algorithm 1 in §E.5). 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 $\theta_\sigma < \theta_{\text{abstain}}$:

$$\text{action}(y_1^*) = \begin{cases} \text{accept}, & \sigma \le \theta_\sigma \wedge C{=}0, \\ \text{regen.}, & \sigma \le \theta_a \vee C{>}0, \\ \text{abstain}, & \sigma > \theta_a, \end{cases} \tag{8}$$

where $\sigma = \sigma_{\text{quality}}(y_1^*)$, $C = E_{\text{constraint}}(y_1^*)$, and $\theta_a = \theta_{\text{abstain}}$. 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

$$y^* = \arg\min_{y \in \mathcal{Y}} E(x, y), \tag{9}$$

subject to a final abstention check on $\sigma_{\text{quality}}(y^*)$.

4. Experiments

We evaluate the Distributional EBM on five benchmarks spanning mathematical reasoning, multi-step narrative inference, multi-constraint planning, competitive programming, and deductive logic. Our experiments (Section 4.2) address five questions: **(i)** Is the bottleneck in structured LLM reasoning selection rather than generation? **(ii)** Does separating learned quality from analytical constraints improve correctness beyond either signal alone? **(iii)** Can ensemble disagreement reliably indicate when to trust, regenerate, or reject an output? **(iv)** Does the scorer learn genuine reasoning quality or exploit model-identity shortcuts? **(v)** Does the quality scorer transfer across reasoning domains without retraining? Extended results, including ablations (§E.1), energy diagnostics (§E.2), per-difficulty breakdowns (§E.3), selective prediction (§E.4), two-pass details (§E.5), TravelPlanner per-constraint analysis (§E.6), full transfer matrix (§E.7), TACO confounding (§E.8), dataset EDA (§D), and model/training details (§A, §B, §C), are in the Appendix.

4.1. Experimental Setup

Datasets. GSM8K (Cobbe et al., 2021) comprises 8,792 grade-school math word problems (7,473 train, 1,319 test). $E_{\text{constraint}}$ does not apply; the method reduces to μ_{quality} reranking alone. GSM8K is widely included in LLM pretraining corpora (Yang et al., 2023; Oren et al., 2024): candidate accuracy across all four generators averages 95%, and self-consistency reaches 97.3%. We report GSM8K results for completeness but treat the benchmark as partially contaminated and rely on MuSR as the uncontaminated headline. **MuSR** (Sprague et al., 2024) (Multi-step Soft Reasoning) consists of 756 narrative reasoning problems across three subsets: murder mysteries (2 choices), object placements (3 choices), and team allocation (3 choices). Released in 2024, it post-dates the pretraining of all four generators, providing an uncontaminated evaluation. Narratives span 3,000–6,000 characters, requiring multi-hop inference; average candidate accuracy is 57%, confirming genuine difficulty. **TravelPlanner** (Xie et al., 2024b) is a multi-constraint planning benchmark where models produce day-by-day itineraries in sole-planning mode. We use a 175/50 train/test split with 32 multi-model candidates per problem. Five analytical constraint terms (budget, connectivity, completeness, preferences, diversity) define $E_{\text{constraint}}$, verified against the sandbox database; candidates receive a continuous violation score in $[0, 1]$. **TACO** (Li et al., 2023) (Topics in Algorithmic Code generation) consists of competitive programming problems with execution-based verification. We use 601 train and 88 test problems (filtering `VERY_HARD` with 0% solve rate); binary labels come from automated test execution. Coverage is 100% at N=32. **Knights and Knaves** (K&K; Xie et al., 2024a) consists of 6,200 train and 700 test formal logic puzzles where each of n characters (2–8) is a knight (truth-teller) or knave (liar). A statement-consistency constraint checker serves as a perfect verifier. Per-model accuracy is extremely skewed: Gemma/GPT-OSS at ∼99% vs. Qwen/LLaMA at ∼15%. Primary metrics: pass@1 for GSM8K, MuSR, TACO, and K&K; mean violation score for TravelPlanner. Full dataset details (Table 8), constraint definitions, exploratory data analysis (§D.1), per-model candidate quality (§D.2), and qualitative examples (Table 10) are in §D; training data statistics are in §C.

Candidate generators and baselines. Table 1 lists the four frozen generators (Qwen Team, 2024; Grattafiori et al., 2024; Google DeepMind, 2026; OpenAI, 2025) and all selection strategies. Each model generates 8 candidates via temperature sampling (T=0.7, top-p=0.95 (Holtzman et al., 2020)), yielding N=32 per problem; all generation uses the OpenRouter API.[1] Pass 2 adds N_2=8 candidates from Qwen-2.5-7B conditioned on constraint feedback.

Scorer backbone. The primary backbone is ModernBERT-base (Warner et al., 2025) (149M parameters, 8,192 token context). DeBERTa-v3-base (He et al., 2023) (86M

[1] https://openrouter.ai

Table 1. Candidate generators and selection strategies.

Candidate generators (pool of N=32)	
Qwen-2.5-7B-Instruct	7B dense, T=0.7, top-p=0.95
LLaMA-3.1-8B-Instruct	8B dense, same params
Gemma-4-26B-IT	26B MoE (4B active)
GPT-OSS-20B	20B MoE (3.6B active)
Selection strategies	
Greedy (T=0)	1 deterministic candidate per model
Random	Uniform random from N=32
Self-consistency	Majority vote on extracted answers
$E_{\text{constraint}}$ only	Lowest analytical constraint penalty
Oracle	Best candidate by gold label (upper bound)
Distributional EBM (Ours)	K=5 **ensemble** $\mu_{\text{quality}} + \lambda \cdot E_{\text{constraint}}$
Distributional EBM + two-pass	**+ σ-triage + constraint feedback**

parameters, 512 token context) is evaluated as an ablation; backbone comparison, adapter configurations, and DeBERTa-specific results are in §A and §E.1. The heterogeneous LoRA configuration follows Table 6 in §B with K=5 adapters; LoRA and energy-head dropout are both 0.2.

Training. Each dataset trains its own scorer independently. Candidates are labelled by task-specific verification: answer matching for GSM8K and MuSR, constraint-violation scoring for TravelPlanner, execution-based testing for TACO, and exact-match with statement-consistency checking for K&K. Contrastive pairs are capped at 16 per problem and split 80/20 at the problem level. Optimiser: AdamW (Loshchilov & Hutter, 2019), weight decay 0.01, cosine annealing (Loshchilov & Hutter, 2017) with 10% linear warmup, gradient clipping at L_2 norm 1.0. Learning rate 5×10^{-5} (GSM8K, MuSR, TravelPlanner, TACO) or 2×10^{-5} (K&K). Batch size 32 (GSM8K, TACO), 16 (K&K, MuSR with DeBERTa), 8 (MuSR with ModernBERT at 4,096 token context). Early stopping with patience 2 (GSM8K, MuSR, TravelPlanner, TACO) or 3 (K&K). All training on a single NVIDIA L40S (46 GB); per-dataset configurations are in Table 7.

Inference. λ=1.0 (GSM8K, MuSR, TACO), λ=2.0 (TravelPlanner). σ-triage threshold θ_σ=0.8; abstention threshold θ_{abstain}=1.5. All evaluations use candidate shuffling (seed 42) to eliminate ordering bias.

4.2. Results

We organise the discussion around the five experimental questions posed above. Table 2 compares the Distributional EBM against greedy, random, self-consistency, and three frontier models across all five benchmarks. Table 3 reports zero-shot cross-domain transfer. The confounding analysis (question iv) is summarised in §E.8; two-pass σ-triage details are in §E.5; selective prediction curves are in §E.4.

(i) Selection vs. generation. On MuSR (Table 2), the oracle ceiling is 86.1% while greedy achieves only 61.2%, a 24.9 pp gap confirming that good candidates exist in the pool but single-shot generation fails to find them. The Distributional EBM closes 26% of this gap, reaching 67.7%, a +6.5 pp improvement over greedy and +3.3 pp over self-consistency. On TACO, the gap is even starker: oracle 100.0% versus random 46.6%, and the Distributional EBM recovers 79% of the gap (88.6%). On K&K, the Distributional EBM achieves 99.9% pass@1 (699/700), with gains scaling by puzzle difficulty: +32 pp on 2-character puzzles to +47 pp on 8-character puzzles (Table 16). On GSM8K, high contamination compresses the gap (oracle 98.5%, random 92.7%), and the Distributional EBM matches self-consistency within 0.3 pp. ***Multi-model candidate diversity is the single largest factor***: replacing Qwen-only pools with four-generator pools improves pass@1 by +13.3 pp on MuSR, +52.2 pp on TACO, and +59.8 pp on K&K (full ablations in §E.1). Figure 3 shows how the Distributional EBM front-loads correct candidates in its energy ranking; energy separation metrics are in §E.2.

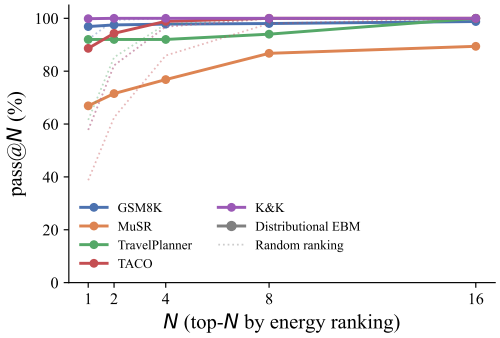

Figure 3. Pass@N (Chen et al., 2021) curves. The Distributional EBM front-loads correct candidates: pass@2 reaches 97.5% on GSM8K and 71.5% on MuSR. TravelPlanner uses a relaxed violation threshold (θ=0.15).

(ii) Learned quality + analytical constraints. On TravelPlanner, the combined energy achieves 0.046 mean violation, an 80% reduction over random (0.231) and a 52% reduction over self-consistency (0.096). Ablating the components reveals complementary contributions: E_{quality} alone achieves 0.058 mean violation, $E_{\text{constraint}}$ alone achieves 0.038, and the combination (0.046) outperforms E_{quality} alone while adding learned quality dimensions that the constraint checker does not cover. Figure 4 visualises the two-dimensional energy landscape; the two components agree on the top candidate in only 2% of problems, and in 28% of cases the combined energy selects a candidate that neither component would choose independently. On tasks without analytical constraints (GSM8K, MuSR, TACO), the method reduces to distributional μ_{quality} reranking alone and still provides substantial gains.

The Distributional EBM outperforms frontier models on TravelPlanner (Table 2): it reduces mean violations by 53% relative to single-shot Claude Opus 4.6 (Anthropic, 2026) and by 62% relative to Qwen-72B, because the analytical

Table 2. Results across five benchmarks. Greedy and frontier models generate deterministic candidates at T=0; all other strategies select from 32 multi-model 7B-class candidates. †: MuSR uses ModernBERT (8,192 tokens). ‡: TravelPlanner reports mean violation score ↓; all others report pass@1 ↑. K&K = Knights and Knaves. Best non-oracle result per column in **bold**; second-best underlined; shaded row is our method. §No model fingerprinting on K&K despite extreme generator accuracy gap.

Strategy	GSM8K (pass@1 ↑)	MuSR† (pass@1 ↑)	TravelPlanner‡ (mean viol. ↓)	TACO (pass@1 ↑)	K&K (pass@1 ↑)
Greedy (T=0)	91.8%	61.2%	0.0981	86.4%	**100.0%**
Random	92.7%	56.3%	0.231	46.6%	58.7%
Self-consistency	**97.3%**	64.4%	0.096	36.4%	56.4%
Claude Opus 4.6 (T=0)	97.0%	**72.0%**	0.098	**100.0%**	99.7%
Claude Sonnet 4.6 (T=0)	97.0%	68.0%	0.103	95.5%	98.7%
Qwen-72B (T=0)	95.4%	54.9%	0.121	64.8%	39.0%
Oracle	98.5%	86.1%	0.028	100.0%	100.0%
Distributional EBM (Ours)	97.0%	67.7%	**0.046**	88.6%	99.9%§

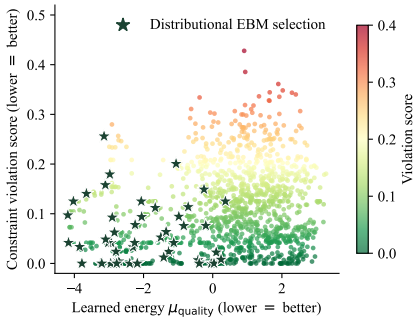

Figure 4. Energy decomposition on TravelPlanner (n=1,487 parseable candidates). Stars = EBM selections. μ_{quality} discriminates horizontally; $E_{\text{constraint}}$ vertically. Selections cluster in the low-energy, low-violation corner.

constraint checker catches hallucinated transport routes (24–28% satisfaction for frontier models) that the LLMs cannot self-detect (§E.6 provides the per-dimension breakdown).

To validate that the violation score reflects genuine plan quality, two independent LLM judges (Opus 4.6 and Sonnet 4.6 (Anthropic, 2026), T=0) score 120 candidates on a 1–5 scale using the original TravelPlanner criteria (Xie et al., 2024b). The judges agree strongly (Spearman ρ=0.67, $p < 10^{-16}$) and both correlate negatively with the violation score (Opus: ρ=−0.46; Sonnet: ρ=−0.38; both $p < 10^{-5}$). Opus 4.6 rates EBM-selected plans higher than oracle plans (3.86 vs. 3.67), suggesting the learned scorer captures holistic coherence beyond what constraints alone measure. The method does not match frontier models on reasoning or code generation (Table 2); its advantage is specific to TravelPlanner, where the analytical constraint checker provides a structural signal unavailable to any LLM. On TACO, the deconfounded EBM matches Sonnet 4.6 on MEDIUM (90.9%) and HARD (80.0%) problems; per-difficulty breakdowns are in §E.3 (Tables 15 to 17).

(iii) Ensemble uncertainty as a decision signal. Two-pass σ-triage helps when pass-1 has headroom and feedback is structured: on TravelPlanner (single-model), 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 reduces pass@1 by 1.7 pp because feedback causes the generator to second-guess memorised answers. For selective prediction (Geirhos et al., 2020a), 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-AUROC is 0.44 (*below* chance), making selective prediction counterproductive. Figure 5 visualises the $(\mu_{\text{quality}}, \sigma_{\text{quality}})$ space; full results are in §E.5 and §E.4.

(iv) Reasoning quality vs. model-identity shortcuts. When candidate generators have different accuracy levels, the scorer may learn to identify which model produced each candidate rather than evaluating reasoning quality (Geirhos et al., 2020b). We conduct a cross-dataset confounding analysis, computing per-model pick distributions and cross-model correct energy spreads for all five benchmarks. On MuSR, the EBM picks Qwen (the weaker model, 51% accuracy) most often (36%) and Gemma (the stronger model, 67%) least (18%), the opposite of what model fingerprinting would produce. On K&K, despite an extreme accuracy split (Gemma/GPT-OSS at 99% vs. Qwen/LLaMA at 15%), the EBM distributes picks uniformly (28/28/27/18%) with a cross-model correct energy spread of just 0.11. On GSM8K and TravelPlanner, picks are similarly distributed with spreads of 0.04 and 0.24 respectively. On TACO, by contrast, the initial scorer selected Gemma in 94% of test problems. A causal style-swap test (§E.8) confirmed the shortcut: identical verified-correct code received +1.95 higher energy when presented in another model's formatting. We mitigate this via deep feature reweighting (DFR; Kirichenko et al., 2023): retraining only the energy heads (0.4% of parameters) on group-balanced data collapses the cross-model correct energy spread from 6.67 to 0.34, distributes picks uniformly (22/24/34/20%), and retains 88.6% pass@1, with improved performance on HARD (+13.3 pp) and UNKNOWN (+14.4 pp) problems where the identity shortcut was least reliable. ***The determining factor is output format, not generator accuracy gap***: K&K has a more extreme accuracy split (88 pp) than TACO (74 pp), yet no fingerprinting, because reasoning text is stylistically similar across generators while code contains strong model-specific structural patterns.

(v) 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). Table 3 reports the key results.

Three patterns emerge. First, ***hard-to-easy transfer succeeds***: a MuSR-trained scorer achieves 96.1% on GSM8K without ever seeing a math problem, matching in-domain self-consistency (DeBERTa backbone; see §E.7 for back-

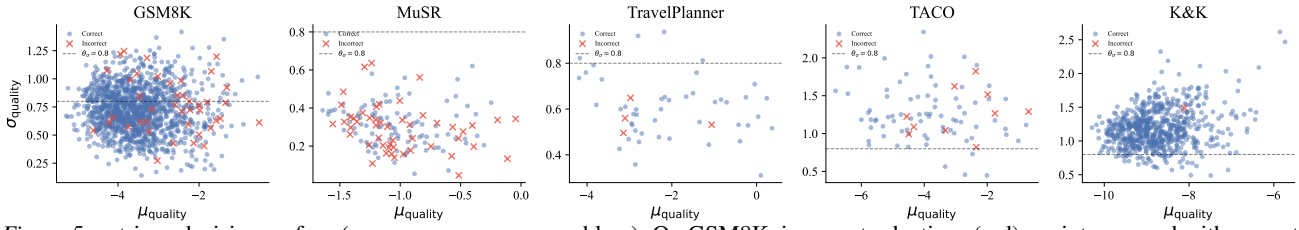

Figure 5. σ-triage decision surface (μ_{quality} vs. σ_{quality} per problem). On GSM8K, incorrect selections (red) are interspersed with correct ones. On MuSR, all $\sigma_{\text{quality}} < \theta_\sigma$=0.8 (dashed), so triage never triggers. On TravelPlanner, the wider σ_{quality} spread reflects continuous violation labels.

Table 3. Zero-shot transfer (Distributional EBM pass@1). Each row is a source checkpoint; each column is a target dataset. Shaded = in-domain; **Bold** = best per column; underlined = second-best. ◇Mean violation (lower = better; random = 0.231).

Source → Target	GSM8K	MuSR	TP	TACO	K&K
GSM8K (MB)	**97.0%**	64.9%	0.0%	39.8%	92.4%
MuSR (MB)	93.9%	**67.7%**	0.0%	36.4%	89.0%
TP (MB)	93.1%	60.9%	30.0%	56.4%	33.5%
TACO (MB)	92.5%	58.9%	**43.3%**	**88.6%**	31.1%
K&K (MB)	94.9%	64.9%	0.136◇	51.1%	**99.9%**
Self-consistency	96.1%	64.4%	12.0%	36.4%	56.4%
Random	91.7%	53.0%	14.0%	46.6%	58.7%

bone comparison). Training on difficult data with 57% candidate accuracy forces the scorer to learn genuine reasoning-quality signals that transfer to easier benchmarks. Second, *easy-to-hard transfer is weak*: the GSM8K checkpoint reaches only 64.9% on MuSR (vs. 67.7% in-domain), barely exceeding self-consistency (64.4%). High contamination in GSM8K (95% candidate accuracy) leaves too few informative negatives for the scorer to learn transferable discrimination. Third, *reasoning-to-code transfer is negative*: both GSM8K (−6.8 pp) and MuSR (−10.2 pp) checkpoints score below random on TACO, indicating that reasoning-quality signals actively mislead on code. Conversely, TACO→TravelPlanner transfers positively (+29.3 pp over random), suggesting that code-execution labels capture quality signals that partially generalise to constrained planning.

Interestingly, calibration transfers more broadly than ranking: a GSM8K-trained scorer achieves σ-AUROC of 0.79 on TravelPlanner zero-shot despite providing no ranking lift, suggesting that ensemble disagreement captures a domain-general difficulty signal. The full transfer matrix including MATH-500 and a heatmap visualisation (Figure 10) are in §E.7.

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 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. CDA (Kaushik et al., 2020) could further strengthen the TACO deconfounding beyond DFR (see paragraph iv). $E_{\text{constraint}}$ requires manual design per task, and the single-round feedback loop is less powerful than iterative refinement (Lee et al., 2025; Madaan et al., 2023). Full ablations are in §E.1.

5. Conclusion

Three lessons emerge from evaluating the Distributional EBM across five structured reasoning benchmarks. First, the quality of candidates matters more than the quality of scoring: multi-model diversity yields the largest gains (+13.3 pp on MuSR, +52.2 pp on TACO), and our scorer's value lies in reliably *selecting* from a diverse pool rather than in improving any single model's outputs. Second, a small encoder (149M parameters, 3% trainable) can match frontier models when paired with the right inductive biases: analytical constraints for TravelPlanner (−53% violations vs. Opus 4.6), Bradley–Terry contrastive training for reasoning, and multi-model candidate pools that supply the diversity the scorer needs. Third, honest confounding analysis is not just good practice but reveals the structure of what the scorer learns: on reasoning tasks, it evaluates content (uniform picks, cross-model energy spread <0.25); on code, it exploits formatting shortcuts (94% Gemma picks), fixable via last-layer retraining (Kirichenko et al., 2023) with no accuracy loss. The transferability of quality signals (a MuSR scorer achieving 93.9% on GSM8K zero-shot) suggests that structured reasoning shares a common quality signature across domains, provided the training data is difficult enough to force genuine discrimination.

Two directions follow naturally. First, replacing the heterogeneous LoRA ensemble with epistemic deep learning techniques (Manchingal & Cuzzolin, 2022; Manchingal et al., 2025) would provide an axiomatically grounded distributional energy without requiring multiple forward passes. Second, extending beyond single-round feedback to iterative energy-guided refinement could bridge the gap between single-pass reranking and evolutionary search methods that achieve higher pass rates at substantially greater compute cost.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. Uncertainty-aware verification of structured LLM outputs has beneficial applications in safety-critical domains such as automated planning, code generation, and multi-step reasoning, where undetected errors can propagate silently. However, overreliance on automated verification carries risks: the confounding analysis in this paper demonstrates that scorers can learn spurious shortcuts that produce high-confidence but unreliable outputs. Deploying such systems without ongoing audits could give users false confidence in LLM-generated plans or code, particularly in domains where the training distribution does not match deployment conditions.

References

Anthropic. The claude model family: Claude opus 4.6, claude sonnet 4.6. Technical report, 2026. https://docs.anthropic.com/en/docs/about-claude/models.

Ba, J., Kiros, J. R., and Hinton, G. E. Layer normalization. *ArXiv*, abs/1607.06450, 2016. URL https://api.semanticscholar.org/CorpusID:8236317.

Blondel, M., Sander, M. E., Vivier-Ardisson, G., Liu, T., and Roulet, V. Autoregressive language models are secretly energy-based models: Insights into the lookahead capabilities of next-token prediction. *arxiv:2512.15605*, 2025.

Bradley, R. A. and Terry, M. E. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

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

Brown, B., Juravsky, J., Ehrlich, R., Clark, R., Le, Q. V., Ré, C., and Mirhoseini, A. Large language monkeys: Scaling inference compute with repeated sampling. *arxiv:2407.21787*, 2024.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Carbone, D. A hitchhiker's guide to the relation of energy-based models with other generative models, sampling and statistical physics. *Transactions on Machine Learning Research*, 2024. arXiv:2406.13661.

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. D. O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., et al. Evaluating large language models trained on code. *arxiv:2107.03374*, 2021.

Chen, Z., Cui, S., Ye, D., Zhang, Y., Bian, Y., and Zhu, T. Think Consistently, Reason Efficiently: Energy-Based Calibration for Implicit Chain-of-Thought. *arxiv:2511.07124*, 2025.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. Training verifiers to solve math word problems, 2021. *URL https://arxiv.org/abs/2110.14168*, 9, 2021.

Deng, Y., Bakhtin, A., Ott, M., Szlam, A., and Ranzato, M. Residual energy-based models for text generation. 2020.

Fort, S., Hu, H., and Lakshminarayanan, B. Deep ensembles: A loss landscape perspective. *arXiv:1912.02757*, 2019.

Gal, Y. and Ghahramani, Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020a.

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020b.

Gleave, A. and Irving, G. Uncertainty estimation for language reward models. *arXiv:2203.07472*, 2022.

Google DeepMind. Gemma 4 technical report. Technical report, 2026.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. The llama 3 herd of models. *arxiv:2407.21783*, 2024.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *International*

conference on machine learning, pp. 1321–1330. PMLR, 2017.

Guo, D., Yang, D., Zhang, H., Song, J., Wang, P., Zhu, Q., Xu, R., Zhang, R., Ma, S., Bi, X., et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arxiv:2501.12948*, 2025.

He, P., Gao, J., and Chen, W. Debertav3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing, 2023. URL https://arxiv.org/abs/2111.09543.

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

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring Mathematical Problem Solving With the MATH Dataset, 2021. URL https://arxiv.org/abs/2103.03874.

Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y. The Curious Case of Neural Text Degeneration. In *International Conference on Learning Representations (ICLR)*, 2020. arXiv:1904.09751.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., de Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning (ICML)*, 2019. arXiv:1902.00751.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022. arXiv:2106.09685.

Jiang, E. H. et al. Learning to Rank Chain-of-Thought: Using a Small Model. *arXiv:2505.14999*, 2025.

Kaushik, D., Hovy, E., and Lipton, Z. C. Learning the Difference that Makes a Difference with Counterfactually-Augmented Data. In *International Conference on Learning Representations (ICLR)*, 2020. arXiv:1909.12434.

Khalifa, M. et al. Process Reward Models That Think. *arXiv:2504.16828*, 2025.

Kirichenko, P., Izmailov, P., and Wilson, A. G. Last Layer Re-Training is Sufficient for Robustness to Spurious Correlations. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2204.02937.

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. Large Language Models are Zero-Shot Reasoners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. arXiv:2205.11916.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. URL https://arxiv.org/abs/1612.01474.

LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., and Huang, F. J. A Tutorial on Energy-Based Learning. *Predicting Structured Data*, 2006.

Lee, J. and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. volume 3, pp. 4171–4186, 2018.

Lee, K.-H., Fischer, I., Wu, Y.-H., Marwood, D., Baluja, S., Schuurmans, D., and Chen, X. Evolving deeper LLM thinking. *arxiv:2501.09891*, 2025.

Li, R., Fu, J., Zhang, B.-W., Huang, T., Sun, Z., Lyu, C., Liu, G., Jin, Z., and Li, G. TACO: Topics in algorithmic code generation dataset. *arxiv:2312.14852*, 2023.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step. *arxiv:2305.20050*, 2023.

Loshchilov, I. and Hutter, F. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations (ICLR)*, 2017. arXiv:1608.03983.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. arXiv:1711.05101.

Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., Gupta, S., Majumder, B. P., Hermann, K. M., Welleck, S., Yazdanbakhsh, A., and Clark, P. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. arXiv:2303.17651.

Manchingal, S. K. and Cuzzolin, F. Epistemic deep learning. *arxiv:2206.07609*, 2022. URL https://arxiv.org/abs/2206.07609.

Manchingal, S. K., Mubashar, M., Wang, K., Shariatmadar, K., and Cuzzolin, F. Random-set neural networks (RS-NN). In *Thirteenth International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=pdjkikvCch.

OpenAI. GPT-4 technical report. Technical report, 2023. arXiv:2303.08774.

OpenAI. Introducing gpt-oss. Technical report, 2025. https://openai.com/index/introducing-gpt-oss/.

Oren, Y., Meister, N., Bhatt, N., Raji, I. D., and Cotterell, R. Proving test set contamination in black box language models. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2310.17623.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. arXiv:2203.02155.

Qwen Team. Qwen2.5 technical report. *arXiv:2412.15115*, 2024.

Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. arXiv:2305.18290.

Snell, C. V., Lee, J., Xu, K., and Kumar, A. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4FWAwZtd2n.

Sprague, Z., Ye, X., Bostrom, K., Chaudhuri, S., and Durrett, G. Musr: Testing the limits of chain-of-thought with multistep soft reasoning. In Kim, B., Yue, Y., Chaudhuri, S., Fragkiadaki, K., Khan, M., and Sun, Y. (eds.), *International Conference on Learning Representations*, volume 2024, pp. 14670–14728, 2024.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D. M., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. Learning to summarize from human feedback. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Tan, Z. and Hong, Y. A theoretical lens for rl-tuned language models via energy-based models. *arxiv:2512.18730*, 2025.

Uesato, J., Kushman, N., Kumar, R., Song, F., Siegel, N., Wang, L., Creswell, A., Irving, G., and Higgins, I. Solving math word problems with process- and outcome-based feedback. *ArXiv*, abs/2211.14275, 2022. URL https://api.semanticscholar.org/CorpusID:254017497.

Wang, J., WANG, J., Athiwaratkun, B., Zhang, C., and Zou, J. Mixture-of-agents enhances large language model capabilities. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=h0ZfDIrj7T.

Wang, X., Wei, J., Schuurmans, D., Le, Q. V., Chi, E. H., Narang, S., Chowdhery, A., and Zhou, D. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Warner, B., Chaffin, A., Clavié, B., Weller, O., Hallström, O., Taghadouini, S., Gallagher, A., Biswas, R., Ladhak, F., Aarsen, T., 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.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Xia, Y., Fan, J., Chen, W., Yan, S., Cong, X., Zhang, Z., Lu, Y., Lin, Y., Liu, Z., and Sun, M. AgentRM: Enhancing agent generalization with reward modeling. In Che, W., Nabende, J., Shutova, E., and Pilehvar, M. T. (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19277–19290, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.945. URL https://aclanthology.org/2025.acl-long.945/.

Xie, C., Huang, Y., Zhang, C., Yu, D., Chen, X., Lin, B. Y., Li, B., Ghazi, B., and Kumar, R. On memorization of large language models in logical reasoning. In *IJCNLP-AACL*, 2024a. URL https://api.semanticscholar.org/CorpusID:273695832.

Xie, J., Zhang, K., Chen, J., Zhu, T., Lou, R., Tian, Y., Xiao, Y., and Su, Y. Travelplanner: A benchmark for real-world planning with language agents. In *International Conference on Machine Learning*, 2024b. URL https://api.semanticscholar.org/CorpusID:267406800.

Yang, S., Chiang, W.-L., Zheng, L., Gonzalez, J. E., and Stoica, I. Rethinking benchmark and contamination for language models with rephrased

samples. *ArXiv*, abs/2311.04850, 2023. URL https://api.semanticscholar.org/CorpusID:265050721.

Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T. L., Cao, Y., and Narasimhan, K. R. Tree of thoughts: Deliberate problem solving with large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=5Xc1ecxO1h.

Zhou, D., Schärli, N., Hou, L., Wei, J., Scales, N., Wang, X., Schuurmans, D., Cui, C., Bousquet, O., Le, Q. V., and Chi, E. H. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WZH7099tgfM.

A. Model Details

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

A.1. Candidate Generators

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 (Qwen Team, 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 (Google DeepMind, 2026)	Mixture-of-Experts	26B (4B active)	128K	Strongest generator across all tasks
GPT-OSS-20B (OpenAI, 2025)	Mixture-of-Experts	20B (3.6B active)	128K	Reasoning model (extended thinking)

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 Table 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.

A.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, 2026), and Qwen-72B (Qwen Team, 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 (Section 4.2).

A.3. Scorer Backbones

Table 5. 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., 2023)	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).

B. Implementation Details

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

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 on a single NVIDIA L40S (46 GB); peak memory usage is $\sim$31 GB at batch size 32. Wall-clock time per adapter is $\sim$20 minutes per epoch on GSM8K and $\sim$5 minutes per epoch on MuSR.

C. Training Data Details

C.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.

C.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.

C.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%.

C.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 (§E.8) confirmed this imbalance caused a model-identity shortcut; DFR (Kirichenko et al., 2023) retraining on group-balanced data mitigates it.

D. Dataset Details

D.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

D.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 scorer selected Gemma in 98% of test problems; after DFR deconfounding, picks are distributed uniformly (22/24/34/20%).

D.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 method reduces to distributional μ_{quality} reranking only.

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 92.7% and self-consistency at 97.3% 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.

D.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 method reduces to distributional reranking. The primary metric is pass@1.

D.5. TravelPlanner

TravelPlanner (Xie et al., 2024b) 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., 2024b), 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).

D.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.

D.7. Knights and Knaves

Knights and Knaves (K&K) puzzles (Xie et al., 2024a) 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. (2024a) 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 $E_{\text{constraint}}=0 \Leftrightarrow$ correct on all 34,120 candidates checked, confirming perfect verification.

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 $E{=}-4.47$

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 Section 4.2).

D.8. Qualitative Examples

Table 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.

E. Extended Results and Analysis

This section consolidates all extended results and analysis beyond the main paper. §E.1 isolates five design choices (ensemble size, pool size, multi-model training, backbone, energy decomposition). §E.2 reports energy separation metrics and calibration diagnostics. §E.3 breaks down per-difficulty performance on TACO, K&K, and MuSR. §E.4 evaluates

selective prediction via σ-abstention. §E.5 analyses the two-pass σ-triage mechanism across datasets and pool configurations. §E.6 decomposes TravelPlanner violations per constraint dimension, comparing the Distributional EBM against frontier models. §E.7 reports the full zero-shot transfer matrix including DeBERTa backbones and MATH-500. §E.8 presents the six-phase TACO confounding analysis and DFR mitigation.

E.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 (Table 11, Figure 6). Increasing K from 1 to 5 has negligible effect on pass@1 across all datasets: GSM8K stays at 97.0–97.2%, MuSR fluctuates between 64.9% and 66.9%, and TACO is constant at 88.6%. The one exception is TravelPlanner, where $K\geq3$ 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 (Table 12, Figure 7). The Distributional EBM'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.

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	97.0%	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%	—	—	—

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

	GSM8K		MuSR		TACO	
N	EBM	Random	EBM	Random	EBM	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	**97.0%**	92.7%	65.6%	58.9%	**92.0%**	47.7%

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

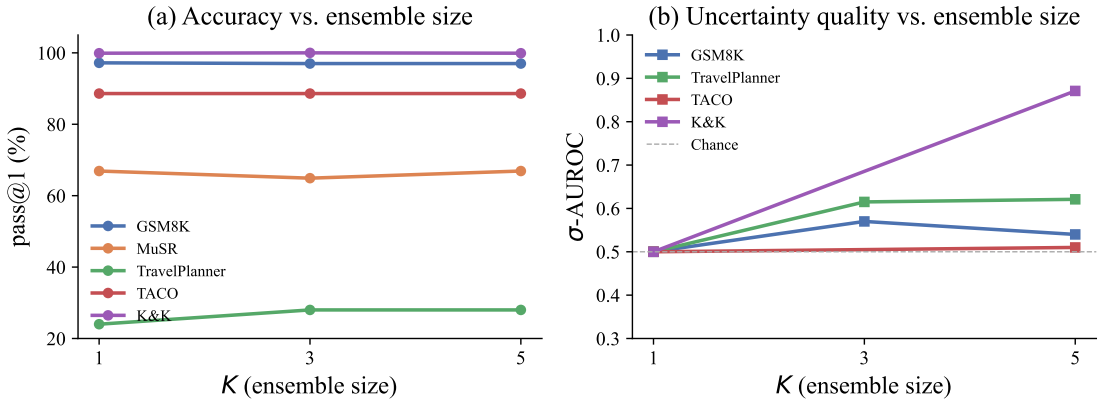

Figure 6. Effect of ensemble size K. **(a)** Pass@1 is stable across K on all datasets; the ensemble does not improve top-1 accuracy. **(b)** σ-AUROC improves from chance (0.5 at $K{=}1$) to 0.87 on K&K at $K{=}5$, confirming that structurally heterogeneous adapters produce informative uncertainty. The ensemble is justified by uncertainty estimation, not accuracy.

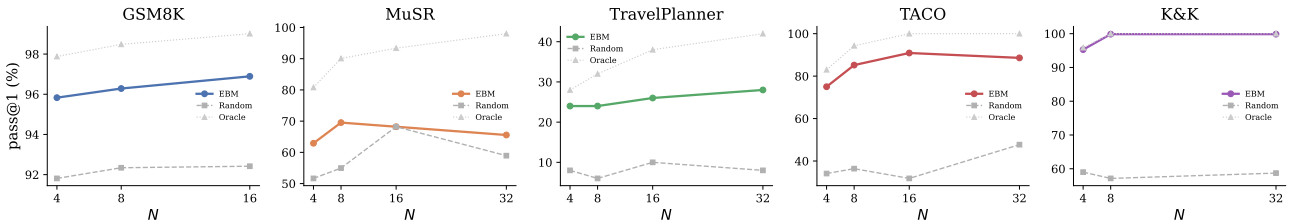

Figure 7. Candidate pool scaling. The Distributional EBM's advantage over random grows with N on GSM8K; on MuSR the advantage peaks at $N{=}8$ and diminishes as the pool saturates.

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 (Table 2) 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 EBM	93.1%	53.6%	8.0%	36.4%	40.1%
Multi-model EBM	**97.0%**	**66.9%**	**28.0%**	**88.6%**	**99.9%**
Δ	+3.9 pp	+13.3 pp	+20.0 pp	+52.2 pp	+59.8 pp

for contrastive learning and higher oracle coverage (98.0% vs. 86.8%). On GSM8K, the gain is smallest (+3.9 pp) because all models are 91–97% correct (contamination), leaving little room for multi-model diversity to help.

E.2. Energy Separation and Calibration

Table 14 and Figure 9 report diagnostic metrics for the Distributional EBM across all five benchmarks; Figure 8 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 verifier 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).

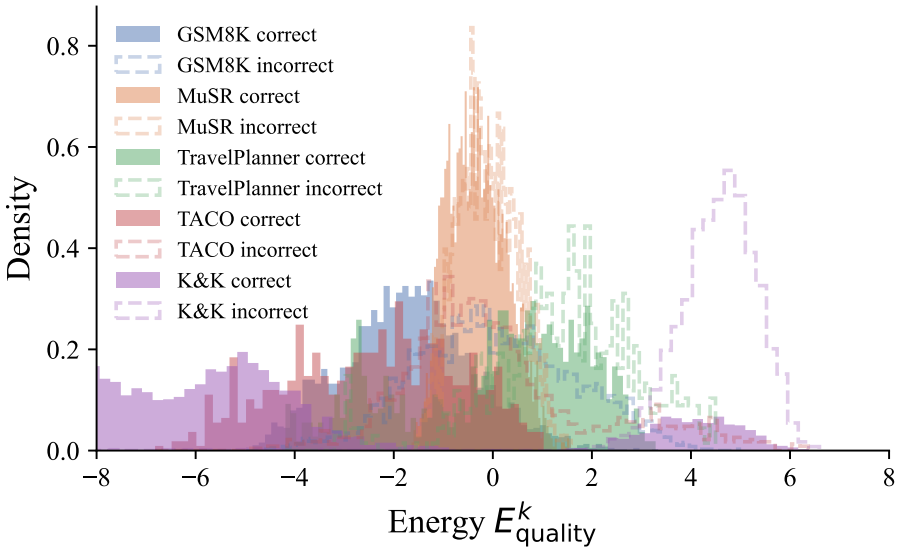

Figure 8. Energy distributions for correct (filled) and incorrect (dashed outline) candidates on GSM8K and MuSR (ModernBERT, multi-model). GSM8K shows clear bimodal separation; MuSR distributions overlap substantially, consistent with the smaller energy gap (0.19 vs. 1.76) and the difficulty of discriminating correct from incorrect narrative reasoning.

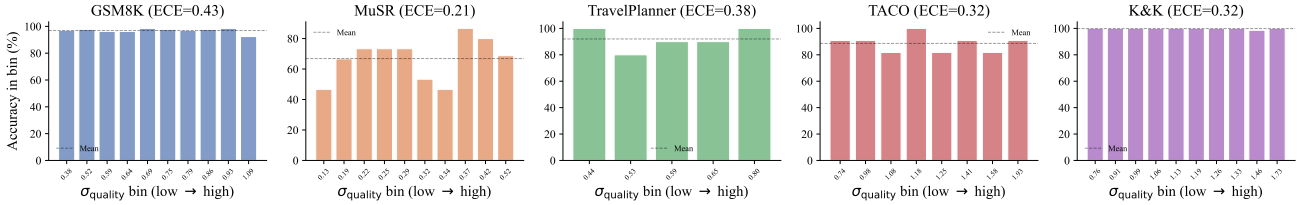

Figure 9. Calibration reliability diagrams. Bars show accuracy within each σ_{quality} bin. On GSM8K, accuracy is uniformly high across all bins (ECE = 0.43, the scorer is poorly calibrated). On MuSR, accuracy fluctuates non-monotonically with σ_{quality}, confirming that the uncertainty signal is not reliably informative on this task.

Table 14. Energy separation metrics (ModernBERT, multi-model, K=5). Energy gap = $\bar{E}(\text{incorrect}) - \bar{E}(\text{correct})$; positive means the 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
$\bar{E}(\text{correct})$	−1.71	−0.32	−1.26	−2.44	−4.47
$\bar{E}(\text{incorrect})$	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

K&K exhibits the largest energy gap (9.00), reflecting the combination of binary correctness labels, a perfect constraint verifier, and high candidate diversity across generators. TACO's deconfounded gap (2.49) reflects genuine quality discrimination after removing the identity shortcut (pre-DFR: 4.30). MuSR has the smallest gap (0.19), consistent with the difficulty of discriminating correct from incorrect narrative reasoning. TravelPlanner's continuous labels produce a moderate gap (2.00) with the highest Kendall τ (0.49), indicating that continuous violation scores produce smoother energy landscapes.

E.3. Per-Difficulty Analysis

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

TACO (Table 15). The deconfounded EBM 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 15. TACO pass@1 by difficulty (multi-model). The EBM matches or exceeds Sonnet 4.6 on MEDIUM problems. n = number of test problems.

Difficulty	EBM	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

Knights and Knaves (Table 16). The multi-model EBM achieves 100% pass@1 on all tiers except 8-character puzzles (99.0%), demonstrating that the scorer maintains 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 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 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 16. K&K pass@1 by puzzle size (number of characters). Multi-model EBM achieves 100% on all tiers except 8ppl. **Bold** marks the best result per tier. Single-model EBM gains are largest on the hardest tiers.

	Multi-model (N=32)			Single-model (N=16)			Frontier	
Tier	Greedy	EBM	Lift	Greedy	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 (Table 17). 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 EBM's gains are largest on the 3-choice subsets where the random baseline is lower and there is more room for discrimination.

Table 17. 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

E.4. Selective Prediction via σ-Abstention

Table 18 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 (97.0–97.4%) 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 18. 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 shows degradation (anti-correlated σ-AUROC).

Abstain %	GSM8K (MB)	MuSR (MB)	TravelPlanner	TACO	K&K
0%	97.0%	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%**

E.5. Two-Pass σ-Triage Analysis

Algorithm 1 provides the full inference pseudocode for the two-pass σ-triage mechanism described in Section 3.5.

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**
ABSTAIN
17: **else**
y^*
18: **end if**

Table 19 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) fails catastrophically ($-$89.8 pp) because all 88 problems are triaged and pass-2 candidates disrupt the Gemma-centric

energy landscape. After DFR deconfounding, TACO two-pass is neutral (88.6%→88.6%) because the deconfounded 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.

Table 19. 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%	2.3%	−89.8 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

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 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%). TACO multi-model fails catastrophically because: (a) all 88 problems trigger triage ($\sigma_{\text{pool}} > 0.8$ due to high inter-model disagreement), (b) pass-2 candidates from Qwen-7B with opaque error feedback are uniformly poor, and (c) re-pooling these low-quality candidates disrupts the energy landscape established by the strong pass-1 pool. On KK (single-model), two-pass is net neutral (−0.2,pp): constraint feedback helps on easy tiers (2–4ppl:+2–4,pp) but hurts on hard tiers (7–8ppl: −10 -13,pp), where pass-2 candidates receive low energy despite being incorrect.

E.6. TravelPlanner: Per-Constraint Frontier Comparison

Table 20 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 EBM+$E_{\text{constraint}}$ system near-perfect connectivity (86% satisfaction). Conversely, frontier models outperform on budget (Opus: 0.040 vs. EBM+$E_{\text{constraint}}$: 0.030), reflecting stronger numerical reasoning. Preferences are the hardest dimension for all methods, including oracle (0.180 mean violation).

Table 20. Per-violation-dimension mean scores on TravelPlanner (50 test problems). Lower is better. **Bold** marks the best (lowest) value per row.

Dimension	Oracle	$E_{\text{constraint}}$ only	EBM+$E_{\text{constraint}}$	EBM	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 EBM+$E_{\text{constraint}}$ system a structural advantage on this dimension. Conversely, frontier models outperform on budget (Opus 4.6: 0.040 vs. EBM+$E_{\text{constraint}}$: 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).

E.7. Zero-Shot Transfer: Full Matrix

Table 21 reports the full zero-shot transfer matrix, including both ModernBERT and DeBERTa checkpoints and the MATH-500 benchmark. Figure 10 visualises the transfer asymmetry as lift over random.

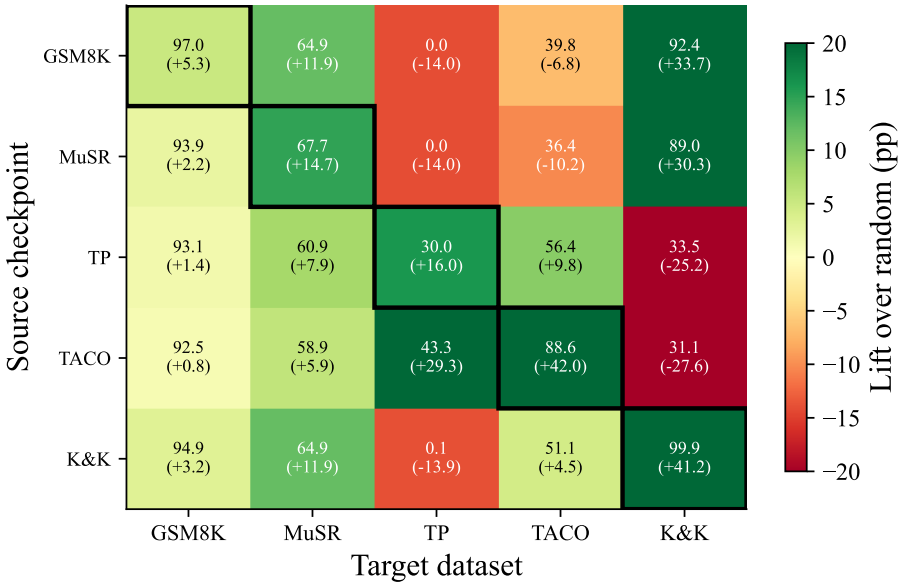

Figure 10. Zero-shot transfer heatmap (ModernBERT): each cell shows pass@1 and (lift over random). Colour encodes lift: green = positive transfer, red = negative. Black borders mark in-domain entries.

Four patterns are evident. First, *hard-to-easy reasoning transfer works*: both GSM8K and MuSR checkpoints achieve 89–92% on K&K, and the MuSR-trained DeBERTa achieves 96.1% on GSM8K, matching in-domain self-consistency. Second, *easy-to-hard reasoning transfer is weak*: the GSM8K checkpoint reaches only 64.9% on MuSR (vs. 67.7% in-domain), because the contaminated training data provides too few informative negatives. Third, *reasoning-to-code transfer is negative*: both GSM8K (39.8%) and MuSR (36.4%) checkpoints score *below* random (46.6%) on TACO, indicating that reasoning-quality signals actively mislead on code. Fourth, *no checkpoint transfers to MATH-500*: all achieve $\leq$68% vs. self-consistency at 72.6%, likely due to the format mismatch (`\boxed{}` vs. `####`).

Key observations: (i) MuSR→GSM8K transfer (96.1% with DeBERTa) matches in-domain self-consistency, confirming that hard-data training produces transferable quality signals. (ii) No checkpoint transfers positively to MATH-500 (all $\leq$ 67.8%, below SC at 72.6%), likely due to format mismatch (`\boxed{}` vs. `####`). (iii) Reasoning→code transfer is negative: GSM8K (39.8%) and MuSR (36.4%) both score below TACO random (46.6%), indicating that reasoning-quality signals actively mislead on code. (iv) TACO→TravelPlanner shows strong positive transfer (43.3% vs. 14.0% random, +29.3 pp), and TravelPlanner→TACO also transfers positively (56.4% vs. 46.6% random, +9.8 pp), suggesting that execution-based labels produce more generalisable quality signals than reasoning-based labels. (v) DeBERTa backbone transfer to K&K is asymmetric: MuSR(DB)→K&K retains 88.4% (-0.6 pp vs. ModernBERT), while GSM8K(DB)→K&K drops to 68.4% (-24.0 pp). The MuSR-trained scorer learns quality signals robust to 512-token truncation, whereas the GSM8K scorer, trained on contaminated data with 95% candidate accuracy, relies on shallow features that are lost when inputs are truncated.

E.8. Confounding Analysis

We screen all five datasets for model-identity confounding by computing two diagnostics: (i) *per-model pick distribution* (does the scorer disproportionately select one generator?) and (ii) *cross-model correct energy spread* (do correct candidates from different generators receive similar energies?). Table 22 summarises the results.

Table 21. Full zero-shot transfer matrix (EBM 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%	30.0%	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	91.7%	53.0%	68.4%	14.0%	46.6%	58.7%

Table 22. Confounding screening across all datasets. Pick distribution shows the fraction of test problems where each generator's candidate is selected. Energy spread is the standard deviation of mean correct energy across generators (lower = less confounded).

Dataset	Qwen-2.5-7B	LLaMA-3.1-8B	Gemma-4-26B	GPT-OSS-20B	Energy 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

Four datasets show near-uniform pick distributions and energy spreads below 0.25, indicating no model-identity confounding. On MuSR, the 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 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 scorer revealed that the initial model exploited a model-identity shortcut rather than genuine reasoning-quality discrimination.

Phase 1-2: Detection. The 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 +1.1 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 normalization ablation. Progressive removal of formatting features (headers, LaTeX, IO normalization, 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., 2020b).

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 scorer occurred on EASY problems; on 7 problems the 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. 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 scorer improves pass@1 on HARD (+13.3 pp) and UNKNOWN (+14.4 pp) problems and is robust to two-pass re-pooling (88.6% $\rightarrow$ 88.6%, vs. pre-DFR catastrophic collapse 92.0% $\rightarrow$ 2.3%).